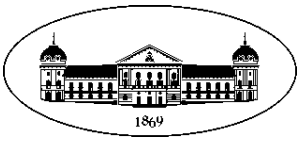


**BULGARIAN ACADEMY OF SCIENCES**

**INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES**

**MIROSLAVA DONCHEVA DIMITROVA**

**DISSERTATION**

# Evaluation Framework of Retrieval-Augmented Generation

**FOR THE DEGREE OF**

**Doctor of Philosophy (PhD)**

**IN "INFORMATICS" PROGRAM**

**PROFESSIONAL FIELD 4.6. "INFORMATICS AND COMPUTER SCIENCES"**

**SCIENTIFIC SUPERVISOR: ACAD. IVAN POPCHEV**

**Sofia, 2026**

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF EQUATIONS

# LIST OF ALGORITHMS

# GLOSSARY OF TERMS AND ABBREVIATIONS

*This glossary contains 148 terms and abbreviations used throughout.*

| TERM / ABBREVIATION | DEFINITION |
|---|---|
| **Adaptive Threshold Selection** | Strategy where the similarity threshold varies by query characteristics rather than remaining fixed. |
| **Anchor Wallet** | Client-side signing workflow for Antelope blockchain transactions, keeping private keys outside the web application. |
| **Answer Relevancy** | Whether the generated response addresses the user query intent. |
| **Antelope Blockchain** | Public, open-source distributed ledger (formerly EOSIO) used for tamper-evident provenance logging of experimental configurations. |
| **Audit Trail** | Verifiable sequence of recorded experimental runs and associated artifacts. |
| **Balance Score** | Stability–performance trade-off metric: Balance Score = (T-CPS improvement % / 100) / CV. Higher values indicate better improvement-to-variability ratio. |
| **BART (Bidirectional and Auto-Regressive Transformers)** | Sequence-to-sequence architecture combining bidirectional encoder with auto-regressive decoder, used in foundational RAG systems. |
| **Baseline** | Same RAG pipeline with similarity threshold disabled; locked definition for Phases III–IV comparisons. |
| **BERT (Bidirectional Encoder Representations from Transformers)** | Transformer model family achieving bidirectional context understanding, used for contextual embeddings and similarity-based evaluation. |
| **BERTScore** | Semantic similarity metric using contextual token embeddings; reports precision, recall, and F1 based on token alignment. |
| **BIG-bench Hard** | Challenging subset of BIG-bench benchmark used for evaluating reasoning capabilities. |
| **BLEU (Bilingual Evaluation Understudy)** | N-gram precision metric with brevity penalty, measuring accuracy and fluency compared to reference texts. |
| **Blockchain Provenance Logging** | Recording configurations and outputs on a blockchain to support auditability, integrity, and independent verification. |
| **BP (Brevity Penalty)** | BLEU component penalizing overly short outputs to discourage truncated responses. |
| **B-RT (BERT-based Reference-free Text)** | Human-readability proxy metric suite providing coherence, consistency, fluency, and relevance signals. |
| **B-RT.Average** | Aggregate B-RT score computed as arithmetic mean of component metrics. |
| **B-RT.Fluency** | Fluency-oriented B-RT sub-metric assessing linguistic quality. |

| | |
|---|---|
| **ChromaDB** | Open-source vector database optimized for high-dimensional similarity search, licensed under Apache 2.0. |
| **Chunk Length** | Size of each text segment used during document ingestion (500 characters in experimental configuration). |
| **Chunk Overlap** | Shared text between consecutive chunks preserving context continuity (100 characters in experimental configuration). |
| **Chunking** | Segmenting documents into smaller passages for indexing and retrieval. |
| **CLS Token** | Special classification token in BERT-style models whose embedding represents the entire input sequence. |
| **Cohen's d** | Effect size measure calculated as mean difference divided by pooled standard deviation. |
| **Composite Performance Score (CPS)** | Weighted aggregate score combining normalized component metrics into a single performance index using min-max normalization with polarity adjustment. |
| **Context Buffer** | Maximum token budget available for prompt plus retrieved context (16k tokens locked in Phase IV). |
| **Context Precision** | How much of the retrieved context is actually relevant (inverse of retrieval noise). |
| **Context Recall** | Whether retrieved context includes the information needed to answer the query. |
| **Context Window** | Maximum token sequence length a model can process in a single inference pass. |
| **Corpus** | A structured collection of documents or text passages serving as the retrieval source in a RAG system. |
| **Correlation Matrix** | Table of pairwise correlations among metrics used to assess redundancy and relationships. |
| **Cosine Similarity** | Similarity measure based on the angle between two vectors, ranging from −1 to 1. |
| **CV (Coefficient of Variation)** | Standard deviation divided by mean; used as a stability/consistency proxy in T-CPS formulation. |
| **Declaration of Originality** | Required statement affirming authorship and originality under Bulgarian Academy institutional rules (Art. 27(2) ЗРАСРБ). |
| **DeepEval** | Testing-harness toolkit providing automated evaluation suites for prompt-based or RAG pipelines. |
| **DeepSeek R1 8B** | Open-source 8B-parameter reasoning-focused model evaluated in Phases III–IV. |
| **Dense Passage Retrieval (DPR)** | Dense embedding-based retrieval placing queries and documents in shared high-dimensional space. |
| **Document Ingestion** | Process of loading, chunking, embedding, and indexing documents into the vector database. |
| **Effect Size** | Magnitude of difference between conditions; classified as negligible (<0.2), small (0.2–0.5), medium (0.5–0.8), or large (≥0.8). |

| | |
|---|---|
| **Embedding** | Numeric vector representation of text used for similarity search in high-dimensional space. |
| **Embedding Model** | Model used to generate embeddings for documents and queries. |
| **End-to-End Evaluation** | Evaluation covering the full pipeline from retrieval through generation and metric computation. |
| **EOSIO** | Original blockchain technology lineage of Antelope, referenced in platform history. |
| **F1 Score** | Harmonic mean of precision and recall, balancing correctness and completeness. |
| **Factual Alignment** | The degree to which generated text accurately reflects verifiable information from retrieved evidence or established facts. |
| **FAO (Food and Agriculture Organization)** | United Nations agency; source of Climate-Smart Agriculture Sourcebook used in agricultural corpus. |
| **FiD (Fusion-in-Decoder)** | RAG architecture enabling synthesis from multiple retrieved passages through decoder attention. |
| **FiD-Light** | Efficient variant of Fusion-in-Decoder using selective attention mechanisms. |
| **Forward Pass / Inference** | Runtime generation step performed by a model during evaluation. |
| **Fragmented Context** | Failure point where retrieved passages are individually relevant but collectively incoherent. |
| **Generation Context** | The set of retrieved passages or documents provided to a language model as input conditioning during text generation. |
| **GQA (Grouped-Query Attention)** | Attention mechanism sharing key-value projections across query heads, improving inference efficiency. |
| **Granite 3.2 8B** | IBM's open-source 8B-parameter enterprise-oriented model evaluated in Phases III–IV. |
| **Grounding** | Constraining generated output to retrieved evidence rather than unsupported parametric inference. |
| **GSM8K** | Grade-school math benchmark testing mathematical reasoning capabilities. |
| **Hallucination** | Generated content not supported by provided evidence or trustworthy sources. |
| **Heatmap** | Visualization of differences from baseline across thresholds and models. |
| **Hugging Face** | Ecosystem and platform for accessing open-source models and NLP tooling. |
| **Information Integration** | Ability to synthesize an answer using evidence from multiple passages/sources. |
| **IPFS (InterPlanetary File System)** | Content-addressed distributed storage referenced for artifact addressing and integrity. |
| **KILT** | Knowledge Intensive Language Tasks benchmark for evaluating knowledge-intensive NLP. |

| | |
|---|---|
| **K-Inc (K Increment)** | Step size for iterative retrieval procedures (locked: K-Inc = 2 after Phase I). |
| **kNN-LM** | Language model augmenting predictions through interpolation with nearest-neighbor distribution. |
| **LangChain** | Library for building applications with language models, used for retriever implementation. |
| **Laplace Perplexity** | Perplexity computed using an n-gram language model with Laplace (add-one) smoothing to handle unseen n-grams. |
| **Laplace Smoothing** | Add-one smoothing that allocates probability mass to unseen n-grams in n-gram language models. |
| **Latent Semantic Analysis (LSA)** | Dimensionality-reduction approach addressing synonymy and polysemy to improve semantic relevance estimation. |
| **LCS (Longest Common Subsequence)** | Sequence overlap capturing ordered but not necessarily contiguous matches, used by ROUGE-L. |
| **Lidstone Perplexity** | Perplexity computed using an n-gram language model with Lidstone (add-λ) smoothing (λ = 0.1). |
| **Lidstone Smoothing** | Add-λ smoothing (λ > 0; λ = 0.1 in experimental configuration), a finer-grained alternative to Laplace smoothing. |
| **Llama 2 7B** | Meta's open-source 7B-parameter general-purpose model evaluated in Phase II. |
| **Llama 3.1 8B** | Meta's open-source 8B-parameter model with extended context support, evaluated in Phases III–IV. |
| **LLM/s (Large Language Model/s)** | Transformer-based language model scaled to billions of parameters for text generation and understanding. |
| **LoRA (Low-Rank Adaptation)** | Parameter-efficient fine-tuning method selectively updating model layers. |
| **Mac Mini M1** | Apple hardware platform with 16 GB RAM and 10-core GPU used in Phase IV experiments. |
| **Mac Mini M2** | Apple hardware platform used in earlier experimentation. |
| **Maximal Marginal Relevance (MMR)** | Re-ranking strategy balancing relevance and diversity among retrieved results. |
| **Maximum Inner Product Search (MIPS)** | Retrieval approach maximizing inner product between query and document embeddings. |
| **Mean CPS** | Average Composite Performance Score across questions for a given configuration. |
| **METEOR (Metric for Evaluation of Translation with Explicit ORdering)** | Evaluation metric balancing precision and recall with stemming, synonym matching, and fragmentation penalty. |
| **Mistral 7B** | Mistral AI's open-source 7B-parameter efficiency-focused model evaluated across all phases. |
| **Mistral 7B v0.3** | Updated version of Mistral 7B with extended 32k context window evaluated in Phases III–IV. |

| | |
|---|---|
| **MMLU (Massive Multitask Language Understanding)** | Benchmark evaluating model knowledge and reasoning across diverse subjects. |
| **Natural Language Generation (NLG)** | A subfield of Natural Language Processing (NLP) focused on producing human-readable text from data or intermediate representations. In the context of large language models (LLMs) and Retrieval-Augmented Generation (RAG), NLG refers to the model's decoding step that generates the final response conditioned on the prompt and any retrieved context |
| **Natural Questions** | Open-domain question answering benchmark using real Google queries. |
| **Negative Rejection** | Ability to abstain when evidence is insufficient, rather than hallucinating an answer. |
| **N-Gram** | Contiguous sequence of n tokens used in overlap metrics and language modeling. |
| **NLP (Natural Language Processing)** | Methods and systems for processing, understanding, and generating human language. |
| **NLTK (Natural Language Toolkit)** | Python library for language processing and n-gram modeling used in metric computation. |
| **Noise Robustness** | Performance when irrelevant passages are included alongside relevant evidence. |
| **Normal Mode** | Retrieval mode using standard LangChain VectorStoreRetriever with top-k selection. |
| **Ollama** | API providing unified interface for managing and invoking open-source LLMs across operating systems. |
| **Orca 2 7B** | Microsoft's fine-tuned derivative of Llama 2 7B emphasizing reasoning, evaluated in Phase II. |
| **Paired T-Test** | Two-tailed statistical test comparing per-question results against baseline (uncorrected p-values). |
| **Parametric Knowledge** | Factual information encoded in a language model's weight parameters during training, accessible without external retrieval. |
| **PaSSER (Platform for Systematic and Structured Evaluation of RAG)** | Web-based evaluation platform developed for threshold-aware RAG assessment with blockchain provenance logging. |
| **Pearson Correlation (r)** | Linear association measure quantifying strength and direction of relationship between variables. |
| **Perplexity (PPL)** | Proxy indicator of surface predictability under an n-gram model; lower values indicate higher typicality. |
| **Precision** | Proportion of predicted positives that are correct; in retrieval contexts, measures how much of returned content is relevant. |
| **Provenance** | Documented origin and configuration context of experimental outputs. |

| | |
|---|---|
| **P-Value** | Probability under null hypothesis of observing results at least as extreme as measured. |
| **Pyntelope** | Python library for Antelope blockchain interactions used in PaSSER. |
| **Q&A Pair (Question–Answer Pair)** | Evaluation unit consisting of a question and its reference answer. |
| **Query** | Input question or prompt used to retrieve evidence and generate an answer. |
| **RAGAS (Retrieval-Augmented Generation Assessment)** | Multi-dimensional framework for automated RAG evaluation covering faithfulness, relevancy, precision, and recall. |
| **Recall** | Proportion of actual positives correctly identified; in retrieval contexts, measures coverage of relevant content. |
| **Retrieval** | Step that selects candidate passages relevant to a query from a vector database. |
| **Retrieval Policy** | The decision rules governing how a RAG system determines relevance, selects context, and applies thresholds. |
| **Retrieval-Augmented Generation (RAG)** | Approach combining retrieval of external evidence with generation conditioned on retrieved context. |
| **Retriever** | Component that selects candidate passages relevant to a query using similarity search. |
| **RGB (Retrieval-Augmented Generation Benchmark)** | Benchmark evaluating RAG under noise robustness, negative rejection, information integration, and counterfactual conditions. |
| **RLHF (Reinforcement Learning from Human Feedback)** | Training approach using human preferences to align model outputs with desired behavior. |
| **RoBERTa** | Robustly optimized BERT pretraining approach used in some BERTScore implementations. |
| **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** | Overlap-based metric family measuring n-gram and subsequence similarity to reference text. |
| **ROUGE-1** | Unigram overlap variant of ROUGE. |
| **ROUGE-1 F** | Unigram overlap F-score combining precision and recall. |
| **ROUGE-2** | Bigram overlap variant of ROUGE. |
| **ROUGE-2 F** | Bigram overlap F-score combining precision and recall. |
| **ROUGE-L** | Longest common subsequence variant of ROUGE capturing structural alignment. |
| **ROUGE-L F** | LCS overlap F-score combining precision and recall. |
| **Score Mode / Score Retriever** | Retrieval mode applying minimum similarity threshold before selecting passages (ScoreThresholdRetriever). |
| **ScoreThresholdRetriever** | LangChain retriever component that filters passages below a specified similarity threshold before selection. |

| | |
|---|---|
| **SCPDx (Smart Crop Production Data Exchange)** | Platform combining blockchain and IPFS for secure data management; parent platform of PaSSER. |
| **Semantic Similarity** | Similarity of meaning measured using embeddings or contextual token representations. |
| **SentencePiece** | Tokenization framework used in Llama 2 7B for efficient text handling. |
| **Sentence-Transformers** | Framework for computing dense vector representations of sentences and paragraphs. |
| **SFT (Supervised Fine-Tuning)** | Training approach using labeled examples to adapt pre-trained models. |
| **Similarity Score** | Numeric score estimating relevance between query and passage, typically cosine similarity. |
| **Similarity Threshold** | Minimum similarity score required for a passage to be eligible for retrieval. |
| **Single-Page Application (SPA)** | Web client running in browser communicating via APIs; architecture of PaSSER frontend. |
| **Sliding Window Attention (SWA)** | Attention mechanism limiting computation to fixed local window for efficient long-sequence processing. |
| **Smart Contract** | Self-executing code on blockchain; llmtest contract stores evaluation results in PaSSER. |
| **Spearman Correlation (ρ)** | Rank-based association measure used in correlation matrices for non-linear relationships. |
| **T-CPS (Threshold-aware Composite Performance Score)** | Stability-adjusted score: $T\text{-}CPS = \mu \times (1 + \alpha \times (1 - CV)) - \beta \times CV^2$, with $\alpha = 0.1$, $\beta = 0.05$. |
| **Temperature** | Generation parameter controlling randomness; lower values produce more deterministic outputs (0.2 in experimental configuration). |
| **TF-IDF (Term Frequency-Inverse Document Frequency)** | Term weighting scheme elevating terms frequent within documents but rare across corpora. |
| **Threshold Sweep** | Systematic evaluation across predefined threshold range (0.50–0.95 in 0.05 increments). |
| **Token** | Basic unit of text processed by models (subword or word-piece units). |
| **Tokenizer** | Component that converts text into tokens; affects efficiency, multilingual handling, and robustness. |
| **Top-k (K)** | Number of highest-scoring passages retrieved per query regardless of absolute similarity (locked: K = 100 after Phase I). |
| **Transformer** | Attention-based neural architecture using self-attention mechanisms, underpinning modern LLMs. |
| **TREC** | Text REtrieval Conference; benchmark suite for retrieval and question answering evaluation. |

| | |
|---|---|
| **Ubuntu Server** | Linux operating system used for CPU-only inference environment (128 GB RAM configuration). |
| **Vector Database** | Storage optimized for similarity search over embedding vectors (ChromaDB in experimental configuration). |
| **Vector Similarity Search** | Retrieval over embeddings using similarity metrics such as cosine similarity. |
| **Vector Space Model (VSM)** | Information retrieval model representing documents and queries as vectors for geometric similarity computation. |
| **VectorStoreRetriever** | LangChain retriever component that selects top-k passages by similarity without threshold filtering. |

# INTRODUCTION

Transformer-based language models [1] scaled to billions of parameters, commonly termed Large Language Models (LLMs), have advanced Natural Language Processing (NLP) across machine translation, summarization, question answering, and dialogue [2]. These models generate fluent text, yet their reliability is limited in settings that require factual correctness, current knowledge, or verifiable sourcing [3]. When answers must be grounded in external evidence rather than inferred from parametric knowledge, purely generative architectures frequently produce unsupported or outdated content [4]. Retrieval-Augmented Generation (RAG) addresses this limitation by separating evidence retrieval from text generation [5].

Instead of relying exclusively on parametric knowledge, RAG retrieves evidence from an external corpus and conditions generation on the retrieved context. This design enables knowledge updates through corpus refresh rather than model retraining and supports evidence-grounded responses when sources are available. However, the effectiveness of grounding depends on the retrieval policy: how relevance is determined, what context is selected, and how the similarity threshold and ranking strategy shape the evidence presented to the generator [6]. It also depends on the generator's capacity to use retrieved context consistently rather than overriding it with unsupported content. Evaluating such systems therefore requires assessing not only generation fluency but also how retrieval behavior and context selection shape factuality and completeness.

## Relevance of the Topic

RAG systems are increasingly applied in domains where factual accuracy and traceability directly influence decision quality, including healthcare [7] and legal research [8]. In such settings, unsupported statements carry practical consequences: a misattributed clinical guideline or an unverifiable legal precedent can compromise downstream decisions. Retrieval failures, where relevant evidence exists but is not surfaced—can negate the intended benefits of grounding.

Although some LLMs now incorporate web browsing capabilities, such functionality does not inherently provide reproducible provenance under controlled evaluation conditions. Web sources vary in permanence, access conditions are not always documented, and the criteria by

which content is selected or retained may be opaque—factors that complicate systematic comparison across experimental runs. Parametric LLMs also remain prone to producing fluent but incorrect outputs when evidence is missing, conflicting, or weakly linked to the query, and they frequently provide limited or opaque attribution for individual claims [3], [9].

## Research Motivation

Although surveys describe rapid growth in RAG architectures and pipelines, they also highlight fragmentation in configurations and evaluation practices, which limits comparability and informed deployment decisions [10]. Three deficiencies motivate the research:

***Deficiency 1: Threshold-aware evaluation.*** Similarity threshold determines whether retrieved documents are included in the generation context and directly influences retrieval precision and recall [6], [11]. In practical deployments, similarity threshold selection is not merely an implementation detail: it governs whether a system supplies insufficient context (leading to missing evidence and incomplete answers) or excessive context (introducing irrelevant passages that can distract generation and degrade factual alignment). Similarity threshold choices also affect computational cost by controlling context size and downstream processing. A threshold-aware evaluation procedure is therefore necessary to characterize retrieval selectivity systematically and to support evidence-based configuration choices. Retrieval selectivity is operationalized through similarity thresholds varied under controlled conditions, enabling systematic identification of threshold-sensitive performance patterns across datasets and models.

*Note: "Similarity threshold" refers to the minimum cosine similarity score required for a retrieved passage to be included in the generation context. This parameter is also referred to as a "relevance threshold" or "selectivity threshold" in some literature; in the following chapters "similarity threshold" or just "threshold" is used for consistency.*

***Deficiency 2: Reproducibility infrastructure.*** RAG pipelines introduce multiple interacting configuration layers: corpus preprocessing, chunking strategy, embedding model choice, index construction, retrieval settings, generation settings, and evaluation logic. Because these layers interact, independent verification becomes difficult when configurations are not captured

precisely and reported in a complete and comparable form [12], [13]. When configurations are incompletely specified, results cannot be independently verified, and apparent improvements may be attributable to hidden differences rather than the intended experimental variable. This deficiency is especially acute in threshold-sensitive studies, where small changes in retrieval configuration can alter the evidence presented to the model. A reproducibility infrastructure is therefore required to record and preserve the complete run context and outputs in a form that supports independent verification and cross-study comparison.

***Deficiency 3: Practical guidance for open-source deployments.*** Organizations that require local or on-premises deployment due to security, data governance, or cost constraints must rely on open-source LLMs and locally controlled pipelines. However, comparative evidence that links similarity threshold sensitivity in retrieval, generation quality, and deployment feasibility under consistent experimental conditions remains limited [10], [14]. Without such evidence, model selection and retriever configuration are frequently guided by informal benchmarks or mismatched assumptions about retrieval behavior across systems. Practical guidance is therefore needed to make threshold sensitivity visible and comparable across open-source deployment candidates.

## Research Aim

The aim of the dissertation is to develop an evaluation framework for Retrieval-Augmented Generation that supports evidence-based retrieval configuration decisions for RAG systems with open-source LLMs, with particular focus on similarity threshold configuration.

## Research Questions

Building on the identified deficiencies, three research questions are addressed:

- **RQ1**: Does varying the similarity threshold produce measurable changes in generation quality?
- **RQ2**: Do similarity threshold effects differ across language models?
- **RQ3**: Do comparable similarity threshold ranges hold across knowledge domains?

Each question corresponds to a specific experimental phase: RQ1 is addressed through systematic threshold variation in Phases II–IV, RQ2 through cross-model comparison in Phases

III–IV, and RQ3 through cross-domain analysis comparing agricultural and biodiversity corpora in Phase IV. Phase I provides system demonstration and runtime profiling, establishing baseline platform functionality prior to threshold experimentation.

## Objectives

Four objectives are pursued:

***Objective 1: Define and implement the core components of the evaluation framework*** by integrating three layers: **(a)** a threshold-aware evaluation procedure with composite scoring, **(b)** the Performance Assessment System for Similarity Evaluation and Retrieval (PaSSER) platform [15] providing reproducibility infrastructure with blockchain-based provenance logging, and **(c)** a controlled experimental design producing comparative threshold-aware evidence across models and domains.

***Objective 2: Establish model selection criteria.*** Define selection criteria aligned with local deployment feasibility, licensing constraints, and computational requirements, including profiling of selected models with respect to context window size and decoding settings.

***Objective 3: Define metric selection and computation procedures.*** Select metrics aligned with the evaluation constructs of lexical overlap, semantic similarity, fluency, accuracy, and language modeling, and implement metric computation consistently across models and experimental conditions.

***Objective 4: Conduct controlled testing and analysis.*** Prepare domain corpora and question-answer datasets with specified preprocessing and retrieval configurations; execute controlled evaluations under systematic parameter variation, including similarity threshold sweeps; and aggregate results to interpret outcomes with respect to retrieval selectivity, generation quality, and reproducibility, producing practical guidance for model and threshold selection.

**Relationship between research questions and objectives.** Research Questions RQ1-RQ3 are empirical and examine how similarity threshold configuration affects RAG performance across models and domains. Objective 1 defines the core components of the evaluation framework, including the evaluation procedure, the reproducibility infrastructure, and the

controlled experimental design. Objectives 2-4 operationalize this framework through model selection, metric definition and computation, and controlled testing and analysis. Together, these objectives provide the basis for answering RQ1-RQ3 and for deriving practical guidance for model and threshold selection.

Table I.1 summarizes the mapping between the identified deficiencies, research questions, objectives, and resulting contributions.

*Table I.1 Mapping of Deficiencies, Research Questions, Objectives, and Contributions.*

| Deficiency | Research Question(s) | Objective(s) | Chapter(s) | Contribution | Framework Layer |
|---|---|---|---|---|---|
| D1: Threshold-aware evaluation | RQ1, RQ2, RQ3 | Obj 1 (evaluation workflow and threshold-aware retrieval), Obj 3, Obj 4 | Ch. 3–4 | C1 | Evaluation Procedure |
| D2: Reproducibility infrastructure | — | Obj 1 (provenance logging and blockchain recording) | Ch. 2 | C2 | Infrastructure |
| D3: Practical guidance for open-source deployments | RQ1, RQ2, RQ3 | Obj 1, Obj 2, Obj 3, Obj 4 | Ch. 3–4 | C3 | Evidence |

** Objective 1 contributes to all three deficiencies by defining the threshold-aware evaluation procedure (D1), implementing provenance logging and blockchain recording (D2), and establishing the controlled experimental design that produces comparative evidence (D3). Objectives 2-4 operationalize specific components of this framework.*

Three scientific-applied contributions together constitute the evaluation framework.

**The Evaluation procedure layer (C1)** introduces a threshold-aware evaluation procedure incorporating Composite Performance Score, Threshold-aware Composite Performance Score, and Balance Score for characterizing retrieval selectivity across similarity threshold settings [16], [15].

**The Infrastructure layer (C2)** implements reproducibility infrastructure through the PaSSER platform, combining blockchain-based provenance logging with complete configuration capture [17], [15].

**The Evidence layer (C3)** produces practical guidance for open-source RAG deployments, grounded in comparative empirical evidence linking similarity threshold sensitivity, generation

quality, and deployment feasibility across seven models in the 7-8 billion parameter range under controlled experimental conditions [16], [18].

## Structure

The dissertation consists of an **Introduction**, **Five chapters**, a **Conclusion**, **appendices**, and a **bibliography**.

**Chapter 1** establishes the research foundations and reviews related work on RAG architectures, evaluation practices, and reproducibility challenges, positioning Deficiencies D1–D3 within the relevant literature.

**Chapter 2** presents the infrastructure of the PaSSER platform. It describes the workflow, the configuration of different settings, the automated testing process, and blockchain-based provenance logging.

**Chapter 3** specifies the model selection rationale and defines the evaluation metrics and computation procedures applied in cross-model assessment.

**Chapter 4** reports empirical results from controlled testing across the agriculture and biodiversity datasets, analyzing similarity threshold sensitivity, model performance and cross-domain comparison.

**Chapter 5** discusses the research questions and scientific-applied contributions, addresses limitations, and outlines future research directions.

The **Conclusion** summarizes the main results.

The back matter includes Supporting Publications, a Citation Record, a Summary of Project Participation, Acknowledgements and a Declaration of originality of the results.

# CHAPTER 1: RETRIEVAL-AUGMENTED GENERATION

LLMs produce fluent text but remain constrained by static parametric knowledge; outputs may be ungrounded, temporally outdated, or difficult to trace. RAG addresses these limitations by incorporating external evidence at inference time. The development of RAG is traced from foundational work in information retrieval (IR) [11] and natural language processing (NLP) [19] through contemporary modular architectures, establishing context for the evaluation framework.

RAG architectures are categorized according to the challenges addressed (retrieval precision, hallucination reduction, domain specialization, and interpretability), and documented failure modes are reviewed. Existing evaluation approaches are examined to identify limitations in current assessment practice.

Research gaps corresponding to Deficiencies D1–D3 are then identified.

## 1.1 Foundational Developments

RAG emerges from a long convergence of IR and NLP. This section traces that convergence through four strands that shaped contemporary RAG systems: (1) basic indexing and data organization, (2) formal IR evaluation and early IR–NLP fusion, (3) advanced semantic retrieval and NLP methods, and (4) large-scale IR–NLP integration with modern embedding-based approaches [20]. Table 1.1 summarizes these categories.

***Table 1.1*** *Key Technological Milestones Contributing to the Development of RAG. Adapted from [20].*

| Category | Key Advancements | Contribution to RAG Development |
|---|---|---|
| Basic Indexing and Data Organization | Memex, Statistical Text Analysis, KWIC Indexing | Laid the groundwork for efficient large-scale retrieval of information, a fundamental requirement for RAG systems to access and utilize external knowledge. |
| Formal IR Evaluation and Early IR–NLP Fusion | Cranfield Studies, BASEBALL System | Established methods for evaluating retrieval effectiveness and demonstrated early attempts to use natural language for querying, paving the way for more sophisticated query handling and evaluation in RAG. |
| Advanced Semantic Retrieval and NLP Methods | TF-IDF, Vector Space Models, Word2Vec | Enabled retrieval based on semantic similarity rather than just keywords, and provided methods for understanding the meaning of text, which are crucial for selecting relevant context to augment generation in RAG. |
| Large-Scale IR–NLP Integration and Modern Embeddings | IBM Watson, Transformers, Dense Passage Retrieval, GPT Series | Showcased the integration of retrieval and generation at scale, and provided the powerful embedding techniques and fluent generative models that are the core components of modern RAG systems. |

### 1.1.1 Basic Indexing and Data Organization (1945–1965)

The first development phase established fundamental techniques for organizing and accessing textual information at scale. Memex [21] pioneered associative trails, prefiguring modern hypertext navigation by enabling context-rich connections among documents. Statistical text analysis [22] advanced beyond manual indexing through word-frequency counting and distributional pattern analysis, enabling objective document characterization based on quantifiable evidence.

Key Word in Context (KWIC) indexing [23] automated extraction of snippets surrounding keywords, supporting rapid relevance assessment. Physical retrieval systems of this period, including mechanical selectors and edge-notched card systems [24], [25], introduced systematic document filtering methods that presaged automated retrieval.

### 1.1.2 Formal Evaluation and Elementary Language Processing (1960–1975)

The second phase formalized retrieval evaluation and incorporated elementary natural language processing. The BASEBALL system [26] applied pattern-matching and parsing rules to interpret domain-specific queries, demonstrating that natural language input could drive search operations. Concurrently, Cranfield studies [27] established standardized performance metrics—notably precision and recall [11]—transforming retrieval from ad hoc practice into systematic engineering.

These metrics quantified systems' capacity to retrieve relevant documents while excluding irrelevant ones, becoming benchmarks that persist in contemporary RAG evaluation. Linking retrieval with natural language interfaces and systematic quality measurement laid groundwork for architectures combining retrieval with generation.

### 1.1.3 Semantic Understanding and Advanced Retrieval (1970–2000)

Third phase introduced statistical approaches to relevance estimation. Term Frequency-Inverse Document Frequency (TF-IDF) [28] provided principled term weighting by elevating terms frequent within documents but rare across corpora. Salton's Vector Space Model (VSM) [29] reconceptualized retrieval by representing documents and queries as high-dimensional vectors, enabling similarity computation through geometric operations rather than exact keyword matching.

Domain-specific systems such as The Lunar Science Natural Language Information System (a.k.a. LUNAR) [30], [31] demonstrated how rule-based methods with explicit grammars could support technical queries in constrained fields—an early strand of domain-focused interpretation that RAG systems revisit when adapting to specialized terminology.

The probabilistic retrieval framework [32] formalized relevance through statistical language models, establishing theoretical foundations for ranking algorithms. Latent Semantic

Analysis (LSA) [33] addressed synonym and polysemy challenges through dimensionality reduction. These advances collectively shifted retrieval from surface-level keyword matching toward semantic similarity assessment, directly prefiguring dense embedding-based retrieval in contemporary RAG.

### 1.1.4 Large-Scale Integration and Modern Embeddings (2000–2020)

The fourth phase witnessed large-scale IR-NLP integration enabled by modern embedding techniques and neural architectures. IBM Watson's 2011 Jeopardy! victory [34], [35] demonstrated practical integration of retrieval and question-answering at scale through the DeepQA architecture (Figure 1.1). Despite successful performance, Watson encountered scalability and real-time integration challenges, prompting exploration of more modular retrieval-generation designs.

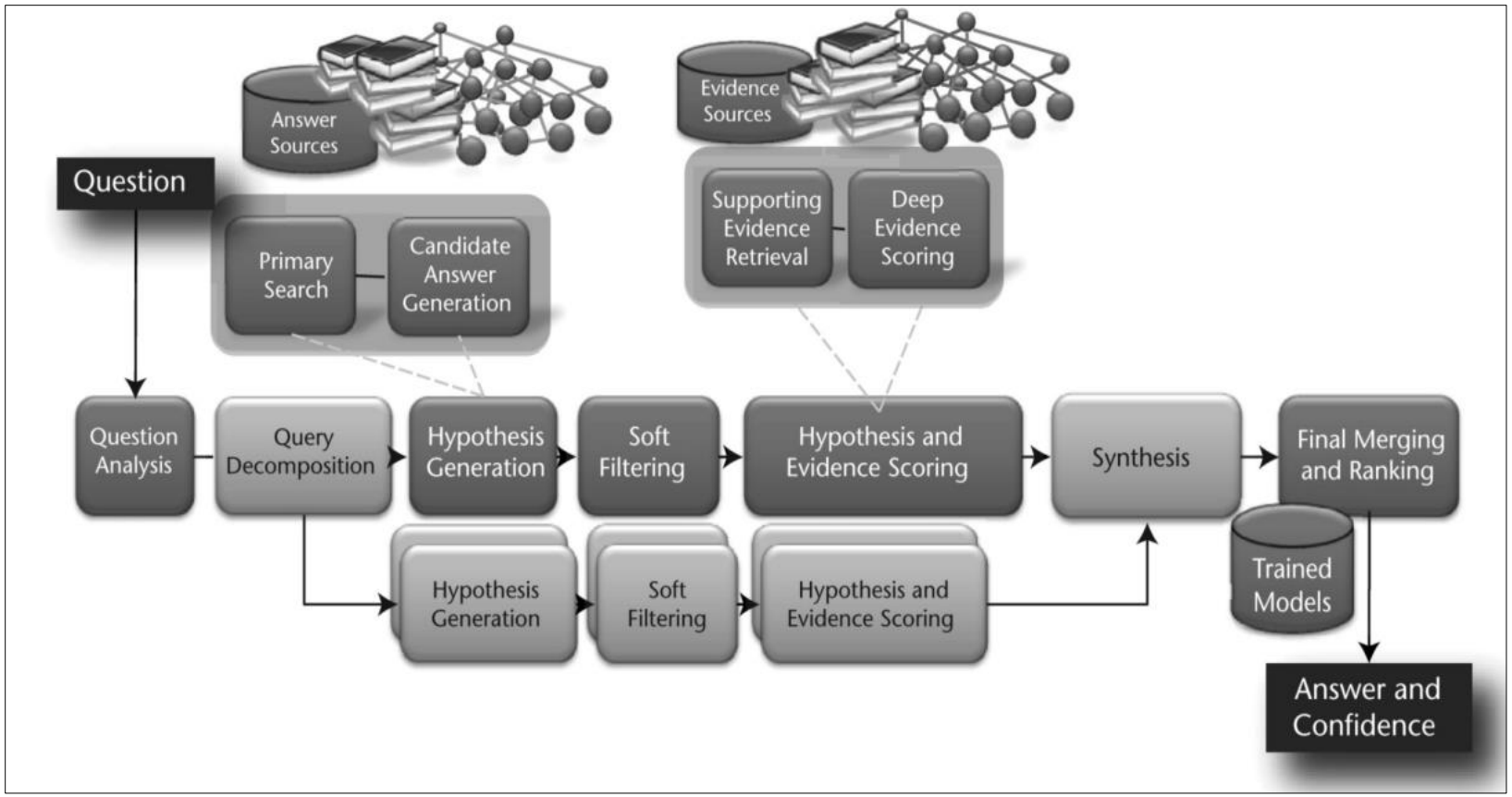


***Figure 1.1 DeepQA high-level architecture.*** *IBM Watson's question-answering pipeline processes a natural language question through Question Analysis and Query Decomposition, enabling parallel retrieval from Answer Sources and Evidence Sources. Candidate answers are generated through Primary Search and Candidate Answer Generation, then filtered via Soft Filtering. Supporting evidence undergoes Deep Evidence Scoring before being aligned with hypotheses in Hypothesis and Evidence Scoring. Multiple processing streams converge at Synthesis, with Final Merging and Ranking applying trained models to produce the final answer with a confidence score. This architecture established key patterns later adopted by RAG systems: query decomposition, multi-source retrieval, evidence-grounded generation, and confidence-weighted ranking. Reproduced from [34].*

Word2Vec [36] introduced efficient continuous vector representations capturing semantic relationships through geometric operations, enabling similarity based retrieval that generalizes across lexical variations. Bidirectional Encoder Representations from Transformers (BERT) [37] achieved bidirectional context understanding by training transformers to predict masked tokens given surrounding context. Dense Passage Retrieval (DPR) [38] placed queries and documents in shared high-dimensional space, matching content based on semantic affinity rather than keyword alignment. The Generative Pre-trained Transformer (GPT) series [39], [40] achieved unprecedented fluency through large-scale transformer architectures, completing the generative component of modern RAG systems.

These phases illustrate evolution from rigid keyword matching to semantically rich, context-aware systems, laying the technical foundation for RAG as a response to traditional language model limitations.

## 1.2 The Emergence of RAG

The RAG framework was formally introduced in 2020 by Facebook AI Research (now Meta AI) in the foundational study "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" [5]. It integrates IR directly into the generative process, addressing key limitations of traditional LLMs: reliance on static training data, lack of source attribution, and tendency to produce fluent but inaccurate outputs—commonly known as hallucination [41], [42].

### 1.2.1 Architectural Components

Unlike conventional models encoding all knowledge within parameters, RAG separates retrieval and generation. At inference time, it dynamically retrieves relevant external documents—typically from Wikipedia or domain-specific corpora—which condition the response. This approach improves factual accuracy, enables transparency, and supports domain adaptation without retraining.

The retriever component employs DPR [38], encoding queries and documents into dense embeddings using BERT-based transformers [37]. Relevant passages are identified via Maximum Inner Product Search (MIPS) [43], which retrieves documents whose embedding vectors have the

highest dot product with the query embedding. Practical implementations employ approximate nearest-neighbor algorithms [44], [45] to maintain efficient search at scale.

The generator component is based on the Bidirectional and Auto-Regressive Transformers (BART) architecture [46], combining a bidirectional encoder with an auto-regressive decoder. This hybrid design captures contextual relationships while generating fluent responses token-by-token. BART operates within a sequence-to-sequence framework—originally developed for machine translation [47]—transforming the input sequence (query + retrieved passages) into coherent, evidence-grounded output.

Figure 1.2 illustrates the RAG architecture. The retriever encodes the query into a dense vector and retrieves semantically relevant documents via MIPS; the generator conditions its output on the retrieved passages.

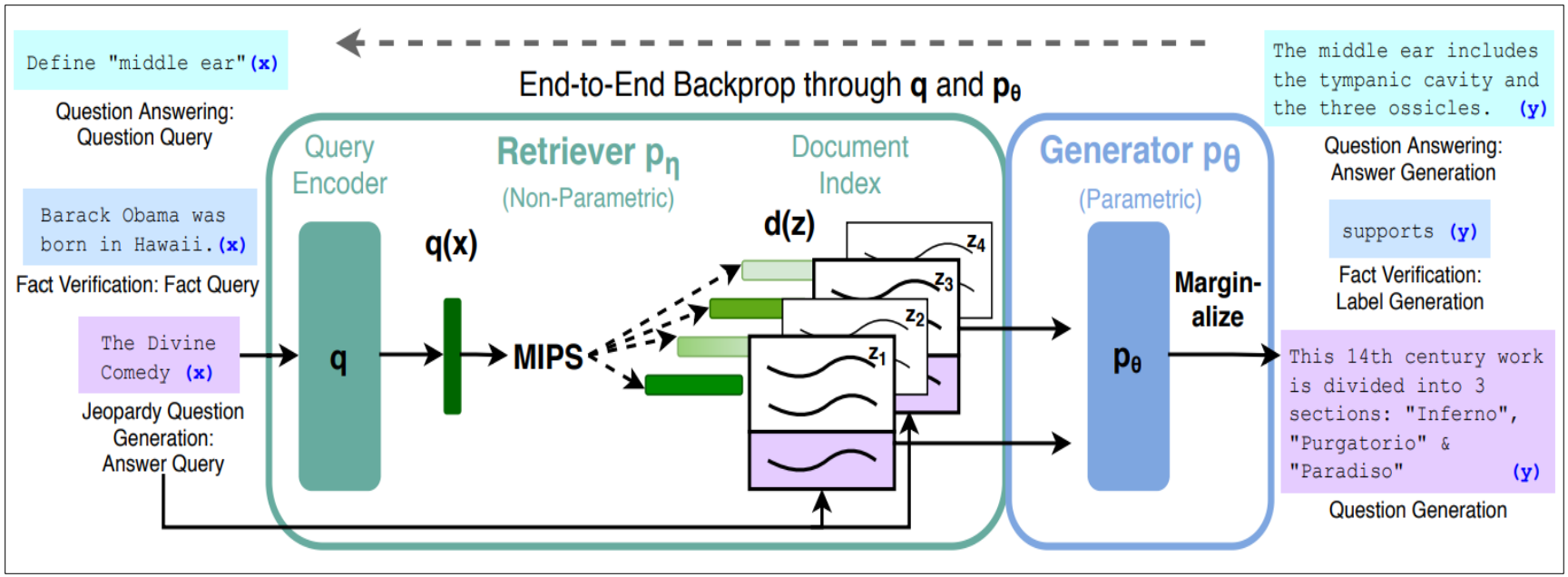


***Figure 1.2 Overview of the RAG architecture***. *The diagram illustrates the end-to-end RAG pipeline introduced by Lewis et al. An input query x (left side) represents different task types: question answering ("Define 'middle ear'"), fact verification ("Barack Obama was born in Hawaii"), or question generation ("The Divine Comedy"). The Query Encoder transforms the input into a dense vector representation q(x). The Retriever p_η (non-parametric) performs Maximum Inner Product Search (MIPS) against a Document Index containing pre-encoded document vectors d(z), retrieving the top-k most semantically similar passages ($z_1$, $z_2$, $z_3$, $z_4$). Retrieved passages are passed to the Generator p_ϑ (parametric), which applies Marginalize to weight document contributions and generate the output y. Example outputs correspond to input task types: answer generation ("The middle ear includes the tympanic cavity and the three ossicles"), fact verification label ("supports"), or generated question ("This 14th century work is divided into 3 sections..."). The dashed arrow indicates end-to-end backpropagation through both the query encoder and generator, enabling joint training. This architecture established the foundational RAG pattern: encode, retrieve, condition, generate. Reproduced from [5].*

### 1.2.2 Training and Integration

RAG explicitly trains the combined retrieval-generation system end-to-end for knowledge-intensive NLP tasks. The joint training enables the retriever to learn which documents are most useful for the generator, while the generator learns to effectively leverage retrieved context. This integrated approach distinguishes RAG from earlier pipeline systems where retrieval and generation operated independently.

Initial evaluations demonstrated RAG's effectiveness on knowledge-intensive tasks including open-domain question answering (Natural Questions [48], TriviaQA [49]), fact verification (FEVER [50]), and abstractive question answering (MS MARCO NLG [51]). RAG achieved state-of-the-art results while requiring significantly fewer parameters than purely parametric models, demonstrating that separating knowledge storage (retrieval corpus) from reasoning (generative model) offers efficiency advantages [5] .

## 1.3 RAG Innovations and Extensions

Following its initial release, RAG underwent substantial enhancements aimed at improving retrieval precision, generative accuracy, interpretability, and computational efficiency. As the field matured, research shifted from foundational development to focused improvements, each addressing specific limitations. Table 1.2 presents a functional overview of advancements, organized by the core challenges they were intended to solve [20]. Section 1.3.8 then examines retrieval configuration and similarity threshold selection—a factor that mediates retrieval outcomes but lacks systematic characterization in published evaluations.

***Table 1.2*** *Functional Categorization of Recent RAG Advancements. Repoduced from [20].*

| Focus Area | Objective |
|---|---|
| Architectural Efficiency and Scalability | Reduce computational cost, improve inference speed, and support real-time applications |
| Data-Centric Optimization | Improve training quality through noise reduction, sampling, and data selection techniques |
| Iterative Retrieval and Self-Refinement | Introduce mechanisms for multi-step reasoning, response revision, or feedback-based retrieval |
| Knowledge Integration and Multimodal Extensions | Combine symbolic and neural retrieval to access diverse knowledge representations; extend retrieval to additional modalities |
| Domain Adaptation and Specialization | Enable RAG systems to perform well in specialized domains or narrow knowledge fields |
| Factual Verification and Grounding | Reduce hallucinations and improve transparency by anchoring outputs in verifiable sources |
| Few-Shot and Low-Resource Enhancement | Improve generalization with limited training data through retrieval-enhanced few-shot learning |

### 1.3.1 Architectural Efficiency and Scalability

Architectural efficiency emerged as an early concern as RAG systems scaled beyond research prototypes. The Fusion-in-Decoder (FiD) architecture [52] enabled synthesis from multiple retrieved passages but introduced computational overhead as decoders attended uniformly to all tokens. FiD-Light [53] addressed this through selective attention mechanisms that dynamically filter and prioritize relevant segments, demonstrating improved performance on the Knowledge Intensive Language Tasks (KILT) benchmark [54] while reducing unnecessary computation.

A parallel strand of work reconsidered how retrieved knowledge is organized. Rather than treating documents as flat text, LightRAG [55] and GraphRAG [56] introduced graph-based representations that encode entity-relationship structures. LightRAG uses graph topology to guide retrieval toward semantically connected content, which proves particularly effective for multi-hop reasoning tasks requiring evidence synthesis across sources. GraphRAG extends this

approach by constructing knowledge graphs from unstructured text using LLM-based entity extraction (Figure 1.3), enabling hierarchical community detection and cross-document synthesis. However, graph-based approaches introduce indexing overhead and maintenance complexity; their performance on dynamic, frequently-updated corpora remains less clearly characterized than on static analytical collections.

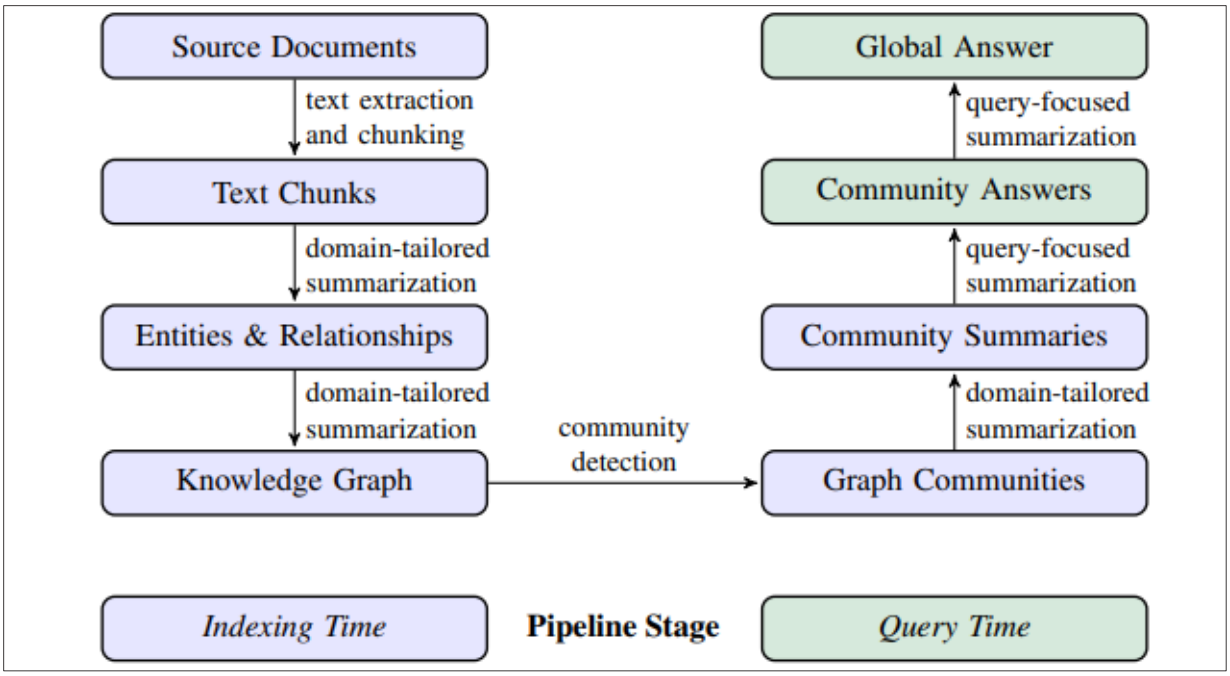


***Figure 1.3 GraphRAG pipeline.*** *Source documents are processed through entity and relationship extraction to construct a knowledge graph. The Leiden algorithm [57] performs hierarchical community detection, grouping semantically related entities into clusters at multiple levels of granularity. At query time, relevant communities are identified and summarized using query-focused summarization, producing context that captures cross-document relationships rather than isolated passages. This graph-based approach enables synthesis across documents for complex queries requiring multi-hop reasoning, though it introduces indexing overhead compared to flat retrieval. Reproduced from [56].*

Self-optimization represents a third direction. Auto-RAG [58] monitors its own performance during inference and adjusts retrieval and generation components through self-supervised feedback loops, enabling adaptation without manual intervention. This real-time self-adjustment addresses a practical limitation of fixed pipelines, which require labor-intensive reconfiguration when deployed across different domains or corpora.

### 1.3.2 Data-Centric Approach

While architectural innovations focus on system design, a complementary research strand addresses the quality of data used during training and inference. Data-centric methods enhance model learning by selecting, filtering, and weighting examples to reduce noise—a concern that becomes acute in multi-task and low-resource scenarios, where inconsistent data hampers generalization.

Relevance Sampling [6] exemplifies this approach by assigning confidence scores to training examples based on model uncertainty, retrieval relevance, or internal consistency. Low-confidence examples are excluded, allowing models to learn from cleaner, more informative data. Empirical results indicate that this filtering improves generalization and reduces hallucination across knowledge-intensive tasks.

A related strategy operates at generation time rather than training time. Speculative RAG [59] introduces a two-stage process: an initial draft is produced speculatively, then refined through retrieval-informed verification that re-checks evidence and corrects or strengthens claims. This approach filters out unsupported content before final output, improving factual grounding without requiring changes to the underlying model. Together, these techniques reflect a broader shift toward quality-aware pipelines that curate inputs and verify outputs rather than relying solely on architectural capacity.

### 1.3.3 Iterative Retrieval and Self-Refinement

Standard RAG retrieves context once before generation, but this single-shot approach can fail when initial retrieval is incomplete or when complex queries require evidence synthesis across multiple reasoning steps. Several approaches address this limitation through iterative retrieval mechanisms.

Self-RAG [60] introduces reflection tokens that allow the model to assess passage relevance, evidential support, and response utility during generation, internalizing retrieval decisions rather than relying solely on external orchestration. Corrective RAG (CRAG) [61] introduces a lightweight retrieval evaluator that classifies retrieval outcomes as correct, incorrect, or ambiguous before generation, enabling adaptive correction but adding pipeline

complexity. Adaptive-RAG [62] routes queries to different retrieval strategies based on estimated complexity, improving the latency-quality trade-off when routing is accurate. Forward-Looking Active REtrieval augmented generation (FLARE) [63] monitors token-level confidence during generation and triggers retrieval when uncertainty rises beyond a threshold, interleaving retrieval with generation for multi-hop questions (Figure 1.4).

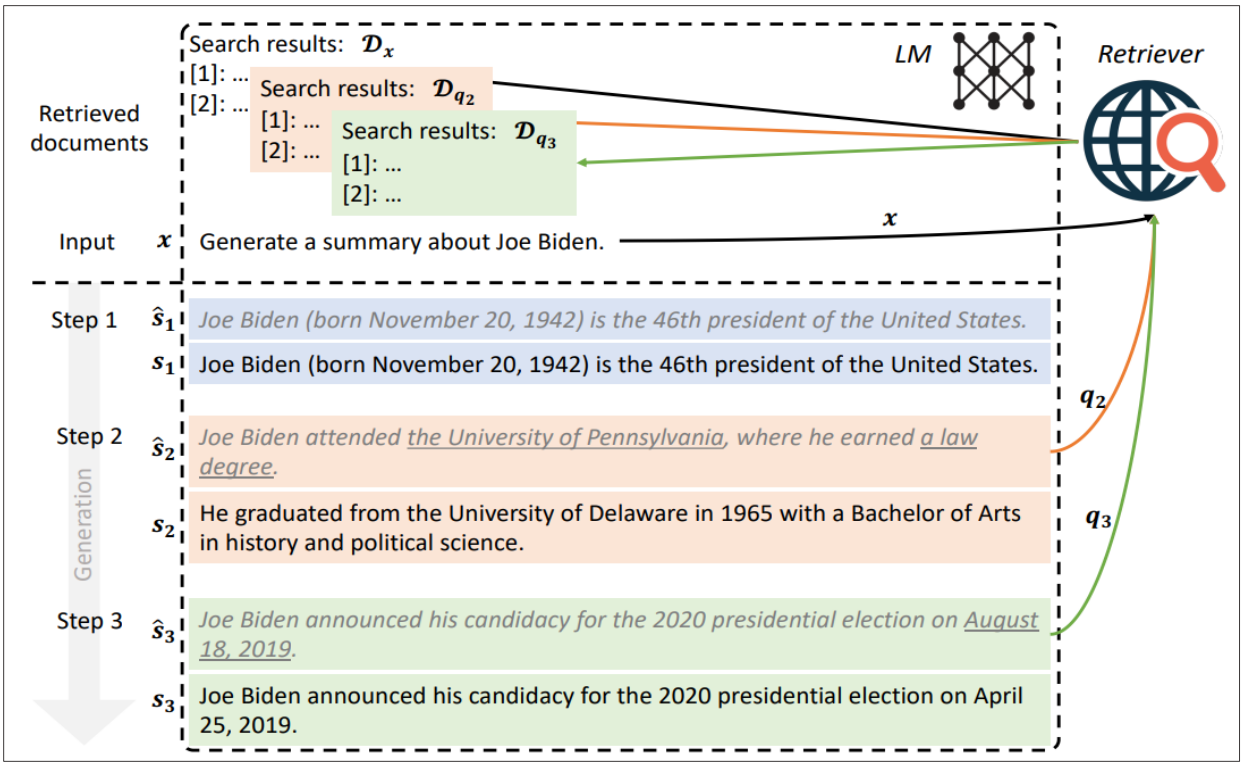


***Figure 1.4 FLARE iterative retrieval-generation process.*** *Unlike single-shot RAG, FLARE monitors token-level confidence during generation and triggers retrieval when uncertainty exceeds a threshold. When low-confidence tokens are detected (indicated by underlined text), generation pauses and the partial output serves as a retrieval query. Retrieved passages are incorporated into the context, and generation resumes with updated evidence. This forward-looking active retrieval enables dynamic evidence gathering for multi-hop questions, though effectiveness depends on confidence calibration and introduces additional latency. Reproduced from [63] .*

Named after the bird, RAVEN [64] is a retrieval-augmented encoder-decoder language model, embeds retrieval within the encoder-decoder attention mechanism, enabling simultaneous attention over query prompts and retrieved evidence for finer-grained evidence weighting than concatenation-based approaches.

These approaches share a common dependency on explicit decision gates—probability thresholds, confidence triggers, and routing boundaries—that control what evidence reaches the generator. The threshold configurations used by these systems are examined in Section 1.3.8.

### 1.3.4. Knowledge Integration and Multimodal Extensions

As RAG systems matured, researchers recognized that unstructured text corpora, while flexible, do not capture the precision and verifiability offered by structured knowledge sources. Many applications demand information organized by entities, attributes, and relationships—structures common in knowledge graphs, relational databases, and domain ontologies.

Hybrid retrieval approaches [65] address this by combining dense vector search with symbolic querying over structured data. These systems blend the semantic flexibility of neural retrieval with the precision and traceability of structured queries. Common techniques include structured queries [66] over knowledge graphs, metadata tagging, schema alignment, and post-retrieval reranking that elevates structured matches. By fusing symbolic and neural retrieval, RAG systems can draw from both unstructured text and structured knowledge bases, producing more grounded outputs—particularly beneficial in domains such as biomedicine, finance, and law where factual accuracy and auditability are paramount.

RAG has also been extended to multimodal settings, where retrieval operates over images, audio, or video segments using modality-specific encoders and cross-modal similarity spaces. VideoRAG [67] couples video understanding with retrieval over selected frames, while MuRAG [68] combines text and visual retrieval for open question answering. These extensions demonstrate the architectural flexibility of the retrieval-augmented paradigm, though they fall outside the text-based scope of the present evaluation and are not further examined.

### 1.3.5 Domain Adaptation and Specialization

General-purpose RAG systems trained on broad corpora such as Wikipedia perform well on open-domain benchmarks but often struggle when applied to specialized fields. Biomedical research, legal analysis, and technical documentation involve domain-specific terminology, citation conventions, and reasoning patterns that require targeted adaptation.

PaperQA [69] exemplifies domain-specific RAG design for scientific literature. The system implements search across paper databases including arXiv, PubMed, and Semantic Scholar, using Maximal Marginal Relevance (MMR) to ensure evidence diversity while maintaining relevance. Citation-aligned synthesis explicitly links generated claims to source papers, and LLM-based relevance scoring iteratively refines the evidence set by removing low-utility passages and retrieving additional sources when gaps are detected (Figure 1.5). Superior performance on PubMedQA [70] and LitQA [71] benchmarks demonstrates that domain-specific architectural adaptations can substantially outperform general-purpose RAG in specialized contexts.

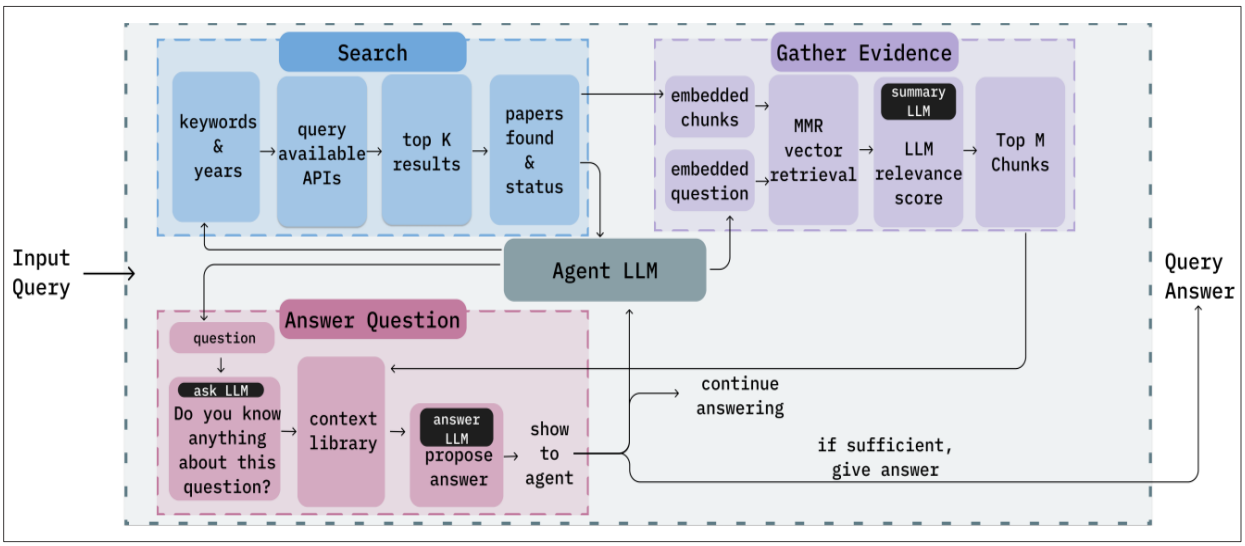


***Figure 1.5 PaperQA RAG for scientific literature.*** *The system searches academic databases (arXiv, PubMed, Semantic Scholar) and retrieves candidate papers using Maximal Marginal Relevance (MMR) to balance relevance with diversity. Retrieved passages undergo LLM-based relevance scoring, with low-utility passages removed and additional sources retrieved when gaps are detected. Citation-aligned synthesis links each generated claim to its supporting source paper, enabling verification of individual statements. This iterative refinement and explicit attribution address the traceability requirements of scientific question answering. Reproduced from [69].*

RaLLe [72] provides a modular environment for retriever-reranker experimentation, enabling systematic evaluation of different retrieval configurations in domain-specific contexts. This modularity supports the kind of controlled comparison that domain adaptation research requires.

Retrieval-Augmented Fine-Tuning (RAFT) [73] combines the advantages of RAG and fine-tuning by creating synthetic datasets for domain adaptation. The process involves generating

synthetic queries, relevant documents, and target responses, then fine-tuning the model on this synthetic dataset to align with domain knowledge. RAFT outperforms traditional RAG in specialized domains where terminology and reasoning patterns prove critical [73].

Domain adaptation strengthens relevance and terminological coverage, but it also raises expectations for traceable claims in regulated or evidence-sensitive settings. This concern has motivated explicit verification mechanisms in several RAG variants.

### 1.3.6 Factual Verification and Grounding

A persistent challenge for RAG systems is factual consistency—ensuring that generated outputs are traceable to verifiable sources and free from hallucinated or unsupported claims. This concern becomes particularly acute in high-stakes domains such as scientific research, journalism, medicine, and law, where ungrounded statements can carry serious consequences.

Standard RAG models typically cite entire documents or broad passages, which provides some traceability but makes verification laborious. ReClaim [74] addresses this limitation by enforcing sentence-level attribution in generated responses. Each factual statement in the output is explicitly linked to a supporting source sentence, enabling users to verify specific claims directly. By constraining generation to content grounded in verifiable evidence, ReClaim reduces hallucination and improves factual reliability. Empirical evaluations demonstrate that this fine-grained attribution outperforms baseline RAG models in source accuracy, making it particularly suited to domains where traceability is non-negotiable.

The grounding challenge connects directly to evaluation: assessing whether generated content aligns with retrieved evidence requires metrics that capture factual fidelity beyond surface-level fluency.

### 1.3.7 Few-Shot and Low-Resource Enhancement

Retrieval-augmented approaches offer particular advantages in few-shot and low-resource scenarios, where limited training data constrains purely parametric models. By leveraging external knowledge at inference time, RAG systems can generalize beyond their training distributions without requiring extensive fine-tuning on domain-specific examples.

Atlas [75] represents a landmark contribution in this area. The system jointly pre-trains a retriever and language model, enabling the retriever to identify documents that maximize downstream task performance rather than merely matching surface-level query similarity (Figure 1.6). This joint training produces retrieval behavior aligned with generation needs. On the Natural Questions benchmark, Atlas with 11 billion parameters outperformed PaLM with 540 billion parameters [75], demonstrating that effective retrieval can substitute for massive parameter scaling. The few-shot capabilities prove particularly striking: with only 64 training examples, Atlas achieved performance comparable to fully supervised baselines on several benchmarks.

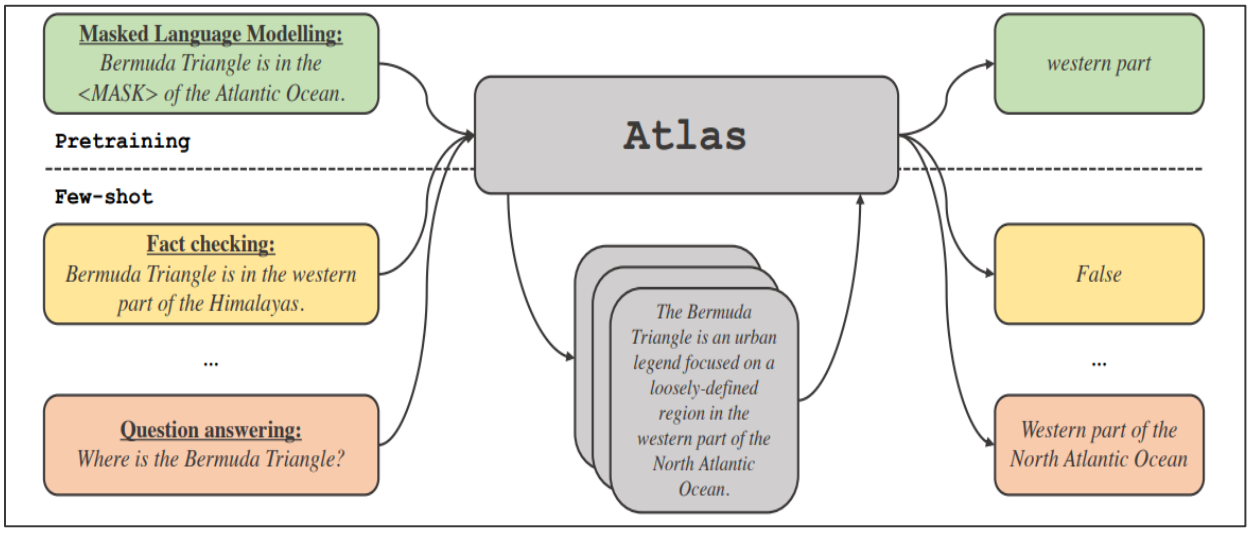


***Figure 1.6 Atlas retrieval-augmented few-shot learning.*** *Atlas jointly pre-trains a retriever and language model, enabling the retriever to identify documents that maximize downstream task performance rather than merely matching surface-level query similarity. The architecture encodes queries and documents into a shared embedding space; retrieved passages condition the generator through cross-attention. Joint training aligns retrieval behavior with generation needs, enabling strong few-shot performance—with only 64 training examples, Atlas matched fully supervised baselines on several benchmarks while using 50× fewer parameters than comparable purely parametric models. Reproduced from [75].*

Retrieval-Augmented Language Model Pre-Training (REALM) [76] introduced the paradigm of pre-training language models with a latent knowledge retriever. During pre-training, REALM learns to retrieve documents that help predict masked tokens, creating retrieval capabilities that transfer to downstream tasks without task-specific retrieval fine-tuning. This proves especially effective in low-resource settings where task-specific training data is scarce.

In-Context RALM [77] explores retrieval augmentation specifically for in-context learning. Rather than fine-tuning, the system retrieves relevant examples and documents to include in the prompt context, enabling few-shot task performance through careful example selection. The approach demonstrates that retrieval quality directly impacts in-context learning effectiveness—semantically similar retrieved examples yield substantially better few-shot performance than random selection.

*k*NN-LM [78] takes a different approach by augmenting language model predictions through interpolation with a nearest-neighbor distribution over cached representations. At inference time, the model retrieves similar contexts from a data store and combines their token distributions with the parametric model's predictions. This non-parametric augmentation improves performance without additional training and proves effective for domain adaptation, where the data store can be populated with domain-specific text.

These few-shot and low-resource techniques demonstrate that retrieval can compensate for limited training data and reduce dependence on massive parameter counts. While the focus is on standard RAG evaluation rather than few-shot scenarios, the underlying principle—that retrieval configuration materially affects generation quality—applies across settings.

The innovations surveyed in this section demonstrate growing sophistication in RAG research, from architectural efficiency and graph-based knowledge organization to iterative retrieval, domain adaptation, and factual grounding. Across these advances, RAG performance depends not only on model capability but also on how retrieval selectivity is configured. The next subsection summarizes common retrieval configuration controls and the role of similarity thresholds in determining what evidence reaches the generator.

### 1.3.8 Retrieval Configuration and Similarity Threshold Selection

The innovations surveyed in preceding sections share a common dependency: their effects are mediated by retrieval configuration decisions that determine what evidence is available to the generator. This subsection summarizes how retrieval selectivity is controlled across RAG implementations and highlights the limited evidence available for similarity threshold selection.

**Retrieval selection mechanisms.** RAG systems commonly use two mechanisms to control which retrieved passages enter the generation context. *Top-k* retrieval returns the *k* highest-scoring passages regardless of absolute similarity values. This guarantees a fixed number of passages but can include marginally relevant content when few strong matches exist. Threshold-based filtering returns only passages whose similarity scores exceed a minimum value. This allows variable context sizes and can return no passages when evidence is weak, but it may also return large context sets for broad queries.

Many deployments combine both mechanisms, applying similarity threshold filtering followed by *top-k* truncation (or using a minimum similarity gate inside *top-k*). The interaction between similarity threshold and *k* defines a configuration space that affects retrieval precision (how much retrieved content is relevant) and retrieval recall (how much relevant content is retrieved).

**Similarity threshold configurations in existing systems.** The iterative retrieval approaches described in Section 1.3.3 each incorporate threshold-like controls: Self-RAG [60] uses a delta gate controlling retrieval triggering frequency, FLARE [63] relies on confidence thresholds for uncertainty-driven retrieval, and CRAG [61] uses decision boundaries separating correct, incorrect, and ambiguous retrieval outcomes. In each case, similarity threshold values are typically chosen through local tuning on a development set or task-specific experiments rather than through broad sensitivity characterization.

In practice, many deployments rely on defaults and heuristics. Vector database and library retrievers often default to pure *top-k* selection, treating *k* as the primary selectivity parameter while ignoring whether absolute similarity scores indicate weak evidence. Some production systems have been observed to adopt a similarity threshold of 0.7 as a default, without systematic justification. A published industry case study in the banking domain showed that, under a baseline configuration with a fixed similarity threshold of 0.7, the false positive rate for some embedding models could reach 99% [79]. Heuristic similarity cutoffs are sometimes reported in practitioner guides, but these values are not transferable across embedding models, similarity functions, and corpora, and may invert meaning when a system returns distances rather than similarities.

**Similarity threshold characterization deficiency.** Despite the importance of retrieval selectivity, published evaluations commonly report results for a single retrieval configuration and do not show how performance changes as similarity thresholds vary. This has three consequences. **First**, practitioners lack evidence-driven guidance for threshold selection. **Second**, model comparisons performed under different selectivity settings can conflate model capability with configuration effects. **Third**, threshold behavior can vary with corpus properties such as vocabulary density, chunking choices, and query distributions, limiting transfer across domains.

Addressing this deficiency requires controlled variation of similarity thresholds across multiple LLMs using a consistent evaluation pipeline, measuring how lexical and semantic quality indicators change across the selectivity range.

## 1.4 Evaluating RAG

Evaluating RAG performance requires assessment beyond answer accuracy or lexical overlap [80]. Effective evaluation must determine whether retrieved evidence is relevant, whether generation remains grounded in that evidence, and whether system behavior is predictable as retrieval conditions vary [81], [14]. The following subsections review prominent evaluation frameworks—Retrieval-Augmented Generation Assessment (RAGAS) [81], the Retrieval-Augmented Generation Benchmark (RGB), TruLens [83], and the TREC 2024 RAG Track—and analyze their strengths and limitations with respect to threshold-sensitive evaluation. In this context, threshold sensitivity refers to how system performance changes as similarity thresholds governing retrieval selectivity are varied; frameworks that evaluate only fixed configurations cannot expose such dependencies.

In practice, evaluation frameworks differ not only in what they measure but also in what they assume is fixed. Some approaches treat retrieval as a static precondition and focus on judging the generated text, while others stress test failure modes such as missing or conflicting evidence. For threshold-sensitive analysis, this distinction matters: if retrieval configuration is held constant, evaluation results can describe average behavior, but cannot explain how performance changes when retrieval selectivity is tightened or relaxed.

### 1.4.1 RAGAS Framework

RAGAS [81] provides a multi-dimensional framework for automated RAG evaluation. It targets failure modes that commonly affect reliability through four dimensions: faithfulness, answer relevancy, context precision, and context recall.

Faithfulness evaluates whether the generated answer is supported by the retrieved context. Implementations commonly operationalize this through LLM-assisted claim checking or entailment-style verification, where statements in the answer are compared against retrieved passages to identify unsupported content.

Answer relevancy evaluates whether the response addresses the user query. RAGAS operationalizes this using semantic similarity and LLM-assisted judging to compare query intent and response content, penalizing tangential or off-target answers.

Context precision measures how much of the retrieved context is relevant, reflecting retrieval noise.

Context recall measures whether the retrieved context includes the information needed to answer the query. Computing context recall at scale typically requires reference annotations or LLM-based surrogate judgments that approximate human relevance assessments.

A practical implication of this design is that RAGAS can separate retrieval and generation concerns at the metric level: low context precision indicates noisy retrieval, while low faithfulness indicates that generation is insufficiently grounded even when relevant context is present. However, because several dimensions may rely on model-based judging, scores can vary with the choice of evaluator model and prompting assumptions. This makes RAGAS valuable for large-scale comparison under a fixed setup, but less suited for isolating retrieval-parameter effects across a threshold range.

RAGAS supports large-scale automated scoring, but it has limitations relevant to threshold-sensitive evaluation. Scores are typically reported for a fixed retrieval configuration, meaning performance changes across varying similarity thresholds are not characterized. Additionally, results can depend on the evaluator model used in LLM-as-judge components, and computing context recall at scale requires relevance annotations or judging assumptions that are not always available.

### 1.4.2 RGB Benchmark

The RGB [14] evaluates RAG systems under conditions designed to probe common deployment failures. Its conditions include noise robustness, negative rejection, information integration, and counterfactual robustness.

Noise robustness measures performance when irrelevant passages are included alongside relevant evidence. This condition tests whether the generator can identify and prioritize pertinent content while ignoring distractors that may dilute or mislead the response.

Negative rejection evaluates whether the system appropriately abstains when retrieved evidence is insufficient to answer the query. Systems that lack this capability tend to hallucinate plausible-sounding responses rather than acknowledging uncertainty.

Information integration assesses synthesis across multiple passages. This condition determines whether the generator can combine complementary evidence from separate sources into a coherent response, rather than relying on a single passage or producing fragmented output.

Counterfactual robustness tests behavior when retrieved passages contain conflicting information. This condition probes whether the system can detect inconsistencies and resolve or flag them, rather than arbitrarily selecting one version or blending contradictory claims.

Because RGB varies the evidence conditions rather than the retrieval gate itself, it is useful for diagnosing how generators behave when retrieval returns imperfect context, which is common in real deployments. Its task structure highlights whether a system can abstain, integrate multiple sources, or remain consistent under contradiction. However, RGB does not explicitly parameterize selectivity through similarity thresholds, so it cannot provide a sensitivity curve showing where performance changes as retrieval becomes more or less selective.

RGB provides diagnostic insight by explicitly stressing systems with noisy, missing, multi-source, and conflicting evidence scenarios. Empirical findings from RGB indicate that models struggle most with negative rejection, often generating responses despite insufficient evidence—a pattern that highlights the need for improved fact-verification mechanisms [14]. However, like RAGAS, RGB evaluations are typically conducted under fixed retrieval settings and do not expose how outcomes shift as retrieval selectivity changes.

### 1.4.3 The TREC 2024 RAG Track

The inaugural TREC 2024 RAG Track [82] targeted standardization and reproducibility for RAG evaluation. It was introduced in response to documented difficulties in replicating reported results and comparing systems evaluated under different conditions.

The track provides shared datasets, baselines, and evaluation scripts, enabling direct comparison across submissions under a consistent protocol. This shared infrastructure removes variability introduced when researchers use different corpora, preprocessing pipelines, or evaluation implementations.

The Ragnarök framework [82], developed as part of the track infrastructure, emphasizes transparent configurations and reproducible execution. It provides end-to-end tooling for retrieval and generation experiments with explicit configuration management, enabling participants to specify and share complete experimental setups.

A key strength of the track design is that it reduces "hidden degrees of freedom" in evaluation by constraining data and tooling, making comparisons more interpretable. This is particularly important for RAG pipelines, where small changes in preprocessing, retrieval, or decoding can materially alter outputs. Nevertheless, the track still evaluates systems under a small number of prescribed configurations, so it supports reproducibility and comparability without exposing how retrieval selectivity affects performance across a continuous threshold range.

Findings reported by track organizers [82] highlight that reproducibility remains challenging for multi-component RAG pipelines. Small configuration differences across retrieval, generation, or preprocessing can materially alter outputs, and documentation is often incomplete even when code is shared. These findings reinforce the need for complete configuration capture and provenance mechanisms that support independent verification.

The TREC RAG Track advances reproducibility infrastructure but, like RAGAS and RGB, does not systematically characterize how performance varies across retrieval selectivity settings. Submissions are evaluated under track-specified configurations rather than across threshold ranges, leaving sensitivity to retrieval gating unexplored.

### 1.4.4 TruLens and Instrumentation-Based Evaluation

TruLens [83] is an instrumentation-based evaluation framework designed to support systematic assessment of LLM applications, including RAG pipelines, through run-level tracing and reusable "feedback functions." Instead of evaluating only the final answer, TruLens can attach evaluation logic to recorded execution traces, enabling scoring of intermediate artifacts such as retrieved context, prompts, and generated outputs. This design supports diagnostics that distinguish retrieval-related failures (e.g., irrelevant or missing context) from generation-related failures (e.g., unsupported claims despite relevant evidence) and facilitates regression-style comparison across application versions and configurations.

A practical strength of instrumentation-based evaluation is that it treats evaluation as part of the development workflow. By capturing traces alongside feedback scores, TruLens helps practitioners localize failure modes and iterate on components such as chunking strategy, retriever settings, or prompting, while preserving the contextual information required to interpret changes in measured quality. In RAG scenarios, this run-level visibility is particularly useful because multiple interacting configuration layers can influence outputs, and aggregate metrics alone may conceal where degradation originates.

However, although instrumentation improves transparency and supports repeated evaluation under fixed configurations, it does not inherently enforce controlled sensitivity analysis across similarity threshold sweeps. In typical usage, retrieval configuration is treated as a selected setup to monitor and compare, rather than as an explicit independent variable varied systematically to produce threshold-response evidence. As a result, TruLens is well suited for observability, debugging, and evaluation at the run level, but it does not directly provide the empirical basis for evidence-driven similarity threshold calibration across models and domains, which is the central requirement of threshold-aware evaluation.

### 1.4.5 Complementary Evaluation Tooling

In addition to dedicated evaluation frameworks, broader ecosystem tooling supports RAG assessment through tracing, experiment management, testing harnesses, and metric computation. Tracing-oriented platforms such as LangSmith [84], [85] emphasize run capture and

comparison across prompts, retrievers, and model versions, enabling practitioners to inspect inputs, retrieved contexts, intermediate steps, and outputs at the level of individual executions. These artifacts support debugging and regression analysis by making it possible to identify whether failures originate in retrieval (e.g., missing or irrelevant context) or generation (e.g., unsupported claims despite relevant evidence), and they enable consistent comparison across iterations by organizing runs into datasets and experiments.

Complementary observability platforms such as Arize Phoenix [86] similarly support experimentation and troubleshooting through tracing and evaluation workflows, including instrumentation that can be deployed in self-hosted settings. Such tools are useful for operational evaluation because they preserve the execution context needed to reproduce and analyze failures, including retrieved passages, timing, and pipeline-level metadata.

Testing-harness toolkits such as DeepEval [87] provide automated evaluation suites that can be integrated into development workflows, including unit-test-like execution over prompt-based or RAG pipelines. These harnesses typically offer reusable metrics, assertion-style tests, and dataset-driven evaluation runs, which makes them suitable for regression testing after changes to chunking, retrieval configuration, or generation settings.

### 1.4.6 Research Gaps in RAG Evaluation

Across the reviewed evaluation approaches, several gaps can be observed.

**First**, evaluation is frequently reported under fixed retrieval configurations. In many cases, results are presented for a single similarity threshold setting, while the response of retrieval and generation quality to controlled threshold variation is not examined in a systematic way. When threshold is treated primarily as a tuning parameter, the resulting evidence may not capture how retrieval selectivity changes across the threshold range or how threshold effects differ across model architectures and domain corpora. This can complicate threshold-aware configuration decisions, particularly when comparable average scores are achieved under different retrieval selectivity regimes.

**Second**, reproducibility support varies across frameworks and toolchains. While datasets and code are sometimes provided, independent verification may still depend on detailed capture of configuration choices affecting retrieval, generation, preprocessing, and evaluation. For RAG

pipelines, small differences in chunking parameters, embedding models, retrieval settings, or decoding configuration can affect both the retrieved context and downstream generation outputs. Execution traces and observability logs can improve traceability of runs; however, they do not necessarily provide tamper-evident provenance linking datasets, configurations, intermediate artifacts, and outputs. In practice, reported results may therefore remain difficult to audit across environments when configuration capture is incomplete or not expressed in a directly verifiable form.

**Third**, comparative evidence is often reported with an emphasis on proprietary baselines [10], [14]. Under such reporting practices, guidance for open-source deployments may be less explicit, particularly when deployment constraints such as data governance, security, licensing, or cost are relevant. In addition, results may not always be presented with the consistency needed to compare open-source candidates under matched experimental conditions, including consistent retrieval configuration, matched evaluation conditions, and comparable metric computation. As a consequence, evidence suitable for selecting open-source models under resource constraints may be limited or fragmented across sources.

These gaps align with the three deficiencies defined in the Introduction. Gaps related to fixed retrieval configurations correspond to **Deficiency 1** and are addressed by the **Evaluation Procedure** layer. Gaps related to configuration capture and provenance correspond to **Deficiency 2** and are addressed by the **Infrastructure** layer. Gaps related to comparative guidance under open-source constraints correspond to **Deficiency 3** and are addressed by the **Evidence** layer.

## 1.5 Persistent Challenges and Emerging Solutions

Despite considerable progress, RAG deployments continue to expose recurring failure modes. Seven failure points were identified [88]: relevant knowledge absent from the retrieval corpus (missing content); retrieval failing to surface the most pertinent evidence (missed top-ranked documents); incoherent or contradictory retrieved passages (fragmented context); failure to extract relevant information from technical documents (poor content extraction); inconsistent formatting across responses (inconsistent output structuring); responses calibrated at

inappropriate levels of detail (incorrect specificity); and queries only partially addressed (incomplete responses).

Each failure point manifests through a distinct mechanism and carries different implications for reliability:

Missing content occurs when the indexed corpus lacks information required to answer a query. This is a coverage limitation rather than a retrieval scoring error; no configuration can retrieve evidence that does not exist in the collection. It is especially problematic in rapidly changing domains or specialized corpora with incomplete coverage.

Missed top-ranked documents occurs when relevant evidence exists but is ranked below the selectivity gate (similarity threshold) or outside the top-k window. Contributing factors include embedding model limitations, vocabulary mismatch, and configuration choices that exclude borderline-but-useful passages. The consequence is incomplete or fabricated answers despite the presence of evidence in the corpus.

Fragmented context arises when retrieved passages are individually relevant but collectively incoherent or mutually inconsistent. Chunk boundaries, temporal drift across documents, and mixed perspectives can produce a context set that the generator struggles to reconcile, increasing the risk of arbitrary prioritization or awkward synthesis.

Poor content extraction refers to failures in converting source documents into usable text. Technical documents with tables, figures, or complex formatting can lose key information during extraction, so a relevant document may be retrieved but its critical content may be missing or distorted in the context available to the generator.

Inconsistent output structuring appears when responses vary in format and organization across similar queries. This undermines user expectations and can complicate downstream processing for applications that rely on predictable output formats.

Incorrect specificity occurs when responses are calibrated at the wrong level of detail: too general to be actionable or too specific given the available evidence; such behavior can be amplified by generation-time decoding effects that favor high-likelihood but degraded text [89]. Over-specific answers are particularly risky because they can introduce unsupported detail that appears authoritative.

Incomplete responses occur when multi-part queries are only partially addressed. This is common when answering requires synthesizing evidence across multiple sub-questions or dimensions, and it can persist even when some relevant evidence is retrieved.

These failure points highlight structural and operational fragilities affecting many RAG systems. As RAG permeates sensitive application areas—healthcare [7], legal reasoning [8], scientific research [69], and enterprise knowledge management—the demand for verifiable, context-aware, and scalable solutions grows.

Many of these challenges are being addressed by targeted innovations. Auto-RAG [58] and GraphRAG [56] address missing content through dynamic corpus updates and modular knowledge structuring. Relevance sampling [6] and FLARE [63] tackle missed top-ranked documents by iteratively reassessing retrieval during generation. LightRAG [55] and Self-RAG [60] prevent fragmented responses through semantic filtering and feedback-driven refinement. Speculative RAG [59], Self-RAG [60], and FLARE [63] mitigate incomplete responses by incorporating multi-pass reasoning and dynamic retrieval.

These developments demonstrate that while core challenges remain, the research community is actively designing mechanisms to address known vulnerabilities. However, the effectiveness of these mechanisms depends on evaluation frameworks capable of detecting failure modes systematically—a capability constrained by the limitations identified in Section 1.4.6.

## 1.6 Chapter Summary

The evolution of RAG was traced from foundational work in IR and NLP to contemporary modular architectures. Innovations were reviewed across architectural efficiency, data-centric improvement, iterative retrieval, knowledge integration, domain adaptation, factual verification, and few-shot learning. Retrieval configuration—particularly similarity threshold selection—was identified as an understudied factor with potential influence on generation quality.

Persistent challenges in RAG deployments were examined through seven documented failure points that expose structural and operational fragilities. Evaluation frameworks including RAGAS, RGB, and the TREC 2024 RAG Track were analyzed, alongside instrumentation ecosystems

such as TruLens and complementary platforms. These approaches provide metric suites, tracing, observability, and experiment management; however, they are configuration-dependent and do not inherently produce controlled evidence across similarity threshold variations. Moreover, while such tools improve traceability of execution logs and support reproducibility, they do not provide tamper-evident guarantees required for independent verification of reported results.

**Three limitations** were identified across the reviewed literature, corresponding to the deficiencies defined in the Introduction:

**(1)** evaluation commonly remains fixed-configuration rather than threshold-aware **(D1);**

**(2)** reproducibility mechanisms often stop at logging rather than verifiable provenance **(D2);**

**(3)** systematic guidance for open-source deployments across models and domains remains limited **(D3).**

These gaps motivate the reproducibility infrastructure presented in Chapter 2, the model and metric selection defined in Chapter 3, and the controlled experiments reported in Chapter 4.

# CHAPTER 2. DESIGN AND ARCHITECTURE OF PaSSER

Performance Assessment System for Similarity Evaluation and Retrieval (PaSSER) is a modular, browser-based platform for configuring and evaluating RAG pipelines with open-source LLMs [15], [17]. The platform integrates threshold-aware retrieval, multi-metric scoring, and blockchain-backed provenance logging into a unified workflow for controlled experimentation, addressing Deficiency 2 (reproducibility infrastructure) and fulfilling the infrastructure component (b) of Objective 1. Section 2.1 covers the initial system design and its relationship to the SCPDx infrastructure. Section 2.2 addresses the three-layer architecture comprising the web interface, backend services, and blockchain subsystem. Section 2.3 covers user-facing functionalities, and Section 2.4 provides a summary. Experimental configurations and results are reported in Chapter 4.

## 2.1 Initial System Design

PaSSER was developed as a complementary module to the Smart Crop Production Data Exchange (SCPDx) platform [90], [91], [92]. SCPDx combines a blockchain layer—Antelope [93], formerly EOSIO—with the InterPlanetary File System (IPFS) [94] to support secure and decentralized data management. Prior studies informed the choice of blockchain infrastructure, addressing platform suitability [90], supply-chain modeling in agriculture, and oracle integration [95].

Antelope was adopted as the ledger layer in SCPDx because the broader platform required auditable, tamper-resistant records with an explicit permission model and low-cost transactions suitable for frequent submissions. Compared with public chains focused on permissionless consensus, Antelope provides short confirmation times and a mature account and role model that aligns with controlled-access research deployments. The existing EOSIO-based tooling and ecosystem also reduced integration effort, while Anchor Wallet [96], [97] offers a client-side signing workflow that keeps private keys outside the web application.

Within this context, PaSSER was designed as a browser-based environment for evaluating RAG workflows. This implementation provides platform independence and eliminates local installation, simplifying reproducibility across heterogeneous client devices. Alternative

deployment strategies were considered: standalone scripts are easy to run but complicate systematic parameter management and result inspection; desktop clients can operate offline but introduce platform-specific dependencies and increase maintenance burden; containerized deployment improves consistency but adds orchestration overhead unnecessary for research evaluation workflows. The browser-based approach therefore prioritizes accessibility and controlled configuration tracking, while accepting network connectivity as a prerequisite.

## 2.2 System Architecture

The overall architecture of PaSSER follows a three-layer structure. A web interface, implemented as a single-page application (SPA), supports configuration and interaction. Backend services manage vector storage, model inference, and metric computation. A blockchain subsystem records evaluation metadata and outcomes as durable, traceable records. This layered design aligns with the broader SCPDx platform objectives of transparency and secure data management.

### 2.2.1 Web Interface

The PaSSER frontend is implemented as SPA written in JavaScript. It serves as the user-facing entry point, responsible for configuration input, query submission, execution monitoring, and result visualization. The frontend does not perform retrieval, generation, or evaluation logic; all computational tasks are delegated to backend services.

The application is built using the PrimeReact component library [98], which provides structured UI elements for forms, tables, dialogs, and progress indicators. PrimeReact was selected for its comprehensive component coverage and consistent styling across interface elements.

The frontend is compiled into static assets and deployed via a standard web server. In the reference deployment, assets are served by Nginx [99], although the frontend remains server-agnostic. The application executes entirely in the browser and requires no local installation.

Figure 2.1 situates the web interface within the broader PaSSER architecture, illustrating configuration submission, query dispatch, result return, and the path through which on-chain entries are displayed alongside other outputs.

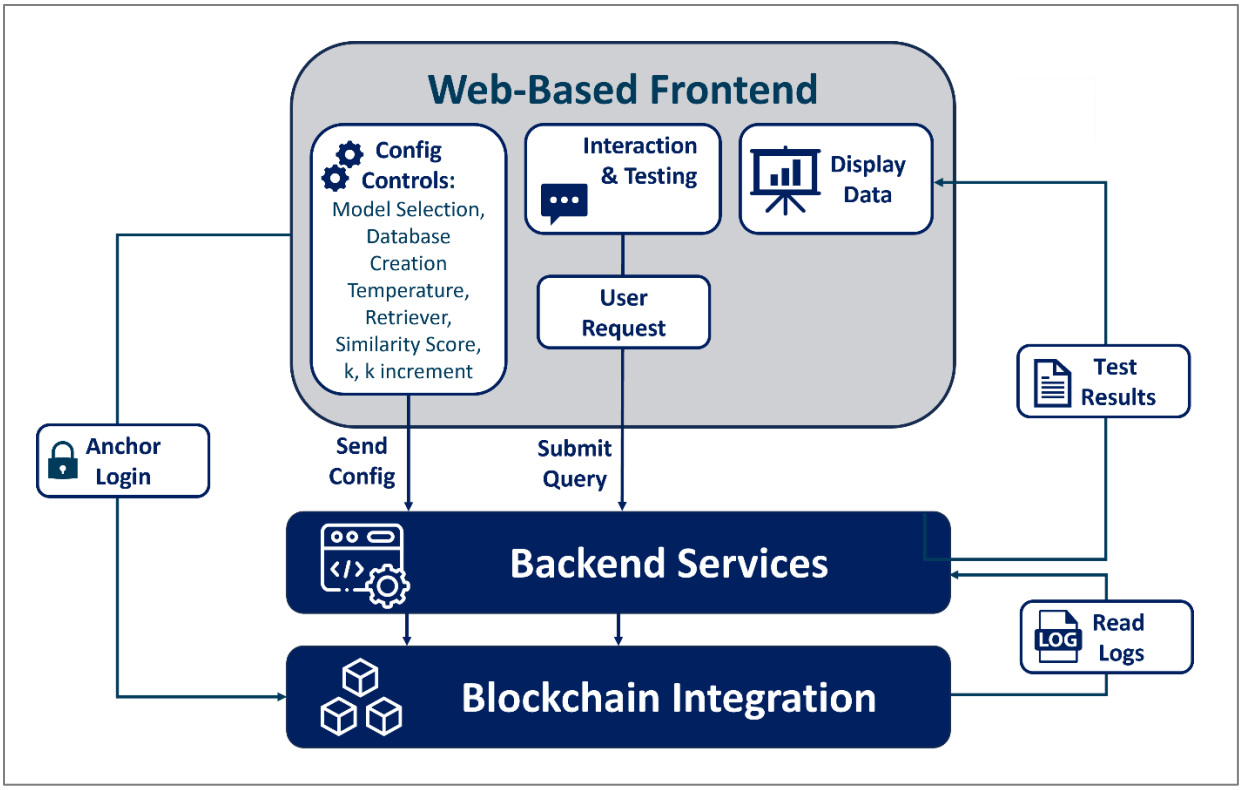


***Figure 2.1*** *PaSSER's Web interface in system context. The frontend is organized into three functional areas: configuration controls (model selection, database creation, temperature, retriever settings, similarity score, k, and k increment), Interaction and Testing interface, and data display. User requests and configuration parameters are transmitted to backend services. Anchor wallet handles login and transaction signing on the client side. Test results and logs are retrieved and displayed through the frontend. Adapted from [15].*

**Scope and Execution Model:** The frontend acts as an orchestration and presentation layer: it collects and applies configuration parameters, submits structured requests to backend endpoints, receives outputs and execution metadata, and renders results in a stable format suitable for comparison. By restricting client-side logic to user-interface concerns, PaSSER keeps retrieval, generation, and scoring behavior invariant across browsers and operating systems, which is essential for the comparative analyses reported in Chapter 4. The frontend maintains only session state such as active views, selected configuration panels, and transient chat history.

**Configuration Interface:** System configuration is organized into two panels: Settings and Add Model. The Settings panel defines backend connection paths for the Ollama application programming interface (API) [100] and ChromaDB [101], selects the active LLM, sets the generation temperature, and specifies the retrieval mode. Two retrieval modes are available (Normal and Score); their technical specifications are described in Section 2.2.2. The Add Model

panel registers additional models for comparative testing. Input ensures that parameters are submitted, and all settings are stored as part of the run metadata to support reproducibility. The selected mode and parameter values are displayed alongside execution results to preserve traceability.

**Query and Evaluation Interfaces.** The frontend provides interfaces for both interactive exploration and batch evaluation. Interactive chat modes allow ad-hoc queries for exploratory use but are not employed in the controlled experiments reported in Chapter 4. Batch evaluation interfaces execute predefined test sequences against configured models and retrieval settings, producing evaluation scores and timing metrics that are logged to the blockchain and displayed through dedicated result views. The evaluation interfaces are described in detail in Section 2.3.

**Result Presentation and Export:** Results returned by the backend are rendered in a structured tabular layout to support comparison across runs. Evaluation scores are displayed in an interface that supports filtering by model, configuration, and timestamp. Results are presented in an organized tabular format and can be viewed with grouping by model, hardware configuration, or test series. The interface supports filtering by model, configuration, or timestamp to facilitate comparison. Results can be exported as .xlsx files for offline analysis, including generated outputs, configuration identifiers, execution metadata, and evaluation scores.

**Authentication and Blockchain Interaction:** User authentication is handled through the Anchor wallet, which provides cryptographic signatures for blockchain submission. The frontend initiates wallet-based authentication and transaction signing; signed transactions are forwarded to the backend, which submits them to the blockchain network. Frontend responsibilities are limited to user interface and configuration logic; all computational tasks, including blockchain interactions, are handled by backend services. Previously recorded evaluation results can be accessed and displayed through the results interface.

**Extensibility and Interface Stability:** The frontend is organized into modular panels and collapsible configuration cards. New functionality can be introduced by adding or extending panels without affecting unrelated views. The modular architecture would accommodate future extensions such as automated threshold sweeps or configurable embedding model selection.

Because configuration payloads and result formats are explicit and versioned, the frontend can evolve without altering archived executions, and older results remain interpretable as new controls are introduced.

### 2.2.2 Backend Services

The backend services implement the execution pipeline of PaSSER, performing all computational tasks associated with RAG, evaluation, and result persistence. The backend acts as a stateless orchestration layer, executing a fixed pipeline that includes vector retrieval, language model inference, evaluation dispatch, and blockchain logging. Figure 2.2 illustrates the position of the backend between the web interface and the blockchain subsystem.

The backend manages all computational tasks, including retrieval, generation, and evaluation, separate from the user interface. This separation enables the system to support larger test batches without affecting frontend responsiveness. Configuration settings are stored as part of the test metadata, ensuring that evaluation conditions can be reproduced exactly in future runs.

**Vector Retrieval Service:** Semantic retrieval is implemented using ChromaDB [101], an open-source vector database optimized for high-dimensional similarity search. ChromaDB stores vectorized document representations and enables efficient retrieval via approximate nearest-neighbor algorithms. The database is licensed under Apache 2.0 and supports multiple programming languages including Python, which facilitated integration with the evaluation pipeline. ChromaDB was selected based on three criteria aligned with the research objectives.

First, its Apache 2.0 open-source license permits unrestricted academic use, modification, and distribution, which supports reproducibility and long-term maintainability of the evaluation environment.

Second, its native Python integration reduces external dependencies and simplifies end-to-end pipeline implementation, including ingestion, retrieval, and experiment orchestration.

Third, its lightweight deployment profile is compatible with the assumed hardware constraints (mid-range systems without dedicated GPU clusters), enabling replication without specialized infrastructure. The choice of vector database implementation is not expected to materially affect similarity threshold sensitivity findings, because ChromaDB implements

standard approximate nearest-neighbor retrieval under cosine similarity. Replication using alternative vector database implementations would nevertheless strengthen generalizability and is noted as future work.

Document corpora are preprocessed offline and embedded using the Ollama embedding endpoint with a specified language model. During ingestion, source documents are segmented into overlapping text chunks prior to embedding and insertion into the vector database. PaSSER exposes chunk size (in characters) and overlap (in characters) as user-configurable parameters through the interface; the default configuration is 1024 characters with 50 characters overlap, which was also used in the controlled experiments reported in Chapter 4. Chunk size and overlap affect retrieval granularity and context continuity: smaller chunks can improve retrieval precision but may fragment semantic units, while larger chunks preserve more context but may reduce selectivity [102]. The implications of chunking sensitivity are discussed in Section 5.3.2.

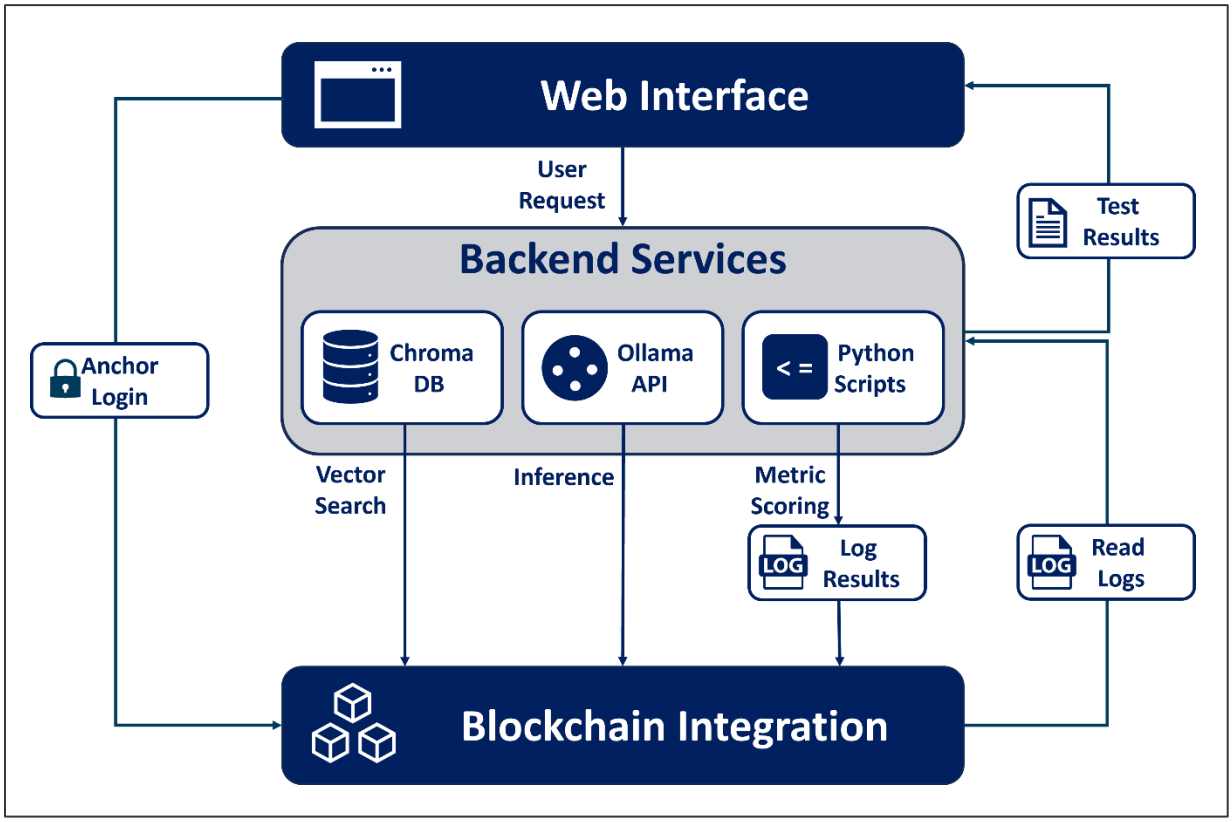


***Figure 2.2*** *Backend services in PaSSER. User requests are processed through three components: ChromaDB for vector search, Ollama API for model inference, and Python scripts for metric scoring. Evaluation results are logged to the blockchain integration layer, and test results are returned to the web interface for display. Reproduced from [15].*

**Retrieval Modes:** At runtime, the backend issues similarity search queries to ChromaDB using the retrieval policy specified in the configuration payload. Two retrieval modes are implemented at this layer. Normal Mode invokes the LangChain *VectorStoreRetriever* [103], returning the *top-k* passages ranked by cosine similarity; this mode was used in Phase I (Section 4.1) under fixed *top-k* retrieval. Score Mode employs the *ScoreThresholdRetriever*, which was introduced to support the threshold sensitivity experiments in Phase II (Section 4.2). Score Mode exposes three parameters: *minSimilarityScore* sets the minimum cosine similarity required for passage inclusion; *maxK* limits the maximum number of passages returned; and *kIncrement* defines the step size for iterative threshold sweeps. Normal Mode returns a fixed number of documents regardless of their absolute similarity values, while Score Mode filters documents by a minimum similarity threshold, resulting in variable context sizes depending on how many passages meet the relevance criterion. The fixed parameter values for Normal Mode runs in Phase I are specified in Section 4.1., while the Score Mode sweep settings used in Phase II are specified in Section 4.2.

The retrieval service returns the selected passages, which are assembled into a context block. The backend combines this context with the user query to form an augmented prompt, which is forwarded to the inference stage.

**Inference Service:** Text generation is executed via the Ollama API [100], which provides a unified interface for managing and invoking open-source LLMs. The API supports model selection and runtime configuration across diverse model families and operating systems, enabling both local and server-based inference setups.

The backend constructs the inference request by combining the user query, the retrieval context (if present), and the decoding parameters specified in the configuration payload.

The inference service executes the request and returns the generated text output together with execution metadata, including model load time and inference duration.

The generated output is forwarded to the evaluation stage, where it is assessed against reference answers using predefined metrics. Evaluation results are then recorded on the blockchain.

**Evaluation Pipeline:** Evaluation is implemented as a separate Python-based service that processes generated outputs together with reference answers supplied through the testing workflow. The service uses established libraries including Natural Language Toolkit (NLTK), torch, NumPy, rouge, transformers, and SciPy [17].

The evaluation service computes a predefined panel of quality indicators and returns the per-question results to the backend for storage and downstream analysis. The full metric set and its formal definitions are provided in Chapter 3, while Chapter 4 specifies the phase-specific metric selections, weighting schemes, and aggregation procedure used for Composite Performance Score (CPS) and Threshold-aware CPS (T-CPS) reporting. In addition to standard automatic metrics such as METEOR, ROUGE, BLEU, and BERTScore, the panel includes the B-RT readability indicators, which are treated as automated proxies rather than human-evaluated judgments; the corresponding limitation is documented in Section 3.4.5.

Evaluation results, along with the associated configuration parameters, are recorded on the blockchain to ensure data integrity and reproducibility.

**Provenance and Blockchain Logging:** PaSSER treats provenance tracking as a backend concern. Each execution records the active configuration, including the model identifier, retrieval parameters, decoding settings, and dataset or vector store identifiers.

The backend submits compact summaries to the Antelope blockchain via a dedicated connector using the Pyntelope library [104]. Logged records include execution timestamps, per-question evaluation metrics (stored as numerical arrays), and the initiating account. Transaction signing is performed externally via the Anchor wallet, while submission is handled by the backend. Full details of blockchain integration are provided in Section 2.2.3

**Execution Control:** The backend supports both interactive and batch execution modes. In batch mode, the system iterates through each question in the dataset, retrieving context according to the configured retrieval policy and submitting the augmented prompt to the selected model. For each item, the backend captures execution metadata including inference duration and model loading time. Retrieval settings, model parameters, and evaluation metrics are applied consistently across all items in the batch, ensuring that runs can be compared under identical

conditions. Results are recorded on the blockchain and exported to spreadsheet format for offline analysis.

**Deployment Characteristics:** Backend services support both local and remote deployment. The inference service may operate on CPU-only hardware or use GPU acceleration where available. During platform development and evaluation, testing was performed on an Ubuntu server (CPU-only, 128GB RAM) running Nginx, an Apple Mac M1 (macOS, GPU-accelerated, 16GB RAM), and an Apple Mac M2 (macOS, 16GB RAM). Formal experimental configurations are specified in Chapter 4.

The backend communicates with Ollama, ChromaDB, Python evaluation scripts, and the Antelope blockchain through their respective APIs.

### 2.2.3 Blockchain Integration

PaSSER employs the Antelope blockchain (formerly EOSIO) to provide immutable logging of evaluation results and associated metadata. As described in Section 2.1, Antelope was selected to leverage the existing SCPDx infrastructure. The blockchain is not part of the retrieval or generation path and does not participate in runtime decision-making. Its function is limited to persisting verifiable records of completed evaluations, supporting auditability, reproducibility, and long-term comparison of results.

User authentication and transaction signing are handled via the Anchor wallet, which generates the cryptographic signatures required for blockchain submission. The backend prepares payloads and submits signed transactions to the network. Figure 2.3 illustrates the blockchain integration architecture and the two smart contract actions used for logging.

**Logging Model and Execution Boundaries:** Blockchain logging occurs after a backend execution completes. No intermediate states, partial outputs, or raw model responses are written on-chain. Each logged entry corresponds to a completed evaluation unit. This design ensures that blockchain records remain compact, deterministic, and suitable for long-term storage.

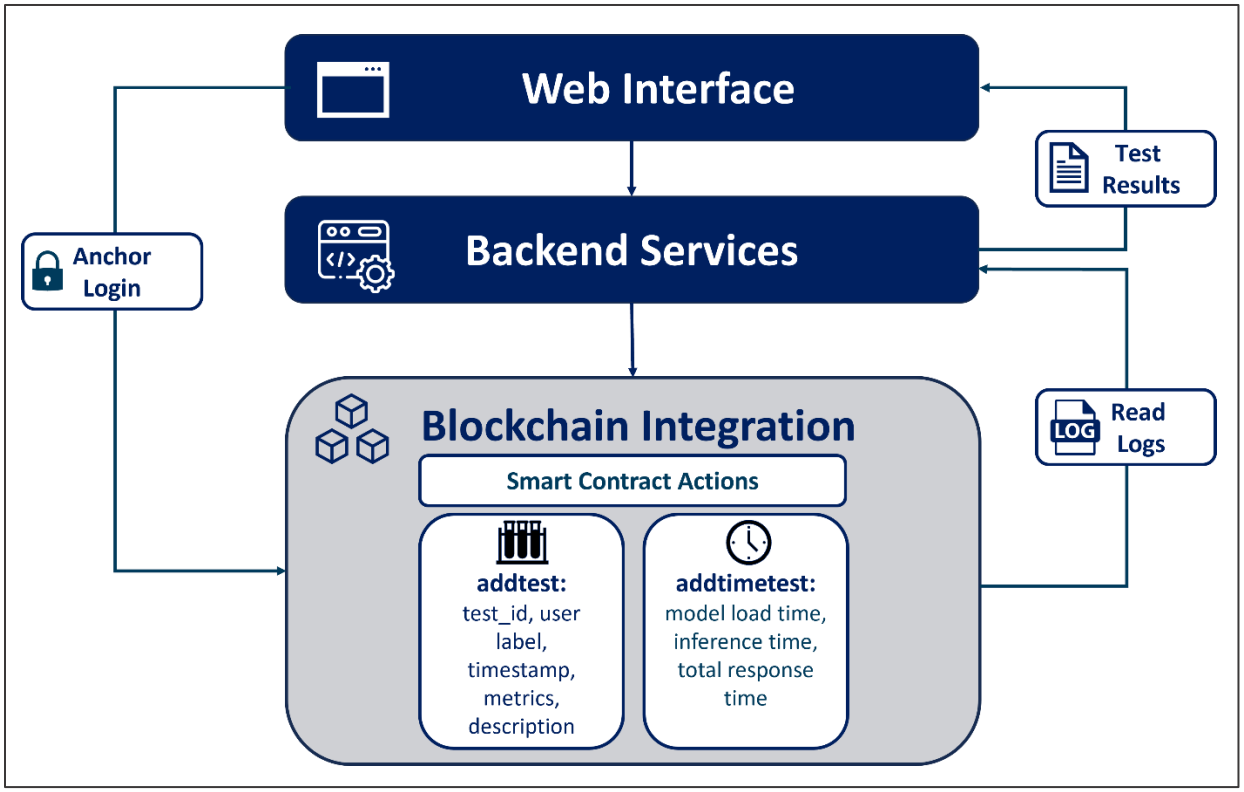


***Figure 2.3 Blockchain integration in PaSSER.*** *Evaluation outputs are recorded via two smart contract actions: addtest stores accuracy-related metrics (test_id, user, label, timestamp, metrics, description), and addtimetest stores timing metrics (model load time, inference time, total response time). Logged results can be retrieved and displayed through the frontend. Reproduced from [15].*

**Smart Contract Structure:** Evaluation results are stored via a dedicated smart contract named *'llmtest'* deployed on the Antelope blockchain. The contract defines a persistent table for test records using the *eosio::multi_index* abstraction. Each table entry includes a unique test identifier, user ID (linked through Anchor authentication), test label, timestamp, evaluation metrics (e.g., ROUGE, cosine similarity), and a textual description field.

The table supports multiple secondary indices to enable efficient retrieval by timestamp, user, or test identifier. This allows backend services to query historical results for reporting, comparison, or export without scanning the full ledger.

The *'addtest'* action records accuracy-related metrics, while the companion *'addtimetest'* action logs time-related performance data such as model load duration, inference time, and total response time. Both actions are append-only; existing records are never modified or deleted, preserving the immutability guarantees of the blockchain.

**Transaction Submission and Authentication:** Transaction submission follows a wallet-mediated model. The frontend initiates authentication through Anchor, which verifies user identity and signs the transaction payload. The signed transaction is forwarded to the backend, which submits it to the blockchain network.

The backend does not perform cryptographic signing and does not have access to private keys. Its responsibility is limited to preparing the compact payload, submitting the signed transaction, receiving confirmation, and storing the resulting transaction identifier alongside the execution record. This separation ensures that credential management remains entirely on the client side while preserving backend control over execution flow.

**Result Retrieval and Presentation:** Logged results can be retrieved from the blockchain through the smart contract's indexed query interface. The contract supports retrieval by user, test type, and execution date, enabling efficient access to historical evaluation records. Retrieved results are displayed through the frontend interfaces described in Section 2.3 and can be exported for offline analysis.

**Independent verification workflow.** Each completed evaluation record is bound to a wallet-signed blockchain transaction submitted through the *addtest* and *addtimetest* actions. The on-chain record persists the submitting account *(creator / userid),* a run identifier *(testid*), a timestamp *(created_at),* a descriptive label *(description)*, and a compact numeric payload (results, stored as *float64[]*). Independent verification is performed by retrieving the corresponding table entry and comparing these fields against the matching fields in the exported run artifacts used in the analysis. Because the record is confirmed on-chain and linked to a signer identity through the wallet signature, post hoc modification of reported results would be detectable as a mismatch between the exported artifacts and the immutable blockchain record.

**Scope and Architectural Role:** Within the overall PaSSER architecture, the blockchain subsystem functions as a verifiable persistence layer. Although the broader SCPDx platform includes IPFS integration for distributed content storage, IPFS is not currently utilized within PaSSER; future enhancements may incorporate it to support capabilities such as storage and distribution of fine-tuning artifacts.

## 2.3 PaSSER App Functionalities

This section describes the functional capabilities of PaSSER as exposed to the user through the web interface. The emphasis is on workflow stages and the artifacts produced at each stage. The workflow is organized into two phases: setup and configuration (Figure 2.4), and evaluation and results (Figure 2.7).

### 2.3.1 System Configuration

The configuration workflow proceeds as follows:

1. **Connect to services:** Specify endpoints for the inference server (Ollama API) and vector store (ChromaDB).
2. **Select model and parameters:** Choose the active LLM from registered models and set the generation temperature.
3. **Choose retrieval policy:** Select Normal Mode (top-k) or Score Mode (threshold-based filtering with configurable parameters). Technical definitions of these modes are described in Section 2.2.2.
4. **Link data resources:** Select the vector store to be used for retrieval and, for evaluation workflows, import a JSON-formatted dataset of question–answer pairs.
5. **Confirm configuration:** Once confirmed, settings are attached to all outputs for reproducibility.

Configuration is session-scoped; subsequent workflow stages operate under the established settings without modification.

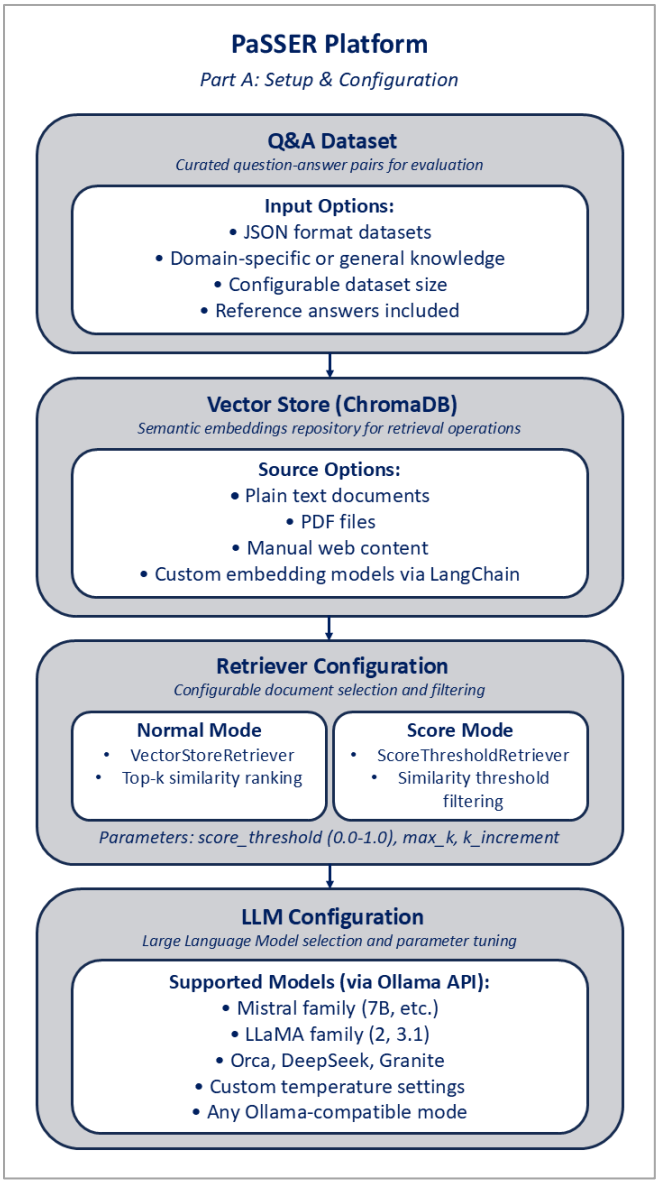


***Figure 2.4 PaSSER workflow overview:*** *Setup and configuration. The diagram illustrates the initial workflow stages, including Q&A dataset preparation, vector store creation via ChromaDB, retriever configuration (Normal and Score modes), and LLM configuration via the Ollama API. Reproduced from [15].*

## 2.3.2 Data Management

Data management covers the preparation and lifecycle control of vector stores used for retrieval. The workflow for creating a vector store proceeds as follows:

1. **Select source type:** Choose from plain text, PDF documents, or curated website content.
2. **Upload or enter content:** Provide the source material through the appropriate interface.

3. **Process and store:** The system segments the content into overlapping chunks, generates embeddings, and stores the resulting vectors as a ChromaDB collection. Technical details of the ingestion pipeline are described in Section 2.2.2.
4. **Register for use:** Newly created stores are automatically registered and become available for selection in configuration and evaluation workflows.

Once created, vector stores are passive resources that can be reused across multiple sessions without accumulating run-specific state. The management interface (Manage Databases menu), shown in Figure 2.5, allows users to list all existing stores, inspect individual entries, and remove obsolete stores to prevent unintended reuse.

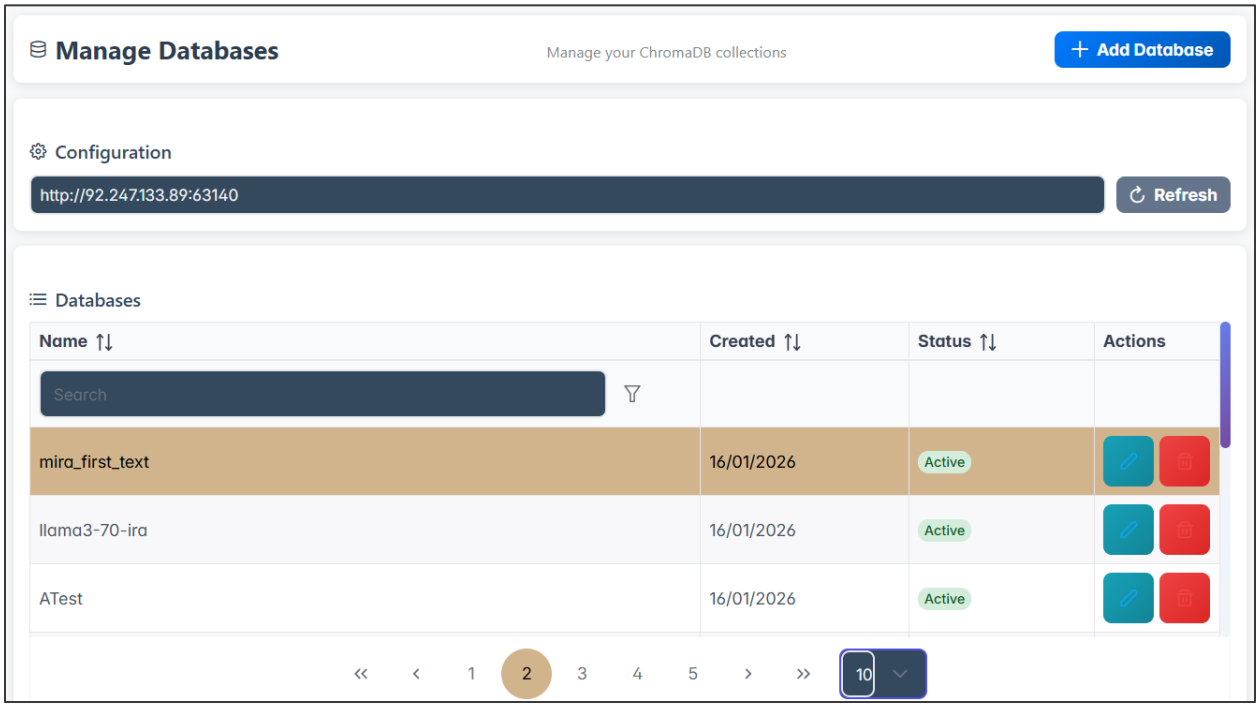


***Figure 2.5 Manage Databases interface.*** *The management view lists all existing vector stores with creation date and status. Users can search, inspect, or remove stores through this interface.*

## 2.3.3 Retrieval Configuration

Retrieval configuration determines how passages are selected from a vector store and assembled as context for generation. Users select one of two retrieval modes through the interface shown in (Figure 2.6):

**Normal Mode** is selected for fixed-size context retrieval;

**Score Mode** is selected when threshold-based filtering is required, with parameters for minimum similarity, maximum passages, and iteration step size entered through the configuration panel. Technical definitions of both modes appear in Section 2.2.2.

**Baseline configuration (Normal Mode).** In PaSSER, the baseline configuration corresponds to Normal Mode retrieval. Normal Mode returns the top-k most similar passages for each query without applying a similarity threshold as a minimum cosine similarity cutoff. This baseline serves as the reference condition against which Score Mode (threshold-enabled) configurations are compared in Chapter 4.

Retrieval parameters are chosen at entry. Once set, the same retrieval policy applies uniformly across interactive querying and batch evaluation, ensuring that observed differences in outputs can be attributed to retrieval settings.

**Controlled similarity threshold sweep procedure.** PaSSER operationalizes controlled similarity threshold evaluation through the Score Mode retrieval option. In Score Mode, a similarity threshold is applied as a minimum cosine similarity cutoff for including retrieved passages in the generation context. A sweep is executed by running the same evaluation workload repeatedly while varying only the similarity threshold value. All other run parameters are held constant, including dataset, vector store identifier, chunking configuration, retrieval depth (top-k), and generator configuration. Each threshold run produces a distinct, labeled output artifact and is stored as a separate evaluation record, enabling direct comparison across similarity threshold values under a fixed experimental setup.

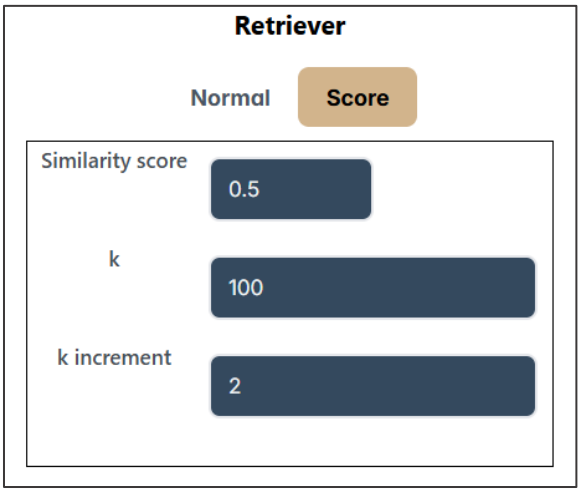


***Figure 2.6. Retriever configuration.*** *Users select Normal or Score mode.*

### 2.3.4 Model Interaction

PaSSER provides two chat modes for interacting with models:

**Standard Q&A Chat**: Prompts are submitted directly to the selected model without retrieval. This mode serves as a baseline for evaluating model behavior in isolation.

**RAG-Based Q&A Chat**: Each prompt first triggers retrieval from the selected vector store under the active retrieval policy. Retrieved passages are appended to the prompt before inference, enabling domain-specific reasoning.

Both modes maintain a session-scoped conversational buffer for multi-turn interaction. For each exchange, the system records the prompt, retrieved context (when applicable), model output, and timing metadata. This enables direct comparison between retrieval-augmented and non-augmented behavior under identical conditions.

### 2.3.5 Evaluation and Testing

Evaluation and testing enable systematic, dataset-driven assessment of models under controlled retrieval configurations. Figure 2.7 provides an overview of the evaluation and results workflow, including response generation, multi-metric evaluation, composite scoring, and results management.

The Tests menu (Figure 2.8a) provides access to the following functionalities:

- **Dataset Preparation (Q&A Dataset)**: Users prepare evaluation datasets by providing domain-specific reference answers and a prompt that instructs the LLM to generate corresponding questions. The resulting question–answer pairs are stored in a standardized JSON format for reuse across models and configurations.
- **Content Evaluation (RAG Q&A Score Test)**: Users select a vector store collection, load a JSON dataset, and initiate evaluation. For each question, the system retrieves context,

constructs an augmented prompt, generates a response, and computes evaluation metrics. Results are recorded on the blockchain via the addtest action.

- **Runtime Evaluation (Q&A Time LLM Test)**: This module measures latency-related metrics including model load time, inference duration, and total response time. Results are recorded on the blockchain via the addtimetest action.
- **Result Inspection (Show Test Results / Show Time Test Results)**: As shown in Figure 2.8b, evaluation outcomes are retrieved from the blockchain and displayed in a searchable list organized by test name and date. Results can be exported as Excel files for offline analysis.

Technical details of blockchain logging and metric computation are described in Sections 2.2.3 and 2.2.2 respectively.

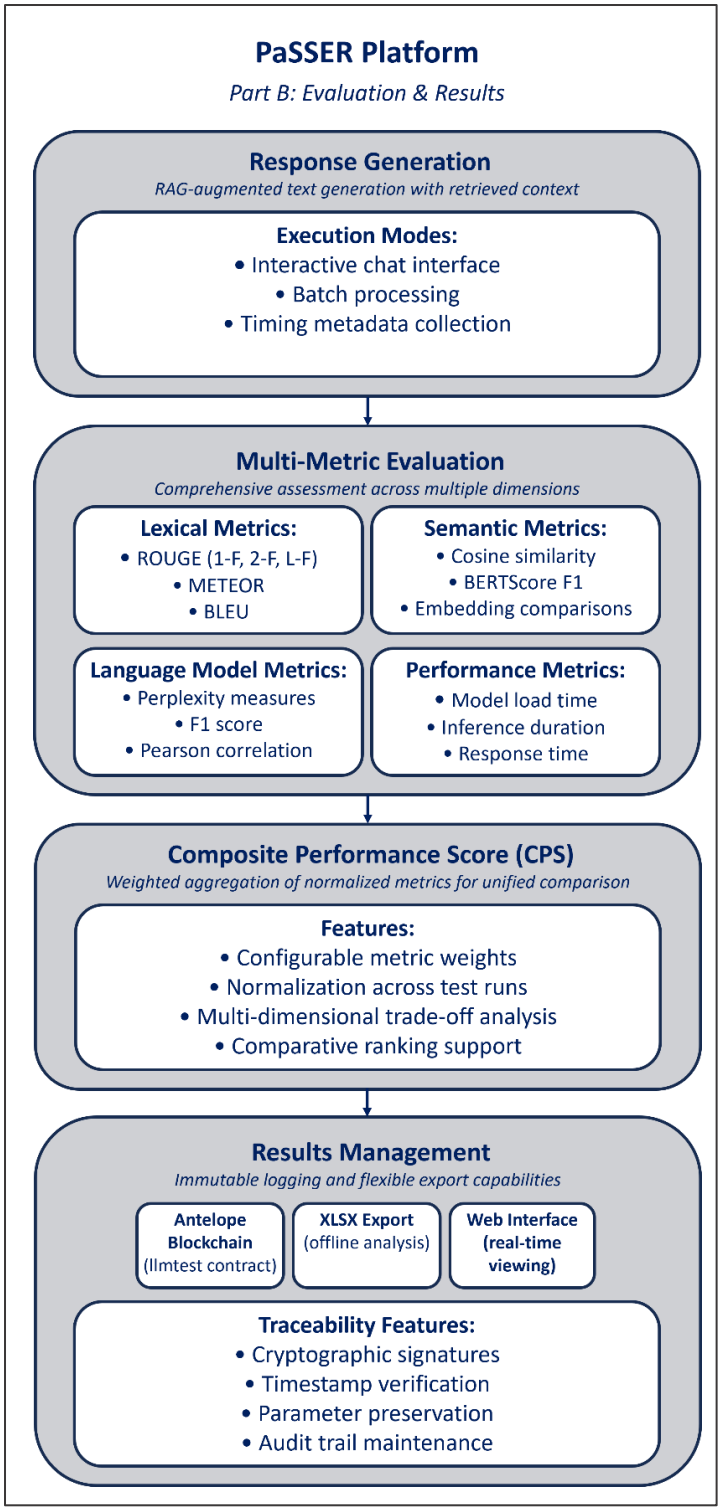


***Figure 2.7 PaSSER workflow overview**: Evaluation and results. The diagram illustrates response generation modes (interactive chat, batch processing, timing metadata collection), multi-metric evaluation across lexical, semantic, language model, and performance dimensions, Composite Performance Score aggregation, and results management including blockchain logging, XLSX export, and web interface viewing. Reproduced from [15].*

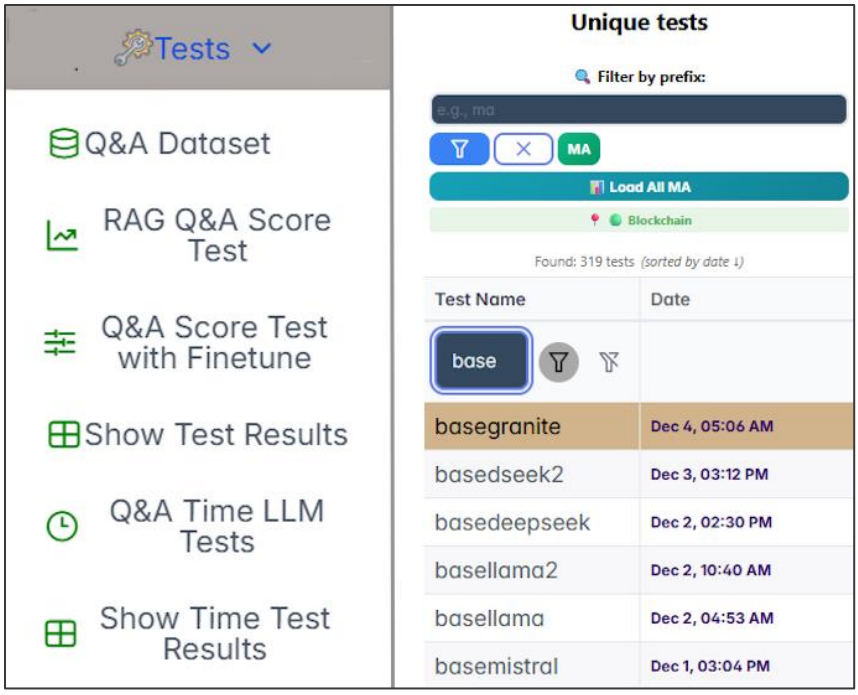


***Figure 2.8. Evaluation and testing interfaces.*** *(a) Tests menu showing available functionalities. (b) Unique tests list displaying results retrieved from blockchain with search and filtering options.*

## 2.4 Chapter Summary

The PaSSER platform implements reproducibility infrastructure for threshold-aware RAG evaluation. The architecture integrates three functional layers: a browser-based interface enabling experiment configuration and result visualization, a Python backend coordinating retrieval operations through ChromaDB vector storage and language model inference through the Ollama API, and an Antelope blockchain layer ensuring tamper-evident provenance logging for all evaluation outcomes.

Six functional categories support systematic experimental workflows. System configuration establishes runtime parameters including service endpoints, model selection, and retrieval policy. Data management provides a uniform pipeline for creating and managing vector stores from text, PDF, and website sources. Retrieval configuration exposes Normal and Score modes, enabling controlled experiments on context breadth and retrieval selectivity across similarity threshold settings. Evaluation and testing enable dataset-driven assessment with multi-metric scoring and blockchain-based result logging. Configuration settings and evaluation

outcomes are recorded with each test, ensuring traceability between results and experimental conditions.

This architectural foundation supports reproducible evaluation by ensuring that experimental conditions and outcomes are fully traceable and verifiable, addressing Deficiency 2. The threshold-aware retrieval configuration enables the experiments reported in Chapter 4. The PaSSER source code is publicly available; repository details are provided in Section 5.2.2.

# CHAPTER 3. MODEL SELECTION AND EVALUATION METRICS

This chapter defines the two components required for the experimental analyses in Chapter 4: the set of open-source LLMs evaluated within the PaSSER platform, and the evaluation metrics used to assess generation quality. Sections 3.1–3.3 document the model set, selection rationale, and integration constraints, addressing Objective 2 (establish model selection criteria). Section 3.4 specifies the evaluation metrics, including definitions, computation procedures, and reporting formats. Section 3.5 presents the Composite Performance Score (CPS) and Threshold-aware Composite Performance Score (T-CPS) formulations used for multi-metric aggregation, addressing Objective 3 (define metric selection and computation procedures). Together, these components contribute to Deficiency 1 (threshold-aware evaluation through CPS, T-CPS, and Balance Score) and Deficiency 3 (practical guidance for open-source deployments through model selection criteria and evaluation procedures).

## 3.1 Overview of Evaluated LLMs and Model Selection Criteria

A representative set of open-source LLMs was integrated into PaSSER to support controlled, reproducible comparisons under consistent experimental conditions. Selection focused on open availability, feasibility of local inference on mid-range hardware, and architectural diversity within the 7B–8B parameter range. This range balances capability and accessibility: models can typically run on 16–32 GB RAM without dedicated GPU clusters while avoiding the memory demands of 40B-class architectures [105]. Models below 7B were excluded due to limited capacity for knowledge-intensive generation in CPU- and memory-constrained deployments, while models above 8B were excluded due to higher hardware requirements that exceed typical mid-range configurations [105]. Proprietary models were also excluded because licensing restrictions, cost, and limited configuration transparency undermine strict reproducibility [106].

To reflect both established and more recent open-source model families, the models are grouped into two sets:

***Initial set:*** Mistral 7B, Llama 2 7B, and Orca 2 7B were selected due to broad adoption, modest hardware requirements, and complementary training emphases (efficiency-oriented

pretraining, general-purpose capability, and instruction-tuned reasoning behavior, respectively). This model set was used in Phase I as a pilot to evaluate end-to-end RAG execution under the baseline retrieval configuration (Normal Mode), and it was retained in Phase II for the pilot similarity threshold sensitivity experiments conducted in Score Mode.

***Updated set:*** Granite 3.2 8B, DeepSeek R1 8B, Llama 3.1 8B, and Mistral 7B v0.3 were selected to extend analysis to newer 8B-class open-source LLMs and to test whether similarity threshold sensitivity patterns generalize beyond the initial 7B-class set. Mistral 7B v0.3 is retained as the anchor model to maintain continuity across phases while capturing revisions within the same model family. This updated set is evaluated under the full similarity threshold range used in Phases III and IV, and Phase IV repeats the same sweep on a second corpus to assess domain shift effects.

This selection directly supports the evaluation objectives. Open availability and local inference feasibility ensure that the evaluation environment can be reproduced on mid-range hardware, enabling execution of large threshold sweeps and repetition of experiments under identical conditions. Architectural diversity is relevant because similarity thresholds may interact with model-specific behaviors: context-window limits constrain how much retrieved evidence can be injected into a single prompt; attention and memory efficiency influence how much context can be processed without truncation or performance degradation; and instruction or reasoning tuning may influence whether a model integrates retrieved passages or defaults to parametric knowledge [5], [107], [108].

By combining efficiency-oriented, general-purpose, and reasoning-tuned models, the evaluation can distinguish whether threshold sensitivity is primarily a retrieval-configuration effect or a model-dependent effect. Chapter 4 applies these criteria in controlled experiments.

The following subsections summarize each model's architectural characteristics and training objectives, with justification for its inclusion in the evaluation workflow.

## 3.2 Initial Set

The three models in this set—Mistral 7B [109], Llama 2 7B [110], and Orca 2 7B [111]—were used to demonstrate end-to-end system functionality and profile runtime characteristics

under fixed top-k retrieval in Phase I [17]. In Phase II [16], the same models were evaluated under systematic threshold variation (0.50–0.80) to characterize threshold sensitivity.

### 3.2.1 Mistral 7B

Mistral 7B is a transformer-based large language model introduced in 2023 by Mistral AI [112], [109]. It targets the 7B parameter range with an explicit focus on inference efficiency and practical deployment on a single GPU or comparable mid-range hardware.

Two architectural choices contribute to this efficiency profile. First, Mistral 7B uses grouped-query attention (GQA), which reduces decoding-time memory cost by sharing key-value projections across multiple query heads and thereby improves throughput during generation. Second, it applies sliding-window attention (SWA), which limits attention computation to a fixed local window while still allowing information to propagate across layers, supporting long prompts with lower incremental cost than full attention. These features are relevant for RAG workflows, where retrieved passages increase prompt length and place direct pressure on latency and memory during inference.

With respect to training data, Mistral AI describes the model as trained on a large-scale mixture of text sources but does not disclose a complete breakdown of the dataset composition. The release specifies a byte-fallback Byte Pair Encoding (BPE) tokenizer, which supports broad token coverage and reduces failure cases for uncommon strings.

Reported benchmark results place Mistral 7B above Llama 2 7B on standard evaluations, [113], including Massive Multitask Language Understanding (MMLU) [114] and GSM8K [115].

Mistral 7B is distributed through Hugging Face and local runners such as Ollama. Within the initial set, it serves as the efficiency-focused representative.

### 3.2.2 Llama 2 7B

Llama 2 7B [110] was released by Meta in 2023 as the second generation of the LLaMA model family. It is a transformer-based large language model in the 7B parameter class, commonly used as a baseline in open-weight research. The reference distribution reports training on a large-scale mixture of publicly available online data.

Meta also released tuned variants (Llama 2 Chat) that apply supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) [108] to improve helpfulness and safety in interactive use cases [108], [110]. The tokenizer is a BPE model based on SentencePiece, designed for efficient handling of diverse text and multilingual content [116].

Llama 2 7B is distributed under a community license that enables broad research and commercial usage, though with conditions that do not meet the Open-Source Initiative definition of "open source." Reported evaluations in the Llama 2 7B paper position the model as competitive among open-weight alternatives on standard benchmarks.

Within the initial set, Llama 2 7B serves as the general-purpose baseline, providing a reference point for comparing how retrieval settings and metric outcomes vary when efficiency-oriented (Mistral 7B) or reasoning-focused (Orca 2 7B) models are used.

### 3.2.3 Orca 2 7B

Orca 2 7B [111], developed by Microsoft Research in 2023, is a fine-tuned derivative of Llama 2 7B designed to strengthen reasoning and structured problem solving in small models. The core distinction from the base model is not architectural; Orca 2 7B applies instruction tuning on curated, high-quality synthetic training data intended to teach the model how to select and apply appropriate reasoning strategies depending on the task.

Because Orca 2 7B is built on a Llama 2 7B base, it inherits the same transformer backbone and inference constraints. The model's reported gains arise primarily from the training approach and data design rather than from attention or context-length modifications.

The Orca 2 7B paper reports improvements on reasoning-focused benchmarks in zero-shot settings, including GSM8K and BIG-bench Hard [117], compared to Llama 2 Chat baselines and other similarly sized open-weight models. These results motivate its inclusion in evaluation scenarios where response quality depends not only on retrieving relevant passages but also on integrating information and maintaining coherent multi-step reasoning.

Within the initial set, Orca 2 7B serves as the reasoning-focused representative, complementing Mistral 7B (efficiency-oriented) and Llama 2 7B (general-purpose baseline).

## 3.3 Updated Set

Following the initial experiments, the initial model set was replaced with an updated set reflecting recent architectural developments and extending the evaluation to the 8B parameter class. This set—Granite 3.2 8B [118], DeepSeek R1 8B [119], Llama 3.1 8B [120], and Mistral 7B v0.3 [121]—was used in Phase III (model-dependent threshold analysis) and in Phase IV (cross-domain threshold analysis). Mistral was retained from the initial 7B group but updated to the later v0.3 release.

### 3.3.1 Granite 3.2 8B

Granite 3.2 8B [118] is part of IBM's Granite family of open-weight foundation models released in 2024. At approximately eight billion parameters, it targets deployments requiring a balance between capability and feasible local inference. IBM documentation presents Granite as an enterprise-oriented model line with attention to dataset governance and traceability, although independent corroboration of these claims is limited.

The architecture follows a transformer design with engineering changes intended to improve scaling behavior and factual consistency. IBM reports training on a curated mixture of public-domain text, licensed sources, and filtered web content, with data provenance practices described in the release documentation.

Benchmark evaluations reported by IBM indicate competitive performance across general language understanding and knowledge-intensive tasks, with scores in line with other 8B-class models on MMLU. IBM claims reduced hallucination rates compared to similarly sized models.

Granite 3.2 8B is available through Hugging Face and Ollama. Within the updated set, it represents an enterprise-oriented design philosophy distinct from the efficiency, general-purpose, and reasoning emphases of the initial models.

### 3.3.2 DeepSeek R1 8B

DeepSeek R1 8B [119] is part of the DeepSeek research initiative, released in 2024 with the aim of advancing open-weight reasoning capabilities in medium-scale models.

The architecture follows the transformer paradigm but is distinguished by a training regime that emphasizes reasoning traces and logical decomposition. The training corpus combined large-scale public datasets with synthetic examples designed to expose the model to intermediate steps rather than only final answers. DeepSeek's documentation describes the use of reinforcement learning techniques to further refine these capabilities [122]. This training philosophy parallels Orca 2 7B but reflects more recent developments in reasoning-oriented distillation and synthetic data generation. The model is based on Llama 3.1 8B and supports long-context use, making it suitable for RAG workflows in which multiple retrieved passages are incorporated into a single prompt

On benchmarks, DeepSeek R1 8B achieves competitive accuracy on reasoning-oriented tasks including GSM8K and BIG-bench Hard relative to other 8B-class models, while maintaining solid performance on general benchmarks such as MMLU.

DeepSeek R1 8B is available on Hugging Face and Ollama. Within the updated set, it serves as the reasoning-focused counterpart at the 8B scale, continuing the line of inquiry begun with Orca 2 7B while reflecting more recent advances in synthetic data and reasoning-oriented training.

### 3.3.3 Llama 3.1 8B

Llama 3.1 8B [120], released by Meta in July 2024, is the successor to Llama 2 in the LLaMA family of open-weight models. At 8 billion parameters, it occupies the same size class as Granite 3.2 8B and DeepSeek R1 8B but maintains a general-purpose orientation with improved instruction-following capabilities.

Compared with Llama 2, which is commonly documented with a 4,096-token context length, Llama 3.1 8B provides substantially greater context capacity, enabling better handling of long-form inputs in RAG workflows where multiple passages are concatenated into a single prompt. The tokenizer has been updated for improved efficiency in multilingual and domain-diverse contexts.

Training combined publicly available text with licensed sources under more rigorous filtering than Llama 2 7B. Alignment was performed through supervised fine-tuning and RLHF, refining the approach introduced in Llama 2.

Benchmark results indicate that Llama 3.1 8B performs competitively on both general knowledge tasks (MMLU) and reasoning benchmarks (GSM8K, BIG-bench Hard) relative to other 8B-class models [123].

Llama 3.1 8B is available on Hugging Face and Ollama. Within the updated set, it serves as the general-purpose baseline at the 8B scale, succeeding Llama 2 7B from the initial set while reflecting current developments in the LLaMA family.

### 3.3.4 Mistral 7B (Latest Edition v0.3)

The updated set includes Mistral 7B v0.3 [121], released in 2024 and distributed via Ollama under the tag *mistral:latest* [100]. This version retains approximately 7.3 billion parameters but introduces an extended context window of 32,768 tokens and support for function calling.

The model preserves the efficiency-oriented architecture described in Section 3.2.1, including GQA and sliding-window attention. Refinements in fine-tuning and dataset curation have improved factual consistency and multilingual handling compared to the initial 2023 release [109].

On benchmarks, Mistral 7B v0.3 achieves competitive results on MMLU, GSM8K, and BIG-bench Hard relative to other models in the 7B parameter range [112].

Within the updated set, Mistral 7B v0.3 serves two purposes: it maintains continuity with the initial experiments by enabling direct comparison across iterations of the same model family, and it captures the current state of development within the 7B class. It serves as the efficiency-focused representative, complementing Granite 3.2 8B (enterprise-oriented), DeepSeek R1 8B (reasoning-focused), and Llama 3.1 8B (general-purpose baseline). Table 3.1 summarizes the key characteristics of all models in the initial and updated sets.

***Table 3.1*** *Comparative summary of evaluated models.*

| Model | Parameters | Key design emphasis | Primary evaluation role | Ollama tag |
|---|---|---|---|---|
| Mistral 7B / v0.3 | ~7.3B | GQA; sliding-window attention | Efficiency-focused; cross-version continuity | Mistral 7B / Mistral:latest |
| Llama 2 7B | 7B | Standard transformer; widely adopted | General-purpose baseline | Llama 2 7B |
| Orca 2 7B | 7B | Reasoning-oriented fine-tuning | Reasoning-tuned comparator | Orca 2 7B |
| Granite 3.2 8B | 8B | Enterprise-oriented curation | Enterprise reliability comparator | Granite3.2 8b |
| DeepSeek R1 8B | 8B | Reasoning-focused RL training | Reasoning-focused 8B comparator | Deepseek r1:8b |
| Llama 3.1 8B | 8B | Updated LLaMA; extended context | Current-generation baseline | llama3.1:8b |

## 3.4 Evaluation Metrics

Evaluating RAG differs from conventional Natural Language Generation (NLG) because a retrieval step explicitly shapes the model's output. Consequently, a single score is insufficient. The assessment must reflect multiple dimensions: lexical overlap with references, semantic alignment, fluency and uncertainty of the generated text, statistical agreement with ground truth, and practical readability.

A multi-metric panel of twenty-four metrics was applied to support balanced and reproducible comparisons (Table 3.2).

**Metric Selection Rationale.** The 24-evaluation metrics in Table 3.2 were selected to provide broad coverage of complementary assessment dimensions - lexical overlap, semantic similarity, fluency and predictability, statistical correlation and readability - while remaining feasible for large-scale experimentation. This selection follows the general ensemble principle that diverse measures can produce more reliable aggregate assessment than any single metric [124]. Sixteen metrics were implemented in Phase I; the remaining eight were added in Phase III.

These include established NLP metrics such as METEOR, ROUGE, BLEU, and perplexity variants, embedding-based semantic measures such as cosine similarity and BERTScore, statistical alignment measures such as the Pearson correlation coefficient, and readability-oriented scores such as the B-RT suite. Taken together, this panel supports analysis of how RAG systems trade off factual correctness and contextual grounding against fluency and interpretability.

The full 24-metric panel is computed by PaSSER for each evaluation run.

***Table 3.2*** *Complete enumeration of the 24-evaluation metrics.*

| Category | Metric | Output column | Description | Implementation |
|---|---|---|---|---|
| Lexical overlap | METEOR | METEOR | Token-level alignment with stemming and synonym matching; balances precision and recall | Phase I |
| Lexical overlap | ROUGE-1 | Rouge-1.r | Unigram overlap recall | Phase I |
| Lexical overlap | ROUGE-1 | Rouge-1.p | Unigram overlap precision | Phase I |
| Lexical overlap | ROUGE-1 | Rouge-1.f | Unigram overlap F1 | Phase I |
| Lexical overlap | ROUGE-2 | Rouge-2.r | Bigram overlap recall | Phase I |
| Lexical overlap | ROUGE-2 | Rouge-2.p | Bigram overlap precision | Phase I |
| Lexical overlap | ROUGE-2 | Rouge-2.f | Bigram overlap F1 | Phase I |
| Lexical overlap | ROUGE-L | Rouge-l.r | Longest common subsequence recall | Phase I |
| Lexical overlap | ROUGE-L | Rouge-l.p | Longest common subsequence precision | Phase I |
| Lexical overlap | ROUGE-L | Rouge-l.f | Longest common subsequence F1 | Phase I |
| Lexical overlap | BLEU | BLEU | n-gram precision with brevity penalty | Phase I |
| Lexical overlap | F1 Score | F1 Score | Token overlap F1; diagnostic for answer correctness | Phase I |
| Semantic similarity | Cosine similarity | Cosine similarity | Embedding-space similarity between generated and reference texts | Phase I |
| Semantic similarity | BERTScore | Bert-Score.precision | Contextual token similarity precision | Phase III |
| Semantic similarity | BERTScore | Bert-Score.recall | Contextual token similarity recall | Phase III |
| Semantic similarity | BERTScore | Bert-Score.f1 | Contextual token similarity F1 | Phase III |
| Fluency / predictability | Laplace perplexity | Laplace Perplexity | Surface-level predictability under a Laplace-smoothed bigram n-gram language model | Phase I |
| Fluency / predictability | Lidstone perplexity | Lidstone Perplexity | Surface-level predictability under a Lidstone-smoothed trigram n-gram language model | Phase I |
| Statistical correlation | Pearson correlation | Pearson correlation | Linear association between generated and reference representations | Phase I |
| Readability proxy (B-RT) | B-RT Coherence | B-RT.coherence | Topic focus and local organization | Phase III |
| Readability proxy (B-RT) | B-RT Consistency | B-RT.consistency | Self-consistency across claims | Phase III |
| Readability proxy (B-RT) | B-RT Fluency | B-RT.fluency | Readability and grammatical flow | Phase III |
| Readability proxy (B-RT) | B-RT Relevance | B-RT.relevance | Alignment to query framing | Phase III |
| Readability proxy (B-RT) | B-RT Average | B-RT.average | Arithmetic mean of B-RT components | Phase III |

### 3.4.1 Lexical Overlap Metrics

Lexical overlap metrics compare system outputs to reference answers at the level of words and fixed-length sequences. They provide a fast, reproducible baseline for automatic evaluation and help verify that retrieved context has been reflected in the generated text.

Three established metric families are used in this category:

METEOR (Metric for Evaluation of Translation with Explicit ORdering). Balances precision and recall, with stemming and synonym matching to credit near-misses.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation). Captures n-gram overlap and longest common subsequence through ROUGE-1, ROUGE-2, and ROUGE-L, each reported as recall (r), precision (p), and F1 Score (f).

BLEU (Bilingual Evaluation Understudy). Emphasizes the precision of matched n-grams (with a brevity penalty), rewarding fluent reproduction of salient reference content.

These metrics do not measure meaning directly, but they offer clear indicators of alignment with references and are useful for tracking whether RAG outputs incorporate retrieved evidence accurately. The F1 Score, which is also computed from token overlap, is discussed in Section 3.4.3 alongside answer quality metrics.

***METEOR***

The METEOR metric [125] evaluates the quality of machine-generated text by comparing it against one or more reference texts through a process of linguistic alignment. Unlike simpler n-gram overlap metrics, METEOR incorporates stemming and synonym matching, with one-to-one word alignments.

The calculation proceeds in several stages. First, precision and recall are derived from unigram matches:

$$P = m/w\_c \, , R = m/w\_r \tag{3.1}$$

where $m$ is the number of matched unigrams, $w_c$ is the total number of unigrams in the candidate, and $w_r$ is the total number of unigrams in the reference. Precision measures the

proportion of words in the candidate that appear in the reference, while recall measures the proportion of words in the reference that are recovered by the candidate.

Second, a fragmentation penalty is introduced to account for disordered or non-contiguous matches. This reflects the intuition that fluent text preserves contiguous sequences rather than scattering relevant words. The penalty is computed as:

$$Penalty = 0.5\left(\frac{c}{m}\right)^3 \tag{3.2}$$

where $c$ is the number of contiguous matched chunks. A higher number of fragments increases the penalty, reducing the score even if the raw overlap is high.

Finally, precision and recall are combined into a weighted harmonic mean, with recall weighted more heavily (9:1) to favor outputs that cover the full content of the reference. The METEOR score is then adjusted by the penalty:

$$M_{score} = F_{mean}(1 - Penalty), \tag{3.3}$$

$$F_{mean} = \frac{10PR}{R+9P} \tag{3.4}$$

This formulation rewards outputs that are both accurate and fluent, while discouraging scattered word matches that lack coherent ordering.

In the context of RAG, METEOR provides a useful measure of how well retrieved context has been integrated into generated responses. A high METEOR score indicates that the model not only recovered key reference content but also reproduced it in a way that respects structure and linguistic coherence. Lower scores may suggest shortcomings in retrieval relevance, insufficient integration of context, or weaknesses in generation quality.

***ROUGE***

ROUGE [126] is a family of overlap-based metrics widely used for evaluating summarization, machine translation, and related generation tasks by comparing system output

to one or more human references. Core variants include ROUGE-N (n-gram overlap), ROUGE-L (longest common subsequence), and ROUGE-W (weighted LCS).

**ROUGE-N (n-gram overlap).** For a fixed n, compute overlap between candidate and reference n-grams; report recall, precision, and F1:

$$Recall_{ROUGE-N} = \frac{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (3.5)$$

$$Precision_{ROUGE-N} = \frac{\sum_{S \in \{System\ Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{System\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (3.6)$$

$$F1_{ROUGE-N} = 2\frac{Precision_{ROUGE-N} \times Recall_{ROUGE-N}}{Precision_{ROUGE-N} + Recall_{ROUGE-N}} \quad (3.7)$$

ROUGE-1 (n=1, unigram overlap) and ROUGE-2 (n=2, bigram overlap) alongside ROUGE-L.

R_ROUGE-L (longest common subsequence). ROUGE-L focuses on the longest common subsequence (LCS) between candidate and reference texts. The LCS captures ordered but not necessarily contiguous matches, providing a measure of structural alignment. Scores are reported as recall, precision, and F1:

$$Recall_{ROUGE-L} = \frac{LCS(System\ Summary, Reference\ Summary)}{Length\ of\ Reference\ Summary} \quad (3.8)$$

$$Precision_{ROUGE-L} = \frac{LCS(System\ Summary, Reference\ Summary)}{Length\ of\ System\ Summary} \quad (3.9)$$

$$F1_{ROUGE-L} = 2\frac{Precision_{ROUGE-L} \times Recall_{ROUGE-L}}{Precision_{ROUGE-L} + Recall_{ROUGE-L}} \quad (3.10)$$

ROUGE-W extends ROUGE-L by assigning greater weight to longer matching sequences, but it is not applied in the PaSSER framework.

The relative emphasis on recall, precision, or F1 depends on the application: recall is favored when maximizing information coverage is essential, whereas precision is prioritized when relevance and conciseness are more critical.

In RAG, ROUGE serves as a direct indicator of how much retrieved context is preserved in generated outputs. High ROUGE scores suggest that the model successfully incorporates key content from references, while low scores may reveal deficiencies in either retrieval relevance or generative consistency.

***BLEU***

The BLEU metric [127] evaluates machine-generated text by measuring n-gram overlap with one or more references. Unlike recall-oriented metrics, BLEU emphasizes precision, rewarding outputs that reproduce relevant phrases while penalizing overly short candidates via a brevity penalty.

Modified n-gram precision (for n = 1…N). Candidate n-gram counts are clipped by the maximum count in any reference to prevent score inflation from repetition:

$$P_n = \frac{\sum_{g \epsilon G_{n(cand)}} \min\left(Count_{cand}(g), \max_{r \epsilon R} Count_{ref}(g)\right)}{\sum_{g \epsilon Gn(cand)} Count_{cand}(g)} \quad (3.11)$$

where *G_n(cand)* is the multiset of candidate n-grams and *R* is the set of reference texts. The numerator contains clipped counts; the denominator contains total candidate counts.

To discourage excessively short outputs, BLEU introduces a brevity penalty (BP):

$$BP = \begin{cases} 1, & if\ c > r, \\ e^{(1-r/c)}, & if\ c \leq r, \end{cases} \quad (3.12)$$

where *c* is the corpus-level total length of the candidate outputs, and *r* is the corpus-level effective reference length (sum of the closest-length reference for each candidate sentence).

Final BLEU. A (typically equal-weight) geometric mean of the modified precisions, scaled by BP

$$BLEU = BP.exp(\sum_{n=1}^{N} w_n lnP_n) \quad (3.13)$$

where *N* is the maximum n-gram length and *w_n* are the corresponding weights (commonly *w_n* = 0.25 for *N* = 4).

In the context of RAG, BLEU serves as an indicator of surface-level fidelity to reference answers. High BLEU scores suggest that the model has successfully reproduced key expressions from retrieved context, while low scores may reflect weaknesses in retrieval precision, contextual integration, or generative fluency.

**Summary.** Together, METEOR, ROUGE, and BLEU provide complementary perspectives on word-level fidelity in generated text. METEOR integrates precision, recall, and linguistic variation; ROUGE captures n-gram and subsequence coverage; and BLEU emphasizes phrase-level precision while penalizing overly short outputs. Applied jointly, they establish a reliable baseline for evaluating how effectively RAG systems reproduce reference content. The F1 Score, which also measures token-level overlap, is discussed in Section 3.4.3.

### 3.4.2 Semantic Similarity Metrics

Semantic similarity metrics evaluate whether a generated response preserves the meaning of a reference beyond surface word overlap. Rather than counting shared n-grams, they operate in an embedding space, where sentences or tokens are mapped to vectors that encode contextual relations such as paraphrase and synonymy. Two metrics are employed in this category. Cosine similarity provides a sentence- or passage-level measure of alignment by computing the cosine of the angle between embedding vectors; values closer to 1 indicate stronger semantic proximity. BERTScore assesses correspondence at the token level using contextualized transformer embeddings and reports precision, recall, and F1 based on the alignment of candidate and reference tokens. Used together, these metrics complement lexical ones by capturing meaning-preserving variation in phrasing, provided that implementation details—embedding model, pooling strategy, and normalization—are kept consistent to ensure comparability across experiments.

***Cosine Similarity***

Cosine similarity [128] is one of the most widely used metric of vector similarity in natural language processing. It quantifies how closely two texts are related by comparing their vector representations in a high-dimensional embedding space. Unlike distance-based metrics that depend on magnitude, cosine similarity evaluates the orientation of vectors, making it invariant to differences in vector length or scale.

Formally, for two vectors *A* and *B* of dimension *n*, cosine similarity is calculated as:

$$Cosin\ Similarity = \frac{A.B}{\|A\|\ \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.14}$$

where A · B is the dot product, and ‖A‖ and ‖B‖ are the Euclidean norms of the vectors. The score ranges between −1 and 1, with 1 indicating identical orientation (maximum similarity), 0 indicating orthogonality (no relation), and −1 indicating opposite orientation.

When applied to text embeddings produced by transformer models or other encoders, cosine similarity captures semantic relatedness rather than exact word overlap. For example, the sentences "The cat is sleeping" and "A feline takes a nap" may share no common n-grams but yield a high cosine similarity because their embeddings occupy nearby positions in vector space.

Within RAG, cosine similarity serves two roles. During retrieval, it ranks candidate passages in a vector store against the query embedding, favoring documents whose content is semantically aligned with the user's intent. During evaluation, it provides a sentence- or passage-level check of semantic agreement between a generated response and its reference, complementing lexical metrics such as ROUGE or BLEU.

This dual application positions cosine similarity as both a core mechanism in constructing RAG systems and a diagnostic tool for assessing their performance. By relying on embedding-based alignment rather than surface forms, it enables evaluation to capture synonyms, paraphrases, and domain-specific terminology that lexical metrics would miss.

***BERTScore***

BERTScore [129] is a semantic evaluation metric that uses contextual embeddings from transformer models to assess similarity between a generated text and a reference. Rather than counting exact n-gram matches, it compares high-dimensional token representations, making it appropriate when meaning matters more than surface overlap.

The procedure has three steps. First, both candidate and reference sentences are encoded with a pretrained transformer (e.g., BERT, RoBERTa), yielding a vector for each token. Second, a similarity matrix is formed by computing cosine similarity between every candidate token and every reference token. Third, scores are aggregated into precision, recall, and F1 by aligning each token to its best semantic match.

Formally, let $X = \{x\_i\}$ and $Y = \{y\_i\}$ be the token embeddings of the candidate and reference, respectively. Using cosine similarity $s(x, y)$*:*

$$Precision = \frac{1}{\|X\|}\sum_{x\epsilon X} \underset{y\epsilon Y}{max}\, s(x,y) \tag{3.15}$$

$$Recall = \frac{1}{\|Y\|}\sum_{y\epsilon Y} \underset{x\epsilon X}{max}\, s(x,y) \tag{3.16}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.17}$$

In RAG, BERTScore is particularly valuable because it can recognize when a generated response captures the intent of the reference even with different wording. For instance, if a model generates “maize cultivation” instead of “corn farming,” BERTScore identifies the semantic equivalence, whereas lexical metrics like ROUGE or BLEU may underestimate the similarity.

By combining precision, recall, and F1, BERTScore provides a fine-grained view of semantic fidelity. High recall indicates that the model has included most of the relevant meaning, while high precision suggests that the generated content avoids unnecessary or irrelevant additions.

The F1 Score balances these perspectives, offering a stable indicator of overall semantic alignment in RAG outputs.

**Summary.** Taken together, cosine similarity and BERTScore extend evaluation from surface-level overlap to semantic fidelity. Cosine similarity provides a straightforward measure of alignment in embedding space, while BERTScore refines this with token-level contextual comparisons reported through precision, recall, and F1. Used jointly, they capture whether generated responses convey the intended meaning of reference texts, even when expressed with different wording.

### 3.4.3 Fluency, Predictive, and Answer Quality Metrics

Fluency and predictive metrics evaluate whether a model's output reads naturally and whether its word sequences follow predictable surface patterns. Unlike lexical or semantic similarity metrics, which compare output to a reference, these metrics quantify surface-level predictability under a fitted n-gram distribution.

This group includes Laplace Perplexity and Lidstone Perplexity; lower values indicate sequences that are more typical under the fitted n-gram distribution. The group also includes the F1 Score. While F1 is traditionally a classification metric and is computed from token overlap (similar to ROUGE and BLEU), it is included in this section because it directly measures answer correctness, that is, whether generated tokens match expected answer elements. Taken together, these metrics complement overlap and embedding-based metrics by indicating whether RAG outputs are not only relevant to retrieved evidence, but also surface-predictable and answer-correct.

**Implementation Note.** Laplace Perplexity and Lidstone Perplexity are computed using classical n-gram language models implemented in NLTK (nltk.lm), not from the evaluated transformer model's token probabilities. Laplace Perplexity uses a bigram model (order = 2) with Laplace (add-one) smoothing, while Lidstone Perplexity uses a trigram model (n = 3) with Lidstone smoothing ($\lambda = 0.1$). The n-gram model is trained on the evaluation reference text, and perplexity is computed on the generated candidate text under the resulting n-gram distribution. These perplexity values are model-independent proxies under the fitted n-gram distribution and should

not be interpreted as the evaluated model's internal confidence. The implications of this n-gram-based approach for metric interpretation are discussed in Section 5.3.3.

### *Laplace Perplexity*

Perplexity [19], [130] reflects surface-level predictability under a fitted language model: lower perplexity indicates that the sequence is more typical under the fitted n-gram distribution. In essence, perplexity measures the "average branching factor"—the number of plausible continuations assigned to each point in a sentence.

A practical problem arises when a model assigns zero probability to unseen events under maximum-likelihood estimation, which drives perplexity to infinity. Add-one (Laplace) smoothing avoids this by redistributing a small amount of probability mass from frequent events to unseen ones so that no outcome has zero probability.

Formally, with vocabulary size *V*, the adjusted probability of word *w_i* given its history *h* is:

$$P_{Laplace}(w_i|h) = \frac{C(w_i,h)+1}{C(h)+V}, \tag{3.18}$$

where *C* (*w_i*, *h*) is the joint count of *w_i* and its history, and *C*(*h*) is the count of the history alone.

Given sequence *W* = *w_1*, ..., *w_N*, perplexity under Laplace smoothing is:

$$PPL\ (W) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} ln\ P_{Laplace(w_i\,|\,h)}\right), \tag{3.19}$$

This formulation penalizes sequences that contain low-probability tokens and rewards sequences whose tokens are consistently predictable.

In RAG outputs that contain rare or domain-specific terms introduced by retrieval, smoothing prevents zero-probability failures, while lower Laplace PPL indicates greater surface-level typicality under the fitted n-gram distribution.

### *Lidstone Perplexity*

While Laplace smoothing ensures stability by eliminating zero probabilities, it does so at the cost of overestimating the likelihood of unseen events. Adding one to every possible word

count, regardless of vocabulary size, disproportionately redistributes probability mass when the vocabulary is large. To mitigate this effect, Lidstone smoothing [19] replaces add-one smoothing with an add-λ adjustment, where λ is a small positive constant. Instead of adding one, a fraction λ is added to each count, allowing finer control over probability redistribution. The parameter λ is set to 0.1.

With vocabulary size *V*, the Lidstone-smoothed conditional probability of word *w_i* given history *h* is:

$$P_{Lidstone}(w_i|h) = \frac{C(w_i,h)+\lambda}{C(h)+\lambda V} \quad (3.20)$$

Perplexity under Lidstone smoothing for a sequence *W = w_1, ..., w_N* is:

$$PPL\ (W) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} ln\ P_{Lidstone(w_i\ |\ h)}\right) \quad (3.21)$$

The flexibility of Lidstone smoothing lies in the tuning of $\lambda$. A small $\lambda$ value reduces the distortion introduced to observed events, ensuring that frequent words retain high probabilities while still allocating some probability mass to unseen events. In practice, this makes Lidstone smoothing more suitable for large vocabularies or specialized domains, where the add-one approach of Laplace would excessively flatten probability distributions.

Applied to RAG evaluation, Lidstone perplexity provides a finer-grained proxy of surface predictability when the generated output contains rare or domain-specific terms, under the fitted n-gram distribution. For example, when retrieval introduces technical terminology, Lidstone smoothing avoids zero-probability assignments without excessively inflating the likelihood of unseen events. Generated text that achieves consistently low Lidstone PPL demonstrates stable surface patterns while maintaining balanced treatment of common and rare terms.

***F1 Score***

The F1 Score [131], [132] is a widely used evaluation metric that balances two complementary dimensions of performance: precision and recall. While F1 is traditionally a classification metric, it is included in this section because it measures answer correctness at the token level—whether generated tokens match expected answer elements. This framing aligns F1 with the answer-quality orientation of this section, though it operates differently from perplexity

metrics. In the context of RAG, F1 provides a composite measure of how accurately and comprehensively generated responses reproduce reference content. Unlike perplexity, which assesses fluency, or semantic similarity metrics, which capture meaning preservation, the F1 Score explicitly captures the trade-off between correctness and completeness.

Formally, precision is defined as the proportion of generated items that are relevant, while recall measures the proportion of relevant items that are successfully generated:

$$Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN} \tag{3.22}$$

where *TP (true positives)* are correctly generated elements, *FP (false positives)* are irrelevant or incorrect outputs, and FN (false negatives) are relevant elements that were omitted.
The F1 Score is calculated as the harmonic mean of precision and recall:

$$\mathrm{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.23}$$

F1 attains 1 only when both precision and recall are 1, and approaches 0 when either term is near 0. This definition is standard in IR and machine learning evaluation.

In multi-class or imbalanced settings, F1 is aggregated via micro, macro, or weighted averaging: micro computes global *TP/FP/FN* before calculating F1, macro averages per-class F1 uniformly, and weighted averaging scales per-class F1 by support. The choice of averaging affects sensitivity to class imbalance and should be reported alongside scores.

While F1 is widely used, it compresses two distinct aspects (precision and recall) into one number; interpretation should be paired with the underlying precision and recall values or with task-specific costs. Nonetheless, when a single scalar is needed, F1 remains a pragmatic summary of retrieval and classification effectiveness.

In RAG, the F1 Score is particularly important because retrieval and generation errors manifest differently: irrelevant context reduces precision, while missed documents or incomplete answers reduce recall. By combining both aspects, the F1 Score provides a balanced diagnostic of output quality, making it a valuable component of multi-metric evaluation.

**Summary.** Together, Laplace and Lidstone perplexity quantify the statistical predictability of generated text under different smoothing assumptions, while the F1 Score complements them by balancing correctness and completeness. Applied jointly, these metrics reveal whether RAG outputs are both linguistically coherent and aligned with the information needs of the task.

### 3.4.4 Statistical Correlation Metrics

Statistical correlation metrics [133] assess the degree of linear association between paired numeric values. Unlike overlap-based, semantic, or fluency-based metrics that evaluate generated text directly, correlation quantifies relationships between scores — for example, between model-produced similarity scores and human-annotated relevance judgments, or between scores assigned by different models to the same responses.

This metric category is represented by the Pearson correlation coefficient, which measures the strength and direction of linear relationships between evaluation scores. High positive correlation indicates that predicted scores vary consistently with reference scores; low or negative correlation suggests weak or inverse alignment.

#### *Pearson Correlation*

The Pearson correlation coefficient *r* [133], [134] quantifies the strength and direction of a linear relationship between two continuous variables. It is the normalized covariance of the variables, bounded in [−1, 1], and is therefore interpretable as how consistently one variable increases or decreases with the other.

Formally,

$$r = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \tag{3.24}$$

where $n$ is the number of pairs, and $\bar{X}$ and $\bar{Y}$ are the sample means.

High $r$ indicates that predicted scores co-vary with reference scores in a consistent linear manner; low or negative $r$ points to weak or inverse alignment. Because $r$ targets linear association and is sensitive to outliers and non-linearity, it should be interpreted alongside scatter

plots or complemented by rank-based metrics (e.g., Spearman correlation) when relationships are monotonic but non-linear.

**Summary.** Taken together with text-based metrics, Pearson correlation offers a complementary perspective: it evaluates consistency with ground-truth trends rather than surface overlap, helping to diagnose whether improvements in scoring or ranking functions actually track reference evaluations.

### 3.4.5 Human-Readability Inspired Metrics (B-RT)

The B-RT suite implements a Nubia-inspired regression proxy intended to approximate human judgments of readability in RAG [135], [136]. It yields four primary signals—Coherence, Consistency, Fluency, and Relevance—and an aggregate index (B-RT.Average). Raw scores are computed on a 0–5 working scale, clipped to that interval, and then normalized to the [0,1] range by division by 5 to enable direct comparison across models and configurations.

**Interpretation note.** B-RT is used as an automated proxy for answer readability and internal consistency, but it is not treated as a substitute for human judgment. Unless evaluated against human ratings on the same outputs, B-RT scores should be interpreted as a comparative signal inside the evaluation pipeline rather than a direct measure of perceived quality. Chapter 4 therefore reports B-RT alongside lexical and semantic metrics, and any B-RT-driven conclusions are phrased conservatively.

**Technical implementation.** B-RT relies on a pretrained BERT-base encoder and the *[CLS]* representation. Let *E_cls(x)* denote the *[CLS]* embedding of a text *x*. For a generated output *G*, a reference string is first formed by concatenating the literal label "Query:" with the user query *Q* and the literal label "Context:" with the retrieved context *R*.

The base semantic similarity is then defined as:

$$s = \cos\big(E_{cls}(Reference),\ \ E_{cls}(G)\big), \tag{3.25}$$

which couples the score directly to the inputs consumed during generation. In addition, a single combined input is prepared containing the literal label "Reference:" followed by the previously formed reference string and the literal label "Candidate:" followed by the model output *G*. This

combined text is encoded once, and its *[CLS]* vector provides auxiliary signals, obtained through fixed projections and norms, that are used for the Coherence and Fluency estimates. Throughout all experiments, the encoder checkpoint, tokenizer, truncation length, projection slices, scaling constants, and clipping rules remain fixed to preserve version-stable behavior. The raw *[CLS]* hidden state is used rather than the library's pooler output, because the raw state is more stable for similarity and avoids task-specific bias.

**Component metrics.** The four B-RT signals are computed as follows:

- **Coherence** approximates local smoothness and structural flow by taking the norm of a designated slice of the combined-input [CLS] embedding (dimensions 0–100), multiplying by the base similarity *s*, applying a fixed scaling factor, and clipping to the 0–5 range.
- **Consistency** captures semantic alignment between the candidate and the provided evidence by linearly scaling *s* into the 0–5 interval, yielding a stable signal of agreement with the fused query-plus-context reference without explicit claim extraction.
- **Fluency** serves as a lightweight proxy for grammaticality and lexical naturalness by using a different, non-overlapping slice of the combined-input [CLS] embedding (dimensions 100–300), applying a fixed factor, multiplying by *s*, and clipping to 0–5.
- **Relevance** measures how directly the candidate addresses the user query and its supporting context as another scaled version of *s* in the same 0–5 working range.

After these four raw values are obtained, each is divided by 5 to produce a normalized score in [0,1], and the aggregate readability index is computed as the arithmetic mean,

$$B - RT.Average = \frac{Coherence + Consistency + Fluency + Relevance}{4} \quad (3.26)$$

The [0,1] values are used for reporting; the 0–5 scale is internal only.

**Rationale for proxy approach.** Nubia was evaluated as an alternative [135]; the Nubia-inspired proxy was adopted because it provided more reproducible and maintainable scoring across heterogeneous environments and aligned directly with RAG inputs (*Q*, *R*, *G*), without introducing the additional dependency and version constraints associated with a full Nubia stack. Chapter-scale sweeps over thresholds, datasets, and models impose tight constraints on throughput; the proxy reduces setup and inference overhead, keeping end-to-end evaluation

times predictable so that large experimental matrices complete reliably. The proxy also scores directly against the actual inputs to generation by forming a single reference string and deriving all readability signals from the same encoder pass, which aligns metric behavior with operational use. Finally, the proxy removes moving parts tied to particular checkpoints and preprocessing conventions that may be sensitive to library versions and device differences, thereby providing a leaner and more stable path to comparable scores.

**Measurement Constraints.** Two measurement constraints are acknowledged. First, using [CLS] projections rather than sentence-level dynamics means B-RT's coherence cannot diagnose specific discourse faults such as dangling references or abrupt topic shifts within long generations; it functions as a stable scalar proxy rather than a fine-grained analyzer. Second, substituting a projection-based fluency signal for token-level perplexity reduces sensitivity to subtle grammatical errors in otherwise strong outputs. Correlation between B-RT scores and actual human judgments has not been empirically validated. The proxy is adopted as a practical approximation that enables scalable evaluation across large experimental matrices, but users requiring validated human-alignment guarantees should complement B-RT with direct human evaluation or employ the full Nubia implementation with appropriate calibration. The implications of these measurement constraints for metric interpretation are discussed in Section 5.3.3.

## 3.5 Composite Performance Scores

Evaluating RAG requires simultaneous assessment of multiple quality dimensions. Individual metrics such as METEOR, BERTScore, or perplexity each capture distinct aspects of generation quality, but comparing system configurations across a panel of twenty-four metrics presents interpretive challenges. When Model A outperforms Model B on some metrics but underperforms on others, no single metric provides a definitive ranking.

The aggregation of multiple metrics into a composite score must handle three complications. First, metrics operate on different scales: METEOR produces values in [0, 1], while perplexity can range into the hundreds. Second, metrics have opposing polarities: higher METEOR indicates better performance, whereas lower perplexity indicates better performance. Third,

metrics differ in diagnostic importance depending on the evaluation context: lexical overlap may matter more in factual question-answering than in open-ended summarization.

**Metric subset and construct mapping.** Although PaSSER computes all 24 metrics for each evaluation run, CPS aggregation uses a nine-metric subset. Correlated variants (e.g., precision and recall when F-score is included, individual B-RT components when B-RT.average is included) are excluded to avoid double-weighting similar constructs. The five metric families defined in Section 3.4 are regrouped into four evaluation constructs for aggregation: (1) lexical overlap, (2) semantic similarity and alignment, (3) fluency and answer correctness, and (4) language modeling. Pearson Correlation is grouped under semantic similarity and alignment; B-RT components are mapped to the constructs they measure. Table 3.3 summarizes this mapping. Phase-specific metric panels and weights are documented in Chapter 4.

To support systematic comparison across retrieval configurations and models, two composite scoring formulations are employed. The Composite Performance Score (CPS) addresses these complications through min-max normalization with polarity adjustment, followed by weighted summation [16]. The Threshold-aware Composite Performance Score (T-CPS) [18] extends CPS by incorporating a stability term that penalizes configurations with high output variability.

This section defines CPS and T-CPS, including normalization and polarity handling (Sections 3.5.1–3.5.2). It then specifies the statistical evaluation used to test phase results (Section 3.5.3) and introduces the Balance Score that relates stability to performance (Section 3.5.4).

Algorithmic specifications for CPS calculation, T-CPS calculation, and statistical significance testing are provided in Appendix A. Complete source code implementations are available in the repository referenced in Section 5.2.2.

***Table 3.3.*** *Mapping of Section 3.4 Metric Families to CPS Evaluation Constructs.*

| Section 3.4. metric family (5 groups) | Primary purpose of the family | Mapped CPS evaluation construct (4 constructs) | Mapping rule / notes |
|---|---|---|---|
| 3.4.1 Lexical overlap (METEOR, ROUGE variants, BLEU) | Measures surface-form overlap with the reference (token/phrase overlap) | Lexical overlap | Direct mapping. All lexical overlap measures belong here. |
| 3.4.2 Semantic similarity (Cosine Similarity, BERTScore variants) | Measures meaning preservation beyond exact word overlap | Semantic similarity and alignment | Direct mapping. Embedding-based similarity measures form the core of this construct. |
| 3.4.3 Fluency, predictive, answer quality (Laplace Perplexity, Lidstone Perplexity, F1 Score) | Captures linguistic predictability and answer correctness | Split mapping: Language modeling (perplexities) and Fluency and answer correctness (F1) | This family spans two constructs: perplexity-based metrics map to Language modeling; F1 maps to Fluency and answer correctness. |
| 3.4.4 Statistical correlation (Pearson Correlation) | Measures linear association between paired evaluation signals | Semantic similarity and alignment | Pearson Correlation is used as an alignment indicator and grouped under Semantic similarity and alignment rather than treated as a separate aggregation construct. |
| 3.4.5 Human-readability inspired metrics (B-RT suite) (B-RT.fluency, B-RT.relevance, B-RT.coherence, B-RT.consistency, B-RT.average) | Multi-aspect readability and quality signals not confined to one dimension | Split mapping across constructs (by component) | B-RT is a metric suite. Each component is mapped to the construct it operationalizes: relevance/average → Semantic similarity and alignment; fluency → Fluency and answer correctness; coherence/consistency → fluency/coherence signals within the same construct. |

### 3.5.1 Composite Performance Score (CPS) Formulation

The Composite Performance Score aggregates normalized metric values using a weighted sum. For a given query *q* evaluated under model *m* at similarity threshold *t*, CPS is computed as:

$$CPSq = \sum_{i=1}^{n} w_i \times \left[ d_i \frac{(m_{i,q} - min_i)}{(max_i - min_i)} + \frac{(1 - d_i)}{2} \right] \quad (3.27)$$

where:

- *m_i,q* denotes the raw value of metric *i* for query *q*
- *min_i* and *max_i* are the observed minimum and maximum values for metric *i* across the evaluation set
- *d_i* ∈ {−1, +1} is the polarity indicator: *d_i* = +1 if higher values indicate better performance, *d_i* = −1 if lower values indicate better performance
- *w_i* is the weight assigned to metric *i*, with Σ*w_i* = 1
- *n* is the number of metrics included in the composite

**Normalization by polarity.** For metrics where higher values indicate better performance (*d_i* = +1), the normalization follows standard min-max scaling:

$$Normalized\ value = \frac{m_{i,q} - min_i}{max_i - min_i} \quad (3.28)$$

For metrics where lower values indicate better performance (*d_i* = −1), the normalization is inverted:

$$Normalized\ value = \frac{max_i - m_{i,q}}{max_i - min_i} \quad (3.29)$$

This transformation ensures that all normalized values fall within [0, 1] and that higher normalized values consistently indicate better performance, regardless of the original metric polarity.

Aggregation across queries. The mean CPS for model m at similarity threshold t over all *Q* queries is:

$$\mu_{m,t} = \frac{1}{Q} \sum_{q=1}^{Q} CPS_q^{(m,t)} \quad (3.30)$$

This aggregated score enables comparison of model-threshold configurations on a common scale.

### 3.5.2 Threshold-Aware Composite Performance Score (T-CPS)

CPS captures mean performance but does not reflect consistency across test instances. A configuration that achieves high mean CPS but exhibits large variance across queries may be less reliable in deployment than a configuration with slightly lower mean but stable outputs. The Threshold-aware Composite Performance Score (T-CPS) addresses this limitation by incorporating a reward-penalty structure based on the coefficient of variation (CV).

The *CV* is defined as the ratio of standard deviation to mean [137].

$$CV_{m,t} = \frac{\sigma_{m,t}}{\mu_{m,t}} \quad (3.31)$$

where:

- $\sigma_{m,t}$ is the standard deviation of CPS scores for model m at similarity threshold t
- $\mu_{m,t}$ is the mean CPS for model m at similarity threshold t

The T-CPS is formulated as:

$$T - CPS = \mu \times (1 + \alpha \times (1 - CV)) - \beta \times CV^2 \quad (3.32)$$

- $\alpha$ defines the reward coefficient for stable configurations
- $\beta$ defines the penalty coefficient for high variability

The reward term $(1 + \alpha \times (1 - CV_{m,t}))$ increases scores for configurations with low variability. The penalty term $\beta \times CV_{m,t}^2$ applies a quadratic reduction for configurations with high variability, penalizing inconsistent behavior more severely as CV increases.

**Theoretical Foundation.** This formulation follows the principle that evaluation should account for dispersion in addition to mean performance when consistency matters, consistent with statistical quality control and risk-adjusted evaluation [137], [138].

**Parameter selection.** The reward weight $\alpha = 0.1$ and penalty weight $\beta = 0.05$ were selected to enforce an asymmetric treatment of consistency and variability. Under this setting, the consistency reward contributes up to approximately +10% relative to the base CPS when variability is low, while the variability penalty can reduce scores by up to approximately −5% under the maximum observed coefficient of variation in the experiments. This 2:1 asymmetry reflects a design choice that favors configurations achieving stable behavior without imposing an overly harsh penalty on exploratory configurations that may exhibit higher variance. These parameter values are not claimed as optimal; rather, they represent reasonable starting points informed by the magnitude of CPS variation observed in preliminary runs. The sensitivity of T-CPS rankings to alternative $\alpha$ and $\beta$ values is examined below to assess whether conclusions depend on these specific choices. This reflects the risk-adjusted evaluation principle that performance should be assessed alongside variability, as formalized by Sharpe-style measures [138].

**Interpretation.** T-CPS favors configurations that combine high mean performance with low variability. When CPS and T-CPS identify the same peak-performing similarity threshold, the configuration offers both strong mean performance and reliable consistency.

**Sensitivity analysis.** To verify that conclusions do not depend on specific parameter choices, sensitivity analysis was conducted across 25 parameter combinations ($\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25\} \times \beta \in \{0.025, 0.05, 0.075, 0.10, 0.15\}$). T-CPS was recalculated for all 31 configurations for each parameter combination. Correlation analysis showed that $\alpha$ exhibited near-perfect positive correlation with T-CPS (mean $r = 0.9993$; all $p < 0.001$), whereas $\beta$ showed negligible correlation (mean $r = -0.034$; not significant). Variance decomposition indicated that $\alpha$ explained 99.87% of the variance in T-CPS, while $\beta$ explained 0.13%. Multiple regression yielded $R^2 = 1.000$. Ranking stability analysis showed that 29 of 31 configurations (93.5%) exhibited no change in rank across all 25 parameter combinations; two configurations fluctuated by one position between adjacent ranks, while the top positions remained stable. These results indicate that although absolute T-CPS values are sensitive to $\alpha$, the relative ordering of configurations used for practical recommendations remains stable within the tested range, and the dominance of $\alpha$ suggests that the consistency reward is the primary driver of T-CPS differentiation while the variability penalty plays a secondary moderating role. This sensitivity analysis was previously reported in the multi-agent study and is summarized here for completeness [18].

### 3.5.3 Statistical Significance Testing

To determine whether observed differences between threshold configurations and baseline are statistically meaningful, each configuration undergoes statistical evaluation using paired t-tests. Two-tailed paired t-tests are applied with pairing defined at the per-question level (the same question across baseline and threshold conditions). The paired design compares CPS values for the same queries across baseline and threshold conditions, controlling for query-specific variation.

**Significance testing.** For each model-threshold combination, a paired t-test compares the CPS distribution against the baseline configuration across all queries. Significance is evaluated at $\alpha = 0.05$, with results reported using conventional notation: * for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$. Reported p-values are uncorrected; the rationale is discussed below.

**Effect size.** To complement significance testing, effect sizes are assessed using Cohen's d for paired samples:

$$d = \frac{M_{diff}}{SD_{diff}} \tag{3.33}$$

where $M_{diff}$ is the mean of the per-question CPS differences (threshold minus baseline) and $SD_{diff}$ is the standard deviation of those differences. This formulation matches the paired experimental design where the same questions are evaluated under both conditions. Effect sizes are classified following standard conventions [139]: negligible ($d < 0.2$), small ($0.2 \leq d < 0.5$), medium ($0.5 \leq d < 0.8$), or large ($d \geq 0.8$).

**Confidence intervals.** 95% confidence intervals are computed for the mean per-question CPS difference (threshold minus baseline) to establish bounds on the expected improvement.

**Multiple comparisons consideration.** Phases III and IV involve multiple statistical tests (Phase III: 4 models × 10 thresholds = 40 comparisons; Phase IV: same structure). At $\alpha = 0.05$, approximately 2 significant results per phase may occur by chance under the null hypothesis of no threshold effect. Reported p-values are therefore uncorrected to preserve statistical power for detecting genuine effects, and significance is interpreted at the pattern level—consistency across thresholds within models, coherence across phases, and alignment with effect sizes—rather than as definitive evidence from any single comparison. The implications of this exploratory approach for result interpretation are discussed in Section 5.3.3.

### 3.5.4 Balance Score (Stability–Performance Ratio)

While T-CPS incorporates stability directly into the composite score, deployment decisions often require an explicit criterion that expresses the trade-off between improvement magnitude and output variability. For this purpose, a derived ratio termed the Balance Score is defined, which quantifies how much stability-adjusted improvement is obtained per unit of variability [18].

Let $T\text{-}CPS\%_{m,t}$ denote the percentage improvement of T-CPS for model *m* at similarity threshold *t* relative to a baseline retrieval configuration (the baseline used in Chapter 4 experiments). The Balance Score is defined as:

$$Balance\ Score_{m,t} = \frac{(T-CPS improvement\ \%_{m,t}/100)}{CV_{m,t}} \quad (3.34)$$

where $CV_{m,t}$ is the coefficient of variation of CPS at *(m,t)*, defined in Section 3.5.2. as the ratio of standard deviation to mean.

This formulation uses the improvement percentage expressed as a proportion: T-CPS improvement % is divided by 100 before normalization by *CV.* The Balance Score therefore remains dimensionless and numerically comparable across experiments and datasets.

**Interpretation.** Higher Balance Scores indicate configurations that provide larger stability-adjusted gains per unit of variability. Configurations with large improvements but also high variability produce lower Balance Scores than configurations with moderate improvement and low variability, reflecting a preference for predictable behavior when selecting retrieval parameters for practical use. The Balance Score does not replace CPS or T-CPS; it complements them by offering a direct selection criterion when both improvement and consistency matter. In Chapter 4, Balance Score is applied to Phase III and Phase IV threshold sweeps to identify threshold settings that deliver meaningful gains without unstable output behavior.

## 3.6 Chapter Summary

The evaluation framework for RAG was operationalized through model selection and metric definition. Sections 3.2 and 3.3 documented seven language models spanning the 7B–8B parameter range: Mistral 7B, Llama 2 7B, and Orca 2 7B (initial set) representing efficiency-oriented design, general-purpose capability, and reasoning specialization respectively; Granite 3.2

8B, DeepSeek R1 8B, Llama 3.1 8B, and Mistral 7B v0.3 (updated set) reflecting the revised model lineup used for later evaluation stages, including cross-version continuity for Mistral and the replacement of earlier baselines.

Section 3.4 formalized evaluation metrics across five categories: lexical overlap (METEOR, ROUGE, BLEU), semantic similarity (Cosine Similarity, BERTScore), fluency and answer quality (Laplace and Lidstone perplexity, F1 Score), statistical correlation (Pearson Correlation), and human-readability proxies (B-RT suite). Together, these twenty-four metrics provide complementary perspectives on retrieval integration, semantic fidelity, and generation quality. For CPS aggregation, these families are regrouped into four evaluation constructs; a nine-metric subset is used to avoid double-weighting correlated variants.

Section 3.5 introduced the Composite Performance Score (CPS) and Threshold-aware Composite Performance Score (T-CPS) as aggregation methods that combine individual metrics into unified indices for systematic comparison. CPS addresses the challenge of multi-metric evaluation through normalized weighted summation, while T-CPS extends this formulation with a stability term that penalizes high output variability. Statistical evaluation using paired t-tests, effect size analysis, and confidence intervals ensures that reported improvements reflect genuine performance differences. The Balance Score provides a derived selection criterion for identifying configurations that balance improvement magnitude with output consistency.

This framework contributes to Deficiency 1 by defining threshold-aware evaluation through composite scoring (Objective 3) and to Deficiency 3 by providing model selection criteria and evaluation procedures for open-source deployments (Objectives 2 and 3), and is applied in Chapter 4 to systematic similarity threshold variation and comparative model analysis.

# CHAPTER 4: EXPERIMENTAL EVALUATION AND RESULTS

The experimental evaluation of RAG systems was conducted using the PaSSER platform described in Chapter 2. The evaluation metrics defined in Chapter 3 were applied to examine how similarity threshold configuration influences generation quality across multiple open-source LLMs and two knowledge domains. The results in this chapter address the three research questions defined in the Introduction: (RQ1) whether varying the similarity threshold produces measurable changes in generation quality, (RQ2) whether these effects differ across models, and (RQ3) whether comparable similarity threshold ranges hold across knowledge domains. These experiments address Objective 4 (controlled testing and analysis) and all three components of Objective 1: the evaluation procedure (a), the infrastructure (b), and the experimental design (c), addressing Deficiency 1 (threshold-aware evaluation), Deficiency 2 (reproducibility infrastructure), and Deficiency 3 (practical guidance for open-source deployments).

The experimental program is structured in four sequential phases of increasing scope, reported in Sections 4.1–4.4. Section 4.1 (Phase I) establishes end-to-end system execution, profiles runtime and output behavior for three 7B-parameter models across two hardware environments under fixed top-k retrieval. Section 4.2 (Phase II) introduces controlled similarity threshold variation within the 0.50–0.80 range and applies the CPS defined in Section 3.5 to aggregate multi-metric outcomes. Section 4.3 (Phase III) expands the model set to newer 8B-parameter architectures and applies T-CPS to capture both performance and output consistency across thresholds. Section 4.4 (Phase IV) extends the analysis to a biodiversity corpus distinct from the agricultural domain used in earlier phases.

Across all phases, experimental configurations were recorded through blockchain-based provenance logging to support reproducibility and independent verification. The agricultural corpus comprises 446 question-answer pairs, with phase-specific subsets used in Phase II (101 pairs) and Phase III (369 pairs). The biodiversity corpus comprises 426 question-answer pairs. Both corpora were constructed from authoritative sources including European Union regulations, Food and Agriculture Organization (FAO) technical documents, and international conventions. All models were accessed through the Ollama API to maintain consistent inference conditions, and evaluation metrics were computed using the implementations documented in Chapter 3.

## 4.1. Phase I: System Testing and Runtime Profiling

Phase I assesses end-to-end system functionality, profiles runtime and quality characteristics by executing the complete RAG pipeline on a single-domain corpus, evaluating the initial set of three 7B-parameter open-source LLMs across two hardware environments [17]. Retrieval therefore operated in Normal Mode with K = 100 (fixed top-k). Temperature was set to 0.2 for all generation runs. Two aspects are examined: (1) system correctness and reproducibility across ingestion, retrieval, generation, metric computation, export, and blockchain logging; and (2) runtime and output quality under a common practitioner configuration where top-k retrieval is used without similarity threshold filtering.

The dual-hardware design separates throughput effects from model-level behavior, providing evidence on framework portability across execution environments. The quality metrics analysis reports model-specific differences in output quality under fixed top-k retrieval.

Phase I establishes platform functionality prior to threshold experimentation and does not directly address RQ1–RQ3; threshold-aware evaluation begins in Phase II. This phase supports verification of the infrastructure component (b) of Objective 1.

### 4.1.1 Experimental Design

**Dataset Preparation.** The experimental corpus was constructed from two authoritative sources in the agricultural and regulatory domain: Regulation (EU) 2018/848 on organic production [140] and the FAO Climate-Smart Agriculture Sourcebook [141]. Both texts were manually preprocessed to remove formatting artifacts, headers, footers, and non-informative content before segmentation. Document chunking used the fixed parameters specified in Section 2.2.2 (1024 characters, overlap 50 characters). The resulting segments were embedded using the Mistral 7B embedding model and stored as vector collections in ChromaDB.

The evaluation dataset comprises 446 question-answer pairs stored in standardized JSON format. Reference answers were extracted directly from the source documents as complete, self-contained passages. Questions were generated by prompting Mistral 7B to generate an appropriate query for each reference answer using a structured instruction that directed the model to produce concise, clear questions directly related to regulatory content without implying the answer.

This design supports scalable dataset construction but introduces a fairness consideration: one evaluated model contributes to question generation. The same fixed question set was applied unchanged to all models under identical retrieval settings, and reference answers are sourced from documents rather than generated. The potential influence of model-assisted question generation is acknowledged as a limitation (Section 5.3.2); Phase IV addresses this by using an alternative model (Claude Opus) for question generation on the biodiversity corpus.

**Hardware Configurations.** Two execution environments were used without changing the software stack. The Mac M1 profile is an Apple Mac Mini with M1 system-on-chip (8 CPU cores, 10 GPU cores) and 16 GB RAM, running macOS 13.4. The Ubuntu Server profile is an Intel Xeon system (32 cores, 128 GB RAM) running Ubuntu 22.04 under CPU-only inference. These configurations represent different computational settings, one with local GPU acceleration available and one CPU-only, enabling comparison of runtime and output characteristics under varied hardware conditions. End-to-end system checks (execution correctness, export integrity, and blockchain logging) are reported in Section 4.1.5.

**Test Procedures and Metrics.** Two complementary test procedures were executed over the 446-item dataset:

The RAG Q&A Score Test evaluates generated answers against reference answers using 16 metrics, representing the first implemented subset of the full 24-metric suite at this phase: METEOR, BLEU, ROUGE-1 (recall, precision, F-score), ROUGE-2 (recall, precision, F-score), ROUGE-L (recall, precision, F-score), Laplace Perplexity, Lidstone Perplexity, cosine similarity, Pearson correlation, and F1 Score. Definitions of these evaluation metrics are provided in Chapter 3. For all Phase I runs, generation temperature was fixed at 0.2.

In parallel, the Timing Performance Test records runtime indicators produced during execution of the same test set, including Evaluation Time, Load Duration, Prompt Evaluation Count, Prompt Evaluation Duration, Total Duration, and Tokens/Second (Table 4.1). These indicators characterize environment-dependent execution behavior across hardware profiles rather than generation quality.

Outputs from both procedures were exported to spreadsheet format and recorded on-chain via smart contract actions, providing a durable trace of configurations and results as described in Section 2.2.3. Performance data are available in the public project repository [142].

**Configuration Rationale.** Phase I confirms system execution under the fixed retrieval and chunking configuration used across all phases. Retrieval operated in fixed top-k mode because similarity threshold filtering was not available in PaSSER at this stage. Results in this phase are reported under fixed top-k retrieval, before similarity threshold filtering is introduced in Phase II.

### 4.1.2 Timing Performance Results

Table 4.1 presents timing indicators for all three models across both hardware configurations, derived from the 446-question test set.

*Table 4.1 Timing Performance Summary Across Models and Hardware Environments. Reproduced from [17].*

| Metric | Llama 2 7B macOS/M1 | Llama 2 7B Ubuntu | Mistral 7B macOS/M1 | Mistral 7B Ubuntu | Orca 2 7B macOS/M1 | Orca 2 7B Ubuntu |
|---|---|---|---|---|---|---|
| Evaluation Time (sec.) | 51,613 | 115,176 | 35,864 | 45,325 | 24,759 | 74,431 |
| Evaluation Count | 720 | 717 | 496 | 284 | 350 | 471 |
| Load Duration (sec.) | 0.025 | 0.043 | 0.016 | 0.039 | 0.037 | 0.045 |
| Prompt Eval. Count | 51 | 68 | 47 | 54 | 53 | 96 |
| Prompt Eval. Duration (sec.) | 0.571 | 5.190 | 0.557 | 4.488 | 0.588 | 6.955 |
| Total Duration (sec.) | 52,211 | 120,413 | 36,440 | 49,856 | 25,387 | 81,434 |
| Tokens/Second | 14.07 | 6.30 | 13.91 | 6.36 | 14.38 | 6.53 |

The Mac M1 environment completes evaluation substantially faster than the Ubuntu Server environment across all three models, achieving approximately 2.2× higher throughput (tokens per second). This difference is consistent with hardware-accelerated inference being available on the Mac profile, while the Ubuntu profile performs CPU-only inference.

Across models, the environment gap differs by model: Llama 2 7B is approximately 2.2× slower on Ubuntu, Mistral 7B is approximately 1.3× slower, and Orca 2 7B is approximately 3.0× slower, indicating different sensitivity to environment-specific execution characteristics. Orca 2 7B achieves the shortest evaluation time on the Mac M1 system (24,759 seconds).

The Ubuntu environment reports a higher prompt evaluation count for Orca 2 7B (96 versus 53). Phase I does not instrument internal batching or scheduling behavior; this difference is therefore noted as an implementation-level observation rather than interpreted as a model capability difference.

### 4.1.3 Quality Metrics Results

Table 4.2 presents mean quality metric values for all three models under fixed top-k retrieval, evaluated across the 446 question-answer pairs.

***Table 4.2*** *Mean Quality Metric Values Across Models.*

| Metric | Llama 2 7B | Mistral 7B | Orca 2 7B |
|---|---|---|---|
| METEOR | 0.248 | **0.271** | 0.236 |
| BLEU | 0.026 | **0.032** | 0.021 |
| ROUGE-1 Recall | 0.146 | **0.161** | 0.122 |
| ROUGE-1 Precision | 0.499 | 0.472 | **0.503** |
| ROUGE-1 F-score | 0.207 | **0.224** | 0.184 |
| ROUGE-2 Recall | 0.045 | **0.050** | 0.035 |
| ROUGE-2 Precision | 0.197 | 0.175 | **0.200** |
| ROUGE-2 F-score | 0.065 | **0.070** | 0.055 |
| ROUGE-L Recall | 0.131 | **0.143** | 0.108 |
| ROUGE-L Precision | 0.455 | 0.424 | **0.457** |
| ROUGE-L F-score | 0.186 | **0.199** | 0.163 |
| Laplace Perplexity (lower is better) | **52.992** | 53.060 | 53.083 |
| Lidstone Perplexity (lower is better) | 46.935 | **46.778** | 56.940 |
| Cosine similarity | 0.728 | **0.773** | 0.716 |
| Pearson correlation | 0.843 | **0.861** | 0.845 |
| F1 Score | 0.178 | **0.219** | 0.153 |

*† Phase I results were first reported in [17]; Table 4.2 reproduces the Phase I metric summary as computed from the PaSSER exports.*

Mistral 7B attains the highest values on most overlap and similarity metrics, including METEOR, BLEU, ROUGE recall and F-scores, cosine similarity, Pearson correlation, and F1 Score. Orca 2 7B attains the highest ROUGE precision values (ROUGE-1/2/L precision). Llama 2 7B attains

the lowest Laplace perplexity; as defined in Chapter 3, perplexity reflects n-gram typicality and is interpreted alongside overlap and similarity measures.

### 4.1.4 Analysis and Interpretation

The timing results demonstrate a consistent throughput advantage for the Mac M1 environment, while the Ubuntu profile performs CPU-only inference. The Mac M1 achieved 13.91–14.38 tokens per second across models compared to 6.30–6.53 tokens per second on Ubuntu Server, translating directly to faster completion times for interactive scenarios.

Load duration showed minimal difference between environments (0.016–0.045 seconds), indicating that model initialization is not the performance bottleneck. The throughput difference is therefore primarily attributable to inference computation rather than setup overhead.

**Model Performance Patterns.** Within the evaluated scope, Mistral 7B shows the strongest overall alignment to the reference answers across the overlap- and similarity-oriented metrics. It leads on METEOR and BLEU, on ROUGE recall-oriented measures, and on the ROUGE F-scores that combine recall and precision, indicating that its responses preserve more of the reference content while maintaining competitive precision. Mistral 7B also achieves the highest cosine similarity and Pearson correlation values, indicating closer embedding-space alignment to the reference answers under the fixed top-k retrieval configuration used in Phase I. These differences are reported descriptively and are not attributed to specific model design factors in this phase; establishing causation would require controlled ablations beyond the experimental scope.

Orca 2 7B exhibits a different profile: it achieves the highest ROUGE precision values (including ROUGE-1, ROUGE-2, and ROUGE-L precision), indicating that when Orca's generated content overlaps with the references, it tends to include fewer additional tokens not supported by the reference. At the same time, its ROUGE recall and ROUGE F-scores are lower than Mistral 7B, which is consistent with producing answers that capture a smaller fraction of the reference content under the same retrieval configuration.

Llama 2 7B shows the strongest perplexity-based typicality signals, achieving the lowest Laplace perplexity and near-lowest Lidstone perplexity values. Because perplexity reflects n-gram typicality rather than factual grounding, these results are interpreted alongside overlap and semantic similarity measures, where Llama 2 7B is consistently below Mistral 7B. Taken together,

Phase I indicates that overlap and embedding similarity metrics favor Mistral 7B, ROUGE precision favors Orca 2 7B, and perplexity-based indicators favor Llama 2 7B under fixed top-k retrieval.

### 4.1.5 End-to-End System Check

Phase I assesses PaSSER's end-to-end functionality across the complete RAG pipeline:

- **Data ingestion:** All 446 question-answer pairs were successfully chunked, embedded, and stored in ChromaDB without errors.
- **Retrieval execution:** All 446 questions were processed through vector similarity search without failures across both hardware environments.
- **Model inference:** 1,338 generation cycles (446 questions × 3 models) completed successfully via the Ollama API.
- **Metric computation:** All 16 Phase I metrics (the first implemented subset of the 24-metric suite) were computed for each model without failures.
- **Export and logging:** Results were successfully exported to spreadsheet format and recorded on the Antelope blockchain via smart contract actions.

Cross-environment comparison shows that quality metric scores are comparable between Mac M1 and Ubuntu Server configurations despite substantial timing differences. This indicates that, within the evaluated setups, quality outcomes can be treated primarily as model- and retrieval-dependent rather than environment-driven.

### 4.1.6 Phase I Summary

Phase I establishes PaSSER as a functional and reproducible evaluation environment and reports runtime and quality outcomes for three 7B-parameter models under fixed top-k retrieval. Phase I reports mean scores; variance analysis using the T-CPS framework is introduced in Phase III.

1. **Hardware environment dominates throughput.** The Mac M1 environment achieves approximately 2.2× higher throughput than Ubuntu Server under CPU-only inference, with practical implications for interactive versus batch deployment scenarios.
2. **Model choice affects alignment metrics under fixed retrieval.** Mistral 7B demonstrates higher scores across most lexical and semantic metrics.

3. **Orca 2 7B achieves the highest ROUGE precision values (ROUGE-1/2/L precision),** while Mistral 7B leads on the corresponding ROUGE F-scores.
4. **Llama 2 7B achieves the lowest Laplace perplexity.** Perplexity-based indicators are interpreted alongside overlap and similarity metrics rather than as a substitute for content alignment.

With system correctness established and runtime and quality patterns characterized, Phase II introduces similarity threshold filtering to examine how retrieval selectivity affects output quality.

## 4.2 Phase II: Similarity Threshold and CPS

Phase II was conducted as a pilot to characterize similarity threshold sensitivity under a constrained configuration and to assess CPS aggregation prior to the expanded analyses in Phases III–IV [16]. Score Mode parameters were set to K = 100 (maximum retrieved passages) and K-Inc = 2 (retrieval step size); temperature remained at 0.2. The similarity threshold was swept from 0.50 to 0.80 in 0.05 increments to quantify how retrieval selectivity influences generation quality under Score Mode (Section 2.2.2), where a minimum similarity threshold is applied before passage selection. Performance was aggregated using CPS across nine evaluation metrics (Section 4.2.2) for three 7B-parameter LLMs: Mistral 7B, Llama 2 7B, and Orca 2 7B. In total, 2,121 evaluations were executed (3 models × 7 thresholds × 101 questions) on an Apple Mac mini (M1, 16 GB RAM). Within Score Mode, lower similarity thresholds may admit weakly related passages, while higher similarity thresholds may increase retrieval sparsity and reduce evidence coverage. The retained interval (0.50–0.80) therefore spans the practical transition from permissive to selective retrieval for the Phase II pilot analysis. Extended analysis covering thresholds up to 0.95 was reported in [16]. Phases III–IV extend the sweep to 0.95 to characterize threshold sensitivity under more selective retrieval conditions and to test whether CPS trends observed in the Phase II pilot persist when evidence availability becomes sparse.

Phase II addresses RQ1 (similarity threshold effects on generation quality) within the pilot configuration, applying the evaluation procedure (a) and experimental design (c) components of Objective 1 and contributing to Objective 4 (controlled testing and analysis).

### 4.2.1 Experimental Design

**Dataset and Corpus.** A subset of 101 question–answer pairs was selected from the 446-pair dataset developed in Phase I (Section 4.1). Chunking, embedding, and vector store configurations remained identical to Phase I: document segmentation used the parameters specified in Section 2.2.2 (1024 characters, overlap 50 characters), with segments embedded using the Mistral 7B embedding model and stored in ChromaDB.

**Hardware Configuration.** Phase II used the same two execution environments described in Section 4.1.1 (Mac M1 and Intel Xeon).

**Test Procedures and Metrics.** Score Mode retrieval was introduced in Phase II, enabling similarity threshold–based filtering. Per-question exports were retained only for thresholds 0.50–0.80, bounding all Phase II analyses to that range. Generation temperature was fixed at 0.2. Metric export labeling was standardized for Phase II and subsequent phases to support consistent interpretation of precision, recall, and F-score variants across configurations.

### 4.2.2 CPS Weighting Scheme

CPS is introduced as a weighted aggregation of nine evaluation metrics for similarity threshold comparison, as established in [16] and defined in Section 3.5 The Phase II metric panel was selected to cover the four CPS construct families defined in Section 3.5: lexical overlap (METEOR, BLEU, ROUGE-1 F-score, ROUGE-L F-score), semantic similarity and alignment (Cosine Similarity, Pearson Correlation), fluency and correctness (F1 Score), and language modeling (Laplace Perplexity, Lidstone Perplexity). Table 4.3 presents the metric weights used in Phase II.

The selected panel is intentionally compact, implemented from the Phase I metric set, and balanced across constructs to avoid over-reliance on any single metric family. This reflects the general principle that aggregate scores benefit from combining complementary, weakly-correlated measures rather than multiple highly-correlated ones, which can amplify redundancy and bias in the combined index [143]. Lexical overlap metrics provide sensitivity to content retention against the reference answers, while the inclusion of both METEOR and BLEU mitigates single-metric bias by capturing complementary recall- and precision-oriented behavior. Cosine Similarity and Pearson Correlation contribute alignment metrics beyond surface overlap, supporting detection of semantic drift when retrieval selectivity changes. Perplexity metrics

provide a reference-independent fluency signal that can vary with evidence availability under different thresholds, and F1 Score contributes an explicit correctness-oriented indicator within the implemented metric set. This combination supports pilot-level detection of threshold effects while keeping computation feasible for repeated evaluation across models and thresholds.

***Table 4.3*** *CPS Metric Panel and Weighting Scheme (Phase II). Reproduced from [16].*

| Metric | Weight | Rationale |
|---|---|---|
| METEOR | 0.15 | Overall assessment of text quality |
| ROUGE-1 F-score | 0.075 | Different levels of text similarity overlap |
| ROUGE-L F-score | 0.075 | Different levels of text similarity overlap |
| BLEU | 0.15 | Overall assessment of text quality |
| Laplace Perplexity | 0.075 | Predicts performance and accuracy (lower is better) |
| Lidstone Perplexity | 0.075 | Predicts performance and accuracy (lower is better) |
| Cosine Similarity | 0.10 | Measures relevance and retrieval correlation |
| Pearson Correlation | 0.10 | Measures relevance and retrieval correlation |
| F1 Score | 0.20 | Most comprehensive and impactful metric |
| TOTAL | 1.00 | |

F1 Score received the highest weight (20%) because it directly reflects answer correctness through the precision-recall balance. METEOR and BLEU received substantial weights (15% each) because they capture complementary overlap behavior, with METEOR emphasizing recall-oriented matching (including synonym handling) and BLEU emphasizing precision-oriented matching with brevity penalty. Cosine Similarity and Pearson Correlation received moderate weights (10% each) to provide alignment signals that complement lexical overlap under changing retrieval selectivity. ROUGE variants and perplexity metrics received lower weights (7.5% each) to preserve construct balance and avoid overweighting overlap- or fluency-oriented signals within the pilot CPS. Laplace Perplexity and Lidstone Perplexity are negative-polarity metrics; after min-max normalization, their values are inverted so that lower perplexity yields a higher contribution to CPS, consistent with Section 3.5.1.

### 4.2.3. Results: Threshold Effects on Composite Performance

Results are reported descriptively for Phase II; statistical significance testing is not applied in this pilot phase. Table 4.4 presents CPS values for the three models across similarity thresholds 0.50–0.80. Within this pilot range, peak CPS was observed at threshold 0.55 for Mistral 7B and Llama 2 7B, and at threshold 0.65 for Orca 2 7B.

***Table 4.4*** *CPS across similarity threshold values 0.50–0.80 (0.05 increments) for Mistral 7B, Orca 2 7B, and Llama 2 7B (Phase II pilot, Score Mode). Reported using the Phase II configuration described in [16].*

| Similarity Threshold | Mistral 7B CPS | Orca 2 7B CPS | Llama 2 7B CPS |
|---|---|---|---|
| 0.5 | 0.463 | 0.483 | 0.459 |
| 0.55 | **0.488** | 0.488 | **0.492** |
| 0.6 | 0.463 | 0.505 | 0.411 |
| 0.65 | 0.460 | **0.527** | 0.484 |
| 0.7 | 0.468 | 0.500 | 0.490 |
| 0.75 | 0.425 | 0.473 | 0.485 |
| 0.8 | 0.463 | 0.518 | 0.476 |

• **Mistral 7B.** The highest CPS within the pilot sweep (0.488) was observed at threshold 0.55. CPS remained broadly stable across 0.50–0.70, followed by a dip at 0.75 (0.425) and recovery at 0.80 (0.463).

• **Orca 2 7B.** The highest CPS within the pilot sweep (0.527) was observed at threshold 0.65. An upward trend was observed from 0.50 to 0.65, with CPS remaining above 0.47 across 0.50–0.80. Orca 2 7B appeared the most stable among the three models in the pilot range.

• **Llama 2 7B.** The highest CPS within the pilot sweep (0.492) was observed at threshold 0.55. Greater variability was observed than for Orca 2 7B, with a notable dip at 0.60 (0.411) and recovery at 0.65–0.70. Results are interpreted only within the retained threshold range of 0.50–0.80.

Figure 4.1 visualizes CPS trends across similarity threshold values 0.50–0.80 for all three models.

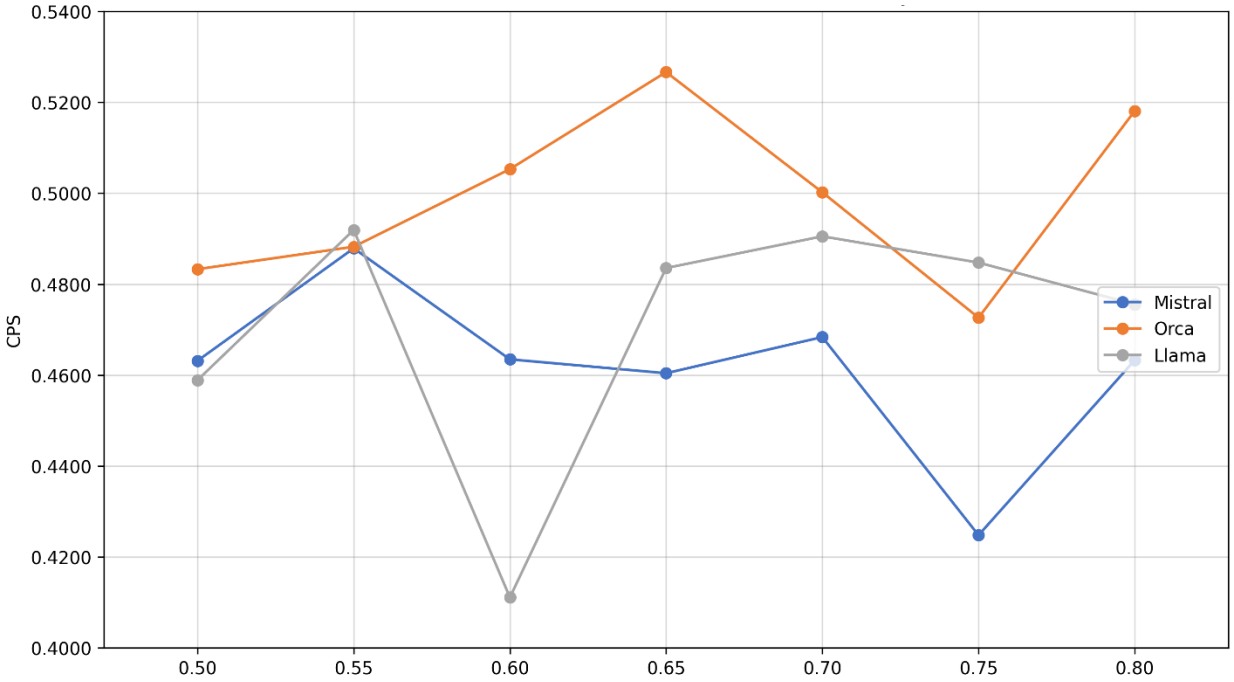


***Figure 4.1*** *CPS values for the three models across similarity threshold values* ***0.50–0.80*** *(Phase II pilot, Score Mode). Adapted from [16].*

**Cross-Model Comparison.** Orca 2 7B appeared to have the most stable CPS profile across the pilot sweep (0.50–0.80). Mistral 7B and Llama 2 7B showed greater threshold sensitivity within this range, with model-specific dips at different thresholds. This observation is descriptive and is tested under extended thresholds and updated model set in later phases.

### 4.2.4 Analysis and Interpretation

The variation in highest-performing similarity thresholds (by CPS) across Mistral 7B, Orca 2 7B, and Llama 2 7B may reflect model-dependent differences documented in retrieval and language modeling literature, including architecture, training data, and context sensitivity [14], [110].

Table 4.5 summarizes hypothesized factors that may affect similarity thresholds for the three models. These characterizations are reproduced from [16] and are presented as explanatory hypotheses; they are not empirically isolated in Phase II.

***Table 4.5.*** *Factors Influencing Similarity Threshold (Phase II Models). Reproduced from [16].*

| Factor | Mistral 7B | Orca 2 7B | Llama 2 7B |
|---|---|---|---|
| Model Architecture and Training Data | Moderate sensitivity due to diverse training data | Balanced architecture and data handling | Requires highly relevant data due to training specificity |
| Context Sensitivity | Benefits from moderately relevant context | Handles broader context well | Needs highly relevant context for best performance |
| Retrieval and Generation Balance | Prefers balance between diversity and relevance | Excels with balanced retrieval and generation | Performs best with highly relevant but less diverse context |
| Noise Handling and Robustness | Robust to some noise, can handle moderate relevance | Moderate noise tolerance | Sensitive to noise, prefers high relevance for quality |
| Generalization Capabilities | Strong generalization, moderate context suffices | Good balance between generalization and specificity | High specificity required |
| Domain and Query Specificity | Performs well across diverse domains | Adaptable to various domains | Requires domain-specific fine-tuning for best results |

Models with stronger noise-handling capabilities may perform well at lower thresholds, filtering irrelevant information during generation. Models that are less tolerant of noisy or weakly related context may require higher thresholds to keep retrieved chunks strongly aligned with the query.

**Model-Specific Observations**

Mistral 7B is reported to handle moderately relevant context effectively, which is consistent with a peak in the lower portion of the retained pilot thresholds. This suggests that moderately selective retrieval can be exploited during generation under the tested configuration.

Orca 2 7B is reported as a reasoning-oriented fine-tuned variant of Llama 2. Its relatively stable CPS profile across the retained pilot thresholds is consistent with a moderate selectivity setting that reduces weakly related context while preserving sufficient evidence coverage for reasoning-oriented generation.

Llama 2 7B showed greater within-range sensitivity in the pilot results, indicating that small changes in similarity threshold can shift the balance between helpful and distracting context under the tested setup.

These interpretations are explanatory rather than causal. The mechanisms linking model design to threshold behavior were not isolated in controlled experiments; confirmation would require targeted ablation and controlled retrieval-noise manipulations.

These observations are bounded by the Phase II pilot scope: a single domain (agriculture), a 101-question subset, fixed preprocessing parameters, and retained per-question artifacts limited to thresholds 0.50–0.80. Generalization beyond this pilot scope is examined in later phases.

### 4.2.5 Phase II Summary

Similarity threshold values were varied from 0.50 to 0.80 across the initial set of three 7B-parameter models to quantify the effect of retrieval selectivity on generation quality under the pilot configuration. The evaluation used 101 question-answer pairs from the agricultural domain, with performance aggregated via the CPS formulation defined in Section 3.5.

1. **Thresholds yielding the highest CPS were model-specific within the pilot sweep.** Peak CPS was achieved at a similarity threshold value of 0.55 for Mistral 7B and Llama 2 7B, and at 0.65 for Orca 2 7B.
2. **Threshold sensitivity varies across models.** Orca 2 7B appeared to have the most stable CPS profile across 0.50–0.80, while Mistral 7B and Llama 2 7B showed greater variability with model-specific dips within the pilot thresholds.

Interpretation is limited to the Phase II pilot scope and the retained threshold range (0.50–0.80). These results provide initial evidence for RQ1 within the pilot configuration; broader threshold behavior and cross-model comparison are examined in Phases III–IV.

## 4.3 Phase III: Model-Dependent Similarity Thresholds

Phase III examines how generation quality changes across open-source LLMs as the similarity threshold is varied in a RAG pipeline, emphasizing model-dependent threshold sensitivity under increasingly selective retrieval. Updating the model set from Phase II (as motivated in Section 3.3), Mistral 7B was retained for continuity (updated to version 0.3), while three 8B-class models (DeepSeek R1 8B, Llama 3.1 8B, Granite 3.2 8B) replaced Llama 2 7B and Orca 2 7B. Similarity thresholds from 0.50 to 0.95 (step 0.05) were evaluated to characterize threshold sensitivity as context becomes more tightly filtered. Generation quality was aggregated using the CPS and T-CPS formulations specified in Sections 3.5.1–3.5.2, with Phase III metric weights defined in Section 4.3.2.

Baseline refers to Normal Mode (top-k retrieval without thresholding). Thresholded runs use Score Mode (threshold-enabled retrieval). All other pipeline settings remained identical to Phase II, including chunking parameters, embedding model, prompt format, and temperature (0.2). The full 24-metric set defined in Section 3.4 was computed for each run.

Phase III addresses RQ1 (similarity threshold effects on generation quality) and RQ2 (model-dependent threshold sensitivity), applying the evaluation procedure (a) and experimental design (c) components of Objective 1 and contributing to Objective 4 (controlled testing and analysis). Six tasks are conducted: (1) evaluate the effects of updating the model set under threshold variations; (2) quantify generation quality across similarity thresholds from 0.50 to 0.95; (3) apply the 9-metric CPS and T-CPS framework consistently across models; (4) apply statistical significance testing to assess observed improvements; (5) compute Balance Score to identify configurations with favorable stability–performance trade-offs; (6) characterize threshold sensitivity patterns across model architectures.

### 4.3.1 Experimental Design

**Dataset and Corpus.** A subset of 369 question–answer pairs from the 446-pair dataset developed in Phase I (Section 4.1) was used for evaluation. Chunking, embedding, and vector store configurations remained identical to previous phases: document segmentation used the parameters specified in Section 2.2.2 (1024 characters, overlap 50 characters), with segments embedded using the Mistral 7B embedding model and stored in ChromaDB.

**Hardware Configuration.** Experiments were conducted across three hardware environments: an M1 Mac Mini, an M2 Mac Mini, and a CPU-only server. The configured context buffer varied across model runs from 2,048 to 10,000 tokens because of environment-specific memory constraints. This heterogeneity reflected practical resource availability over the extended experimental timeline, and its implications for cross-model comparability are discussed in Section 5.3.2. To reduce hardware-related confounding, statistical comparisons were performed within each model relative to its own baseline configuration rather than between models.

**Test Procedures and Metrics.** Each model was evaluated across similarity thresholds from 0.50 to 0.95 (step 0.05), plus a baseline configuration using Normal Mode (top-k retrieval without thresholding), yielding 11 configurations per model. In total, 16,236 evaluations were conducted (4 models × 11 configurations × 369 question–answer pairs). The full 24-metric set defined in Section 3.4 was computed for each run; generation temperature was fixed at 0.2.

### 4.3.2 CPS Weighting Scheme

Phase III applies CPS and T-CPS to compare similarity threshold configurations across the updated model set. The CPS construct families remain consistent with Section 3.5; however, the metric panel and weights were adjusted after Phase II to reflect the expanded evaluation output and to strengthen semantic and fluency coverage under model-dependent variation. Table 4.6 summarizes the metric panel evolution and reports the weights used in Phases III–IV.

The weights reflect the four-construct evaluation framework: (1) lexical overlap (30%) through METEOR and ROUGE metrics, (2) semantic similarity (25%) through BERTScore.f1 and B-RT.average, (3) fluency and accuracy (25%) through F1 score and B-RT.fluency, and (4) language modeling (20%) through perplexity metrics.

Laplace Perplexity and Lidstone Perplexity are negative-polarity metrics; their values are inverted after normalization, consistent with Section 3.5.1. The T-CPS formulation (Section 3.5.2) applies parameters $\alpha = 0.1$ and $\beta = 0.05$.

***Table 4.6*** *Evolution of CPS Metric Panel Across Experimental Phases.*

| Category | Metric | Phase II | Phase III-IV | Change | Rationale |
|---|---|---|---|---|---|
| **Lexical Overlap** | METEOR | 0.150 | 0.150 | — | Core text quality metric retained |
| | BLEU | 0.150 | — | Removed | Redundant with METEOR; brevity penalty less relevant for QA |
| | ROUGE-1.f | 0.075 | — | Replaced | Unigram overlap less informative than bigram |
| | ROUGE-2.f | — | 0.075 | Added | Bigram overlap captures phrase-level similarity |
| | ROUGE-L.f | 0.075 | 0.075 | — | Longest common subsequence retained |
| | **Subtotal** | **0.450** | **0.300** | **–0.150** | |
| **Semantic Similarity** | Cosine Similarity | 0.100 | — | Removed | Replaced by contextual embeddings |
| | Pearson Correlation | 0.100 | — | Removed | Statistical measure less interpretable |
| | BERTScore.f1 | — | 0.125 | Added | Contextual token similarity |
| | B-RT.average | — | 0.125 | Added | Multi-dimensional readability proxy |
| | **Subtotal** | **0.200** | **0.250** | **+0.050** | |
| **Fluency & Accuracy** | F1 Score | 0.200 | 0.150 | −0.050 | Weight redistributed to semantic metrics |
| | B-RT.fluency | — | 0.100 | Added | Explicit fluency assessment |
| | **Subtotal** | **0.200** | **0.250** | **+0.050** | |
| **Language Modeling** | Laplace Perplexity* | 0.075 | 0.100 | +0.025 | Increased emphasis on text predictability |
| | Lidstone Perplexity* | 0.075 | 0.100 | +0.025 | Increased emphasis on text predictability |
| | **Subtotal** | **0.150** | **0.200** | **+0.050** | |
| **TOTAL** | | **1.000** | **1.000** | | |

### 4.3.3 CPS Performance Overview

The effects of varying the similarity threshold from 0.50 to 0.95 on CPS-based performance were examined across four open-source models in the agriculture domain, using 369 question–answer pairs (16,236 total evaluations). Results are organized to summarize overall CPS improvements before examining stability (T-CPS), metric relationships, and statistical significance.

Table 4.7 presents the top CPS improvement configurations for each model. Mistral 7B v0.3 achieved the largest improvement (+4.58% at threshold 0.95), followed by Llama 3.1 8B

(+1.58% at threshold 0.90), Granite 3.2 8B (+1.25% at threshold 0.95), and DeepSeek R1 8B (+1.01% at threshold 0.90).

*Table 4.7 Top CPS Improvement Configurations by Model (Top 3 Agriculture).*

| Model | Rank | Threshold | Mean CPS | Improvement % |
|---|---|---|---|---|
| Mistral 7B v0.3 | 1 | 0.95 | 0.5454 | **4.58** |
| | 2 | 0.9 | 0.5338 | 2.37 |
| | 3 | 0.7 | 0.5325 | 2.11 |
| Granite 3.2 8B | 1 | 0.95 | 0.5182 | **1.25** |
| | 2 | 0.7 | 0.5179 | 1.2 |
| | 3 | 0.8 | 0.5178 | 1.17 |
| Llama 3.1 8B | 1 | 0.9 | 0.508 | **1.58** |
| | 2 | 0.7 | 0.5065 | 1.33 |
| | 3 | 0.55 | 0.5052 | 1.01 |
| DeepSeek R1 8B | 1 | 0.9 | 0.4559 | **1.01** |
| | 2 | 0.95 | 0.455 | 0.8 |
| | 3 | 0.65 | 0.4548 | 0.77 |

Section 4.3.4 evaluates whether these peak CPS configurations remain preferred when stability is incorporated through T-CPS. Full statistical tables and per-threshold descriptive metrics are provided in Appendix B.

### 4.3.4 T-CPS Performance and Stability

Stability-aware results are summarized in Table 4.8, which reports the top T-CPS improvement configurations by model and incorporates stability through the coefficient of variation (CV). The top T-CPS configuration is model-dependent: Mistral 7B v0.3 peaks at threshold 0.95 (T-CPS = 0.5916; +4.54%), Granite 3.2 8B peaks at threshold 0.95 (T-CPS = 0.5628; +1.25%), Llama 3.1 8B peaks at threshold 0.90 (T-CPS = 0.5501; +1.48%), and DeepSeek R1 8B peaks at threshold 0.65 (T-CPS = 0.4961; +0.79%). The T-CPS rankings largely mirror CPS rankings, while CV values highlight differences in output consistency across models.

Across the top configurations, DeepSeek R1 8B exhibits lower variability (CV = 0.085–0.108) than the other models (CV = 0.122–0.148), indicating more stable output quality across queries despite lower absolute performance. Parameter sensitivity and ranking stability for T-CPS are reported in Section 3.5.2.

***Table 4.8*** *Top T-CPS Improvement Configurations by Model (Top 3 Agriculture).*

| Model | Rank | Threshold | T-CPS | T-CPS Impr. % | CV | Interpretation |
|---|---|---|---|---|---|---|
| Mistral 7B v0.3 | 1 | 0.95 | 0.5916 | 4.54 | 0.134 | Large improvement |
| | 2 | 0.9 | 0.5793 | 2.36 | 0.131 | Moderate improvement |
| | 3 | 0.7 | 0.5783 | 2.17 | 0.128 | Moderate improvement |
| Granite 3.2 8B | 1 | 0.95 | 0.5628 | 1.25 | 0.124 | Small improvement |
| | 2 | 0.8 | 0.5625 | 1.2 | 0.122 | Small improvement |
| | 3 | 0.7 | 0.5622 | 1.15 | 0.129 | Small improvement |
| Llama 3.1 8B | 1 | 0.9 | 0.5501 | 1.48 | 0.148 | Small improvement |
| | 2 | 0.7 | 0.549 | 1.26 | 0.142 | Small improvement |
| | 3 | 0.55 | 0.5484 | 1.16 | 0.129 | Small improvement |
| DeepSeek R1 8B | 1 | 0.65 | 0.4961 | 0.79 | 0.085 | Small improvement |
| | 2 | 0.9 | 0.496 | 0.78 | 0.108 | Small improvement |
| | 3 | 0.95 | 0.4958 | 0.74 | 0.093 | Small improvement |

Full per-threshold T-CPS descriptive metrics for each model are provided in Appendix B.

Figure 4.2 compares threshold-wise CPS and T-CPS improvements for each model across the similarity threshold sweep.

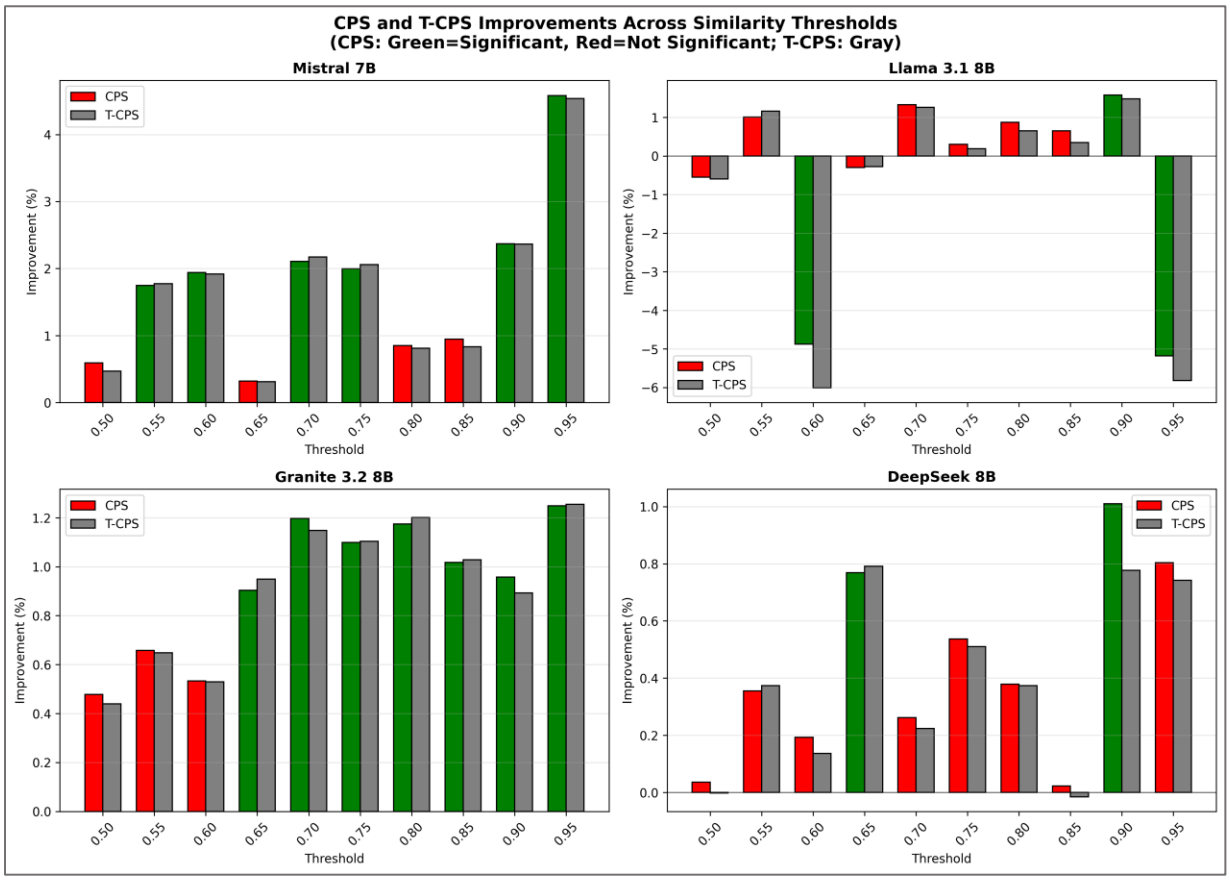


***Figure 4.2*** *Phase III (Agriculture, N = 369): Threshold-wise CPS and T-CPS improvements per model. CPS bars indicate percent improvement relative to baseline and are colored by significance at p < 0.05 (uncorrected). T-CPS bars (gray) indicate stability-aware improvement.*

### 4.3.5 Correlation Analysis

To assess redundancy among component metrics and to examine whether T-CPS provides information beyond mean CPS, correlation analysis was performed. Figure 4.3 presents the Spearman correlation matrix among Phase III component metrics, computed from pooled per-question results across thresholds 0.50–0.95 plus baseline. Table 4.9 reports Spearman correlations between T-CPS and (i) mean CPS and (ii) the coefficient of variation (CV), computed across all model and threshold configurations.

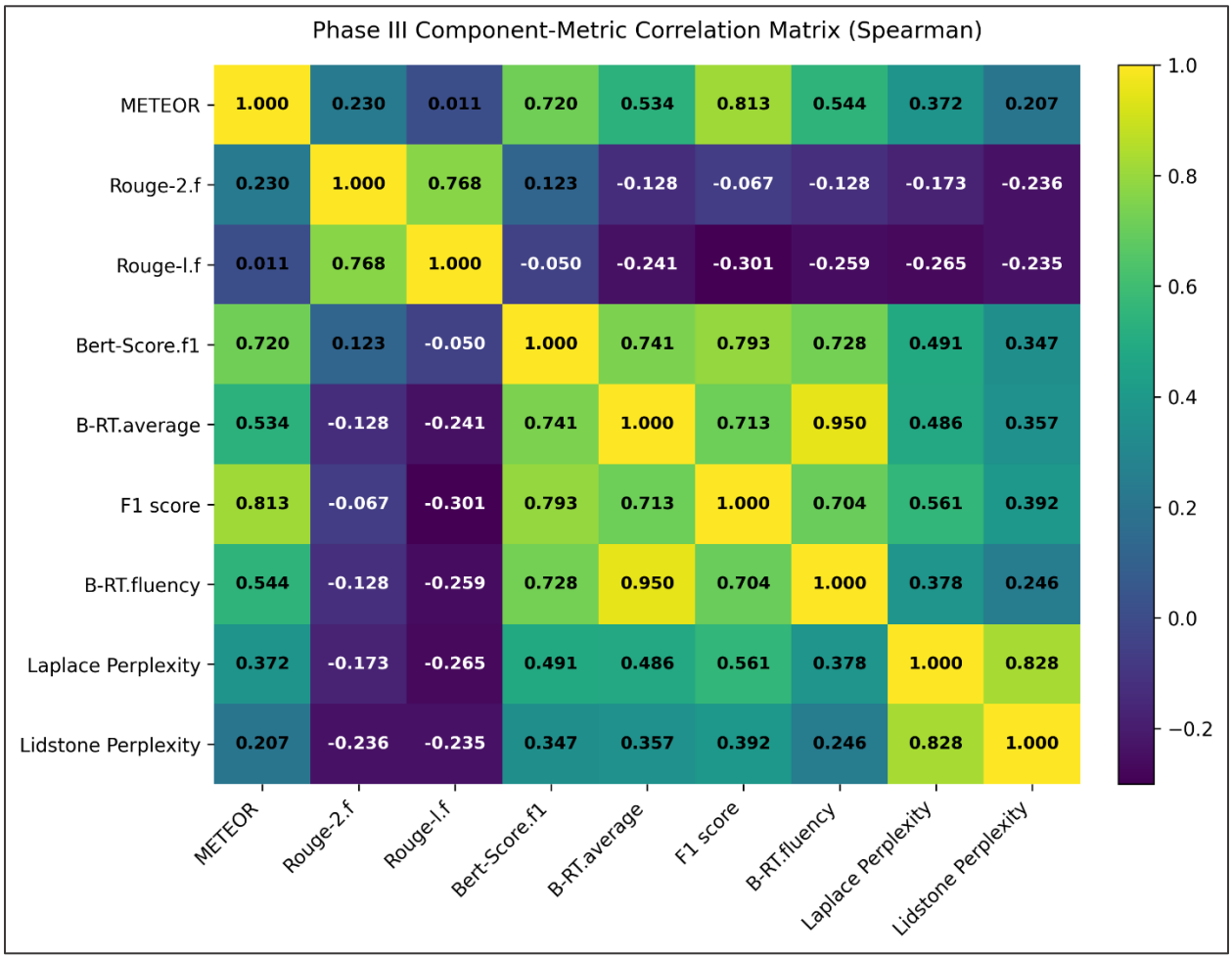


***Figure 4.3*** *Phase III component metric correlation matrix (Spearman), pooled across models and thresholds 0.50–0.95 plus baseline using per-question results.*

Strong clustering persists within the metric panel. The tightest coupling was observed between B-RT.average and B-RT.fluency (ρ ≈ 0.950), Laplace and Lidstone perplexity (ρ ≈ 0.828), and ROUGE-2.f and ROUGE-L.f (ρ ≈ 0.768). METEOR aligned most strongly with F1 score (ρ ≈ 0.813) and BERTScore.f1 (ρ ≈ 0.720), indicating that lexical overlap remains closely associated with semantic similarity under the Phase III scoring panel. Within-group correlations are interpreted as reinforcing evidence of consistent measurement rather than independent signals; the CPS aggregation is designed to stabilize conclusions against noise in any single component metric.

***Table 4.9*** *Phase III associations between stability-aware scoring T-CPS, mean CPS, and variability CV (Spearman; N = 44 model and threshold configurations).*

| Relationship | Spearman ρ |
|---|---|
| ρ(T-CPS, CPS) | 0.999 |
| ρ(T-CPS, CV) | 0.392 |

The near-perfect correlation between T-CPS and mean CPS (ρ ≈ 0.999) reflects their shared derivation from the same per-question outputs. T-CPS is not intended to replace CPS as an absolute quality indicator; rather, it serves as a decision-support score for threshold selection by highlighting configurations that achieve comparable mean performance while exhibiting different sensitivity to similarity threshold changes. The moderate positive correlation with CV (ρ ≈ 0.392) confirms that T-CPS introduces a secondary preference for stable configurations without overriding mean performance.

The utility of stability-aware scoring is illustrated by DeepSeek R1 8B: the top CPS improvement occurred at threshold 0.90, whereas the top T-CPS configuration occurred at threshold 0.65 due to lower variability. T-CPS thus discourages selecting configurations that appear strong on average but behave less consistently across threshold variations.

Full numeric correlation matrix values and computation notes are provided in Appendix B.

### 4.3.6 Balance Score

Table 4.10 ranks configurations by Balance Score, which quantifies improvement magnitude relative to output variability (Balance Score = (T-CPS improvement % / 100) / CV, as defined in Section 3.5.4). The ranking is restricted to configurations with statistically significant positive improvements.

***Table 4.10*** *Balance Score ranking (top 10 statistically significant positive configurations).*

| Rank | Model | Threshold | T-CPS Impr. % | CV | Balance Score | Sig. |
|---|---|---|---|---|---|---|
| 1 | Mistral 7B v0.3 | 0.95 | +4.54 | 0.134 | 0.339 | *** |
| 2 | Mistral 7B v0.3 | 0.9 | +2.36 | 0.131 | 0.18 | ** |
| 3 | Mistral 7B v0.3 | 0.7 | +2.17 | 0.128 | 0.17 | * |
| 4 | Mistral 7B v0.3 | 0.75 | +2.06 | 0.128 | 0.161 | * |
| 5 | Mistral 7B v0.3 | 0.6 | +1.92 | 0.135 | 0.142 | * |
| 6 | Mistral 7B v0.3 | 0.55 | +1.77 | 0.132 | 0.134 | * |
| 7 | Granite 3.2 8B | 0.95 | +1.25 | 0.124 | 0.101 | ** |
| 8 | Llama 3.1 8B | 0.9 | +1.48 | 0.148 | 0.1 | * |
| 9 | Granite 3.2 8B | 0.8 | +1.2 | 0.122 | 0.098 | ** |
| 10 | DeepSeek R1 8B | 0.65 | +0.79 | 0.085 | 0.093 | ** |

Mistral 7B v0.3 dominates the Balance Score ranking, occupying the top 6 positions. This reflects the model's combination of substantial T-CPS improvements with moderate CV values. DeepSeek R1 8B achieves rank 10 despite having the smallest T-CPS improvement (+0.79%) due to its notably low CV (0.085).

Threshold alignment across the three selection criteria is summarized in Table 4.11. Mistral 7B v0.3, Granite 3.2 8B, and Llama 3.1 8B exhibit perfect alignment, with all three metrics identifying the same best similarity threshold. DeepSeek R1 8B shows divergent optima: CPS favors threshold 0.90, while T-CPS and Balance Score favor threshold 0.65.

***Table 4.11*** *Threshold Alignment Across Selection Criteria by Model.*

| Model | Best CPS Threshold | Best T-CPS Threshold | Best Balance Score Threshold | Alignment |
|---|---|---|---|---|
| Mistral 7B v0.3 | 0.95 | 0.95 | 0.95 | Perfect |
| Granite 3.2 8B | 0.95 | 0.95 | 0.95 | Perfect |
| Llama 3.1 8B | 0.9 | 0.9 | 0.9 | Perfect |
| DeepSeek R1 8B | 0.9 | 0.65 | 0.65 | Divergent |

Figure 4.4 examines the correspondence between CPS and T-CPS by plotting paired improvements per threshold against the y = x diagonal. Points near the diagonal indicate agreement between raw and stability-adjusted scores; deviations highlight thresholds where variability penalization meaningfully shifts the evaluation.

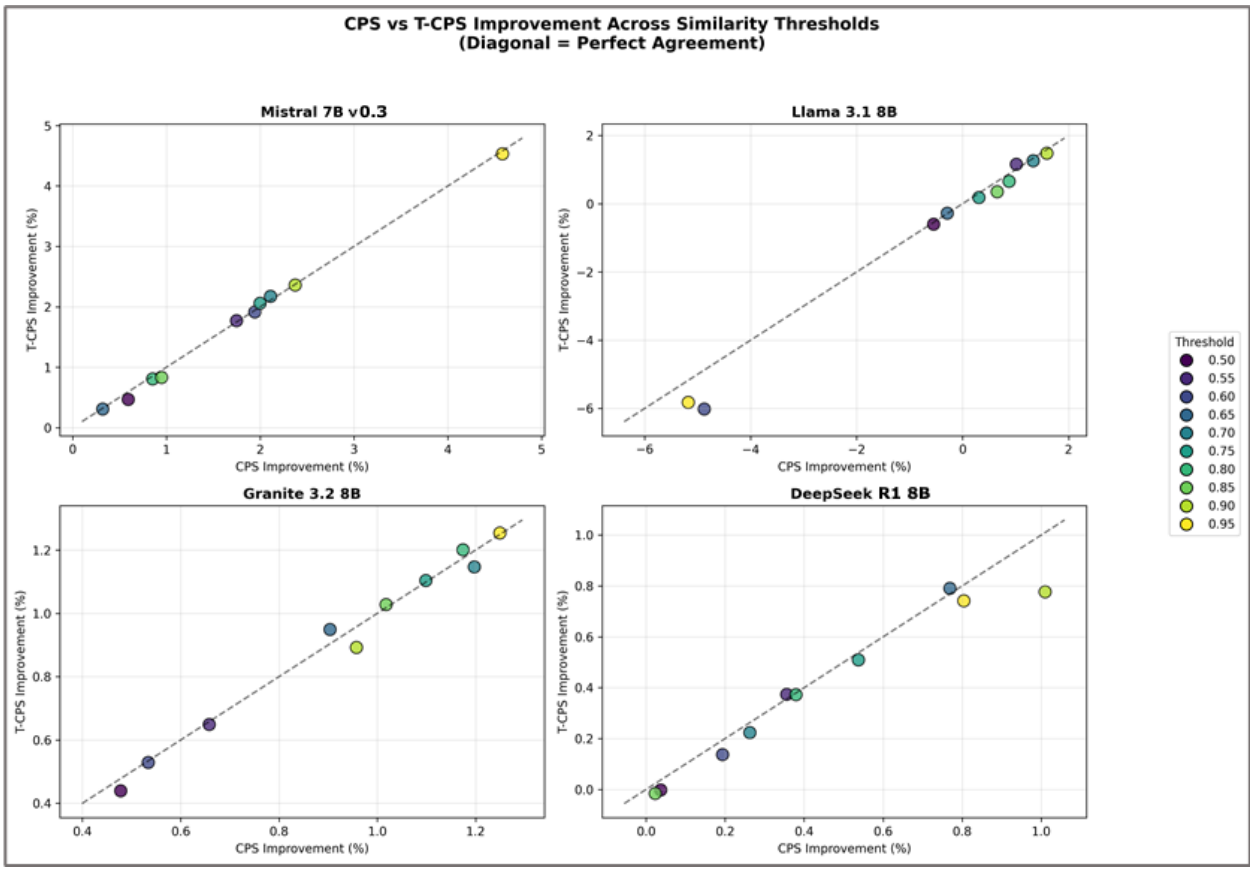


***Figure 4.4*** *Phase III (Agriculture, N = 369): CPS–T-CPS Agreement Across Thresholds per Model (Diagonal = Perfect Agreement).*

### 4.3.7 Statistical Significance

Statistical significance was assessed using two-tailed paired t-tests comparing per-question CPS values at each similarity threshold against the baseline. Reported p-values are uncorrected and should be interpreted as descriptive evidence of sensitivity across the similarity threshold sweep, with emphasis on threshold patterns within each model rather than any single comparison. Table 4.12 summarizes model-level significance coverage.

***Table 4.12*** *Significance Distribution by Model (Agriculture Domain).*

| Model | Significant Positive | Significant Negative | Not Significant |
|---|---|---|---|
| Mistral 7B v0.3 | 6 | 0 | 4 |
| Granite 3.2 8B | 7 | 0 | 3 |
| Llama 3.1 8B | 1 | 2 | 7 |
| DeepSeek R1 8B | 2 | 0 | 8 |

Per-threshold p-values are visualized in Figure 4.5, underlying the summary in Table 4.12. The statistical evaluatio framework is defined in Section 3.5.3. Full statistical tables, effect sizes (Cohen's d), and 95% confidence intervals are reported in Appendix B.

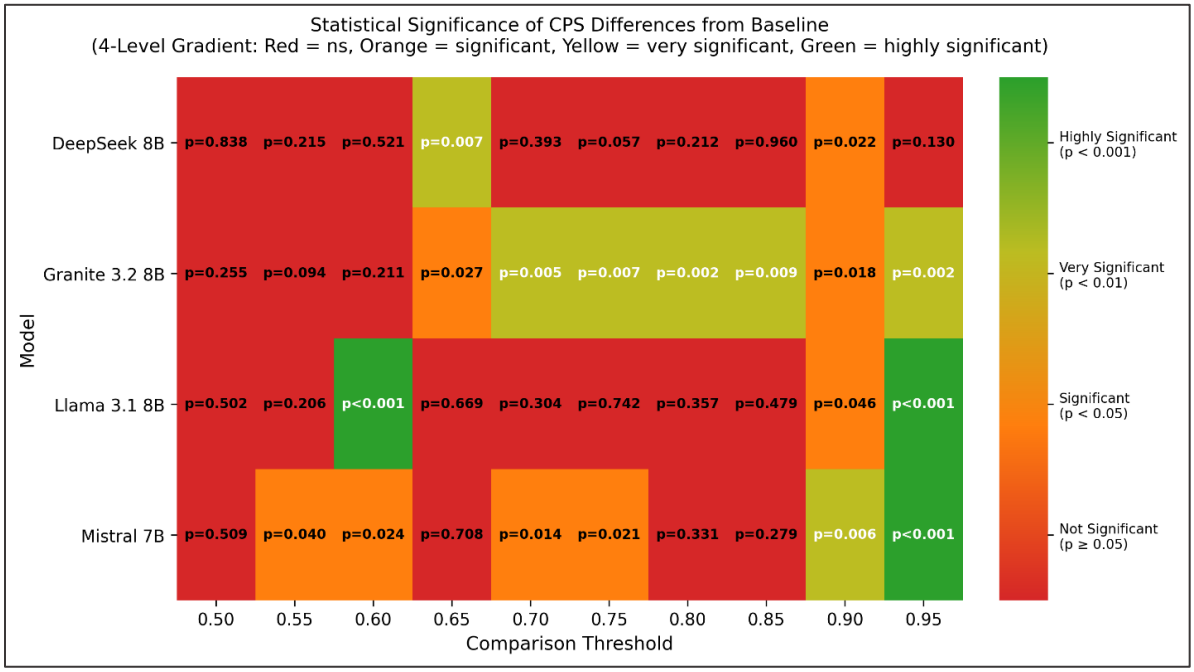


***Figure 4.5 Statistical significance heatmap of CPS differences from baseline across similarity thresholds (Phase III).*** *P-values are from two-tailed paired t-tests against baseline at the per-question level (uncorrected); values below 0.001 are shown as p<0.001. The four-level gradient encodes significance as not significant (p ≥ 0.05), significant (p < 0.05), very significant (p < 0.01), and highly significant (p < 0.001).*

Mistral 7B v0.3 reaches significance at 6 of 10 thresholds (60%), with the strongest evidence at threshold 0.95 ($p < 0.001$). Significant thresholds cluster at moderate to high selectivity (0.55, 0.60, 0.70, 0.75, 0.90, 0.95), indicating benefit from filtered retrieval across a

broad range. Granite 3.2 8B exhibits the most comprehensive pattern, with consecutive significant improvements from threshold 0.65 through 0.95. Llama 3.1 8B is more selective, showing one significant improvement (0.90) and significant decreases at 0.60 and 0.95 ($p < 0.001$). DeepSeek R1 8B is the most constrained, with improvements limited to 0.65 ($p < 0.01$) and 0.90 ($p < 0.05$).

### 4.3.8 Model-Specific Similarity Threshold Sensitivity Patterns

Mistral 7B v0.3 exhibits the most favorable sensitivity profile, with six statistically significant thresholds spanning 0.55–0.95. T-CPS improvements remain above 1.7% across this range, with peak performance at threshold 0.95 (+4.54% T-CPS improvement). Balance Score results also identify threshold 0.95 as the best configuration (Balance Score = 0.339). Overall, the effective similarity threshold range is 0.55–0.95, with a peak zone at 0.90–0.95, indicating low sensitivity and high tolerance.

Granite 3.2 8B demonstrates consistent behavior with seven consecutive significant thresholds from 0.65 to 0.95. T-CPS improvements remain within 0.89%–1.25% across this interval, and CV values cluster in a narrow band (0.120–0.130), indicating stable output quality. Balance Score identifies threshold 0.95 as the best configuration (Balance Score = 0.101). The effective similarity threshold range is 0.65–0.95, reflecting very low sensitivity and stable performance.

Llama 3.1 8B shows a narrow sensitivity pattern, with a single statistically significant positive improvement at threshold 0.90 (+1.58% CPS, +1.48% T-CPS). Performance degrades at thresholds 0.60 (−4.88% CPS, $p < 0.001$) and 0.95 (−5.18% CPS, $p < 0.001$), highlighting the importance of similarity threshold selection for this model. Balance Score identifies threshold 0.90 as the best configuration (Balance Score = 0.100). Accordingly, the effective similarity threshold range is limited to 0.90, indicating high sensitivity and a narrow effective range.

DeepSeek R1 8B exhibits the most constrained sensitivity profile, with statistically significant improvements at two thresholds (0.65 and 0.90). The best CPS performance occurs at threshold 0.90 (+1.01%), whereas the best T-CPS performance occurs at threshold 0.65 (+0.79%), indicating divergence between mean performance and stability-aware scoring. Balance Score identifies threshold 0.65 as the best configuration for stability (Balance Score = 0.093). The

model's low CV values (0.085–0.108) indicate consistent output quality despite limited improvement magnitude, reflecting very high sensitivity and a limited response range.

### 4.3.9 Phase III Summary

Similarity threshold sensitivity was evaluated across four open-source language models using 369 question-answer pairs from the agricultural domain. The CPS and T-CPS framework, combined with Balance Score analysis, enabled comparison of both absolute performance and output stability.

Based on alignment across CPS, T-CPS, and Balance Score, the best-performing agricultural similarity thresholds (summarized in Table 4.13) are:

- **Mistral 7B v0.3**: similarity threshold 0.95 (CPS +4.58%, T-CPS +4.54%, CV 0.134, Balance Score 0.339).
- **Granite 3.2 8B**: similarity threshold 0.95 (CPS +1.25%, T-CPS +1.25%, CV 0.124, Balance Score 0.101).
- **Llama 3.1 8B**: similarity threshold 0.90 (CPS +1.58%, T-CPS +1.48%, CV 0.148, Balance Score 0.100).
- **DeepSeek R1 8B**: similarity threshold 0.65 for stability-aware selection (CPS +0.77%, T-CPS +0.79%, CV 0.085, Balance Score 0.093) or similarity threshold 0.90 for mean-performance selection (CPS +1.01%, T-CPS +0.78%, CV 0.108, Balance Score 0.072).

***Table 4.13*** *Best-Performing Configurations Summary (Agriculture Domain). Significance markers summarize paired t-test results; full statistical tables are reported in Appendix B.*

| Model | Peak Threshold | CPS Impr. % | T-CPS Impr. % | CV | Balance Score | Sig. |
|---|---|---|---|---|---|---|
| Mistral 7B v0.3 | 0.95 | 4.58 | 4.54 | 0.134 | 0.339 | *** |
| Granite 3.2 8B | 0.95 | 1.25 | 1.25 | 0.124 | 0.101 | ** |
| Llama 3.1 8B | 0.9 | 1.58 | 1.48 | 0.148 | 0.1 | * |
| DeepSeek R1 8B | 0.65 / 0.90 | 0.77 / 1.01 | 0.79 / 0.78 | 0.085 / 0.108 | 0.093 / 0.072 | ** / * |

** DeepSeek R1 8B shows divergent optima: similarity threshold 0.65 prioritizes stability-aware selection (T-CPS and Balance Score), while similarity threshold 0.90 prioritizes mean performance (CPS).*

Peak-performing similarity thresholds varied across models: Mistral 7B v0.3 and Granite 3.2 8B peaked at 0.95, Llama 3.1 8B at 0.90, and DeepSeek R1 8B exhibited divergent CPS and T-CPS optima (0.90 vs. 0.65). Balance Score analysis identified stability-performance trade-offs, with

Mistral 7B v0.3 leading the ranking due to its combination of high improvement magnitude and moderate variability; DeepSeek R1 8B achieved competitive Balance Score despite modest improvements due to notably low CV.

Three models (Mistral 7B v0.3, Granite 3.2 8B, Llama 3.1 8B) exhibited perfect threshold alignment across CPS, T-CPS, and Balance Score, whereas DeepSeek R1 8B required trade-off decisions between mean performance and stability. Distinct sensitivity profiles were observed: Granite 3.2 8B showed plateau behavior (7/10 thresholds significant), while DeepSeek R1 8B demonstrated minimal response (2/10 significant). Model-specific threshold selection produced CPS improvements ranging from 0.77% to 4.58% over baseline configurations without requiring architectural modifications.

These results are bounded by the tested configuration: four models, 369 questions, agricultural domain, and heterogeneous hardware environments. Hardware heterogeneity may introduce minor confounds; all statistical comparisons used within-model baselines to mitigate this concern. Cross-domain generalization is examined in Phase IV. These results provide evidence for RQ1 (similarity threshold effects on generation quality) and RQ2 (model-dependent threshold sensitivity) within the agricultural domain.

## 4.4 Phase IV: Cross-Domain Evaluation (Biodiversity)

Phase IV repeats the Phase III similarity threshold sweep under the same experimental procedure to assess cross-domain generalization. The changes from Phase III are limited to the knowledge corpus, the test set, and the execution environment: Phase IV uses a biodiversity corpus and N = 426 question–answer pairs generated using Claude Opus, with all runs executed on a single hardware configuration (M1 Mac Mini). All other settings remain identical to Phase III, including the model set (Mistral 7B v0.3, DeepSeek R1 8B, Llama 3.1 8B, Granite 3.2 8B), similarity threshold range (0.50–0.95, step 0.05), baseline (Normal Mode) and thresholded (Score Mode) run definitions, chunking parameters, embedding model, prompt format, and temperature (0.2). The full 24-metric set defined in Section 3.4 was computed for each run. Detailed statistical outputs are provided in Appendix C.

The primary purpose is to characterize how threshold effects vary when corpus properties differ, including vocabulary, embedding similarity distributions, and the concentration of relevant

evidence per query. A threshold that performs well on one dataset may degrade performance on another if the corpus changes the density of high-similarity passages or introduces distractor text that superficially resembles relevant content. By holding retrieval mechanics constant while changing the domain, direct comparison of model sensitivity profiles across datasets is enabled. The results provide empirical grounding for determining whether best-performing thresholds require domain-specific calibration.

Observed differences across thresholds and domains are interpreted as empirical patterns rather than isolated causal effects. Multiple factors vary simultaneously during generation and retrieval, including document distributions, chunk content, retrieval selectivity, and model behavior. Explanations are therefore presented as plausible hypotheses consistent with the observed results rather than definitive mechanisms.

Phase IV addresses RQ1 (similarity threshold effects on generation quality), RQ2 (model-dependent threshold sensitivity), and RQ3 (whether comparable similarity threshold ranges hold across knowledge domains), applying the evaluation procedure (a) and experimental design (c) components of Objective 1 and contributing to Objective 4 (controlled testing and analysis). Seven tasks are conducted: tasks (1) - (6) replicate the Phase III technique in the biodiversity domain, and task (7) compares threshold sensitivity patterns across domains to assess cross-domain generalization.

### 4.4.1 Experimental Design

Experiments were conducted under a single fixed configuration: M1 Mac Mini with 16,000-token context buffer for all models and thresholds. This standardization eliminates context buffer variation as a confounding factor, enabling direct attribution of observed differences to domain effects.

The biodiversity corpus comprises 426 question–answer pairs constructed from authoritative sources including the Convention on Biological Diversity [144] and the EU Biodiversity Strategy [145]. Reference answers were extracted directly from source documents; questions were generated using Claude Opus, adapting the procedure used for the agricultural dataset in Phase I. In total, 18,744 evaluations were conducted (4 models × 11 threshold configurations × 426 question–answer pairs).

### 4.4.2 CPS Weighting Scheme

The 9-metric weighting scheme from Phase III (Table 4.6) was applied without modification. The weights reflect the four-construct evaluation framework: lexical overlap (30%), semantic similarity (25%), fluency and accuracy (25%), and language modeling (20%). The T-CPS formulation (Section 3.5.2) was applied with parameters $\alpha = 0.1$ and $\beta = 0.05$.

### 4.4.3 CPS Performance Overview

The effects of varying the similarity threshold from 0.50 to 0.95 on CPS-based performance were examined across four open-source models in the biodiversity domain, using 426 question–answer pairs (18,744 total evaluations). Results are organized to summarize overall CPS improvements before examining stability (T-CPS), metric relationships, and statistical significance.

Table 4.14 presents the top CPS improvement configurations for each model. Mistral 7B v0.3 achieved the largest improvement (+13.32% at threshold 0.80), followed by DeepSeek R1 8B (+8.45% at threshold 0.55), Granite 3.2 8B (+6.95% at threshold 0.80), and Llama 3.1 8B (+2.06% at threshold 0.85).

***Table 4.14*** *Top CPS Improvement Configurations by Model (Top 3 Biodiversity)*

| Model | Rank | Threshold | Mean CPS | Improvement % |
|---|---|---|---|---|
| Mistral 7B v0.3 | 1 | 0.8 | 0.4911 | 13.32 |
| | 2 | 0.65 | 0.4764 | 9.94 |
| | 3 | 0.7 | 0.4747 | 9.53 |
| Granite 3.2 8B | 1 | 0.8 | 0.4473 | 6.95 |
| | 2 | 0.95 | 0.4422 | 5.73 |
| | 3 | 0.55 | 0.4413 | 5.51 |
| Llama 3.1 8B | 1 | 0.85 | 0.4713 | 2.06 |
| | 2 | 0.7 | 0.4606 | −0.25 |
| | 3 | 0.8 | 0.4567 | −1.11 |
| DeepSeek R1 8B | 1 | 0.55 | 0.5094 | 8.45 |
| | 2 | 0.6 | 0.4906 | 4.45 |
| | 3 | 0.7 | 0.4775 | 1.66 |

Section 4.4.4 evaluates whether these peak CPS configurations remain preferred when stability is incorporated through T-CPS. Full statistical tables and per-threshold descriptive metrics are provided in Appendix C

### 4.4.4 T-CPS Performance and Stability

Stability-aware results are summarized in Table 4.15, which reports the top T-CPS improvement configurations by model and incorporates stability through the coefficient of variation (CV). The top T-CPS configuration is model-dependent: Mistral 7B v0.3 peaks at threshold 0.80 (T-CPS = 0.5254; +14.23%), Granite 3.2 8B peaks at threshold 0.80 (T-CPS = 0.4785; +7.25%), Llama 3.1 8B peaks at threshold 0.85 (T-CPS = 0.5042; +2.29%), and DeepSeek R1 8B peaks at threshold 0.55 (T-CPS = 0.5529; +8.75%). The T-CPS rankings largely mirror CPS rankings, while CV values highlight differences in output consistency across models.

Across the top configurations, DeepSeek R1 8B exhibits lower variability (CV = 0.129–0.158) than the other models (CV = 0.233–0.254), indicating more stable output quality across queries. Parameter sensitivity and ranking stability for T-CPS are reported in Section 3.5.2.

***Table 4.15*** *Top T-CPS Improvement Configurations by Model (Top 3 Biodiversity)*

| Model | Rank | Threshold | T-CPS | T-CPS Impr. % | CV | Interpretation |
|---|---|---|---|---|---|---|
| Mistral 7B v0.3 | 1 | 0.8 | 0.5254 | 14.23 | 0.242 | Large improvement |
| | 2 | 0.65 | 0.5102 | 10.93 | 0.233 | Large improvement |
| | 3 | 0.7 | 0.5078 | 10.41 | 0.24 | Large improvement |
| Granite 3.2 8B | 1 | 0.8 | 0.4785 | 7.25 | 0.239 | Moderate improvement |
| | 2 | 0.55 | 0.4723 | 5.87 | 0.235 | Moderate improvement |
| | 3 | 0.95 | 0.472 | 5.8 | 0.254 | Moderate improvement |
| Llama 3.1 8B | 1 | 0.85 | 0.5042 | 2.29 | 0.24 | Small improvement |
| | 2 | 0.7 | 0.4928 | −0.03 | 0.24 | Minimal change |
| | 3 | 0.8 | 0.4884 | −0.92 | 0.241 | Minimal change |
| DeepSeek R1 8B | 1 | 0.55 | 0.5529 | 8.75 | 0.129 | Large improvement |
| | 2 | 0.6 | 0.5306 | 4.38 | 0.158 | Moderate improvement |
| | 3 | 0.7 | 0.5165 | 1.59 | 0.157 | Small improvement |

Full per-threshold T-CPS descriptive metrics for each model are provided in Appendix C.

Figure 4.6 compares threshold-wise CPS and T-CPS improvements for each model across the similarity threshold sweep.

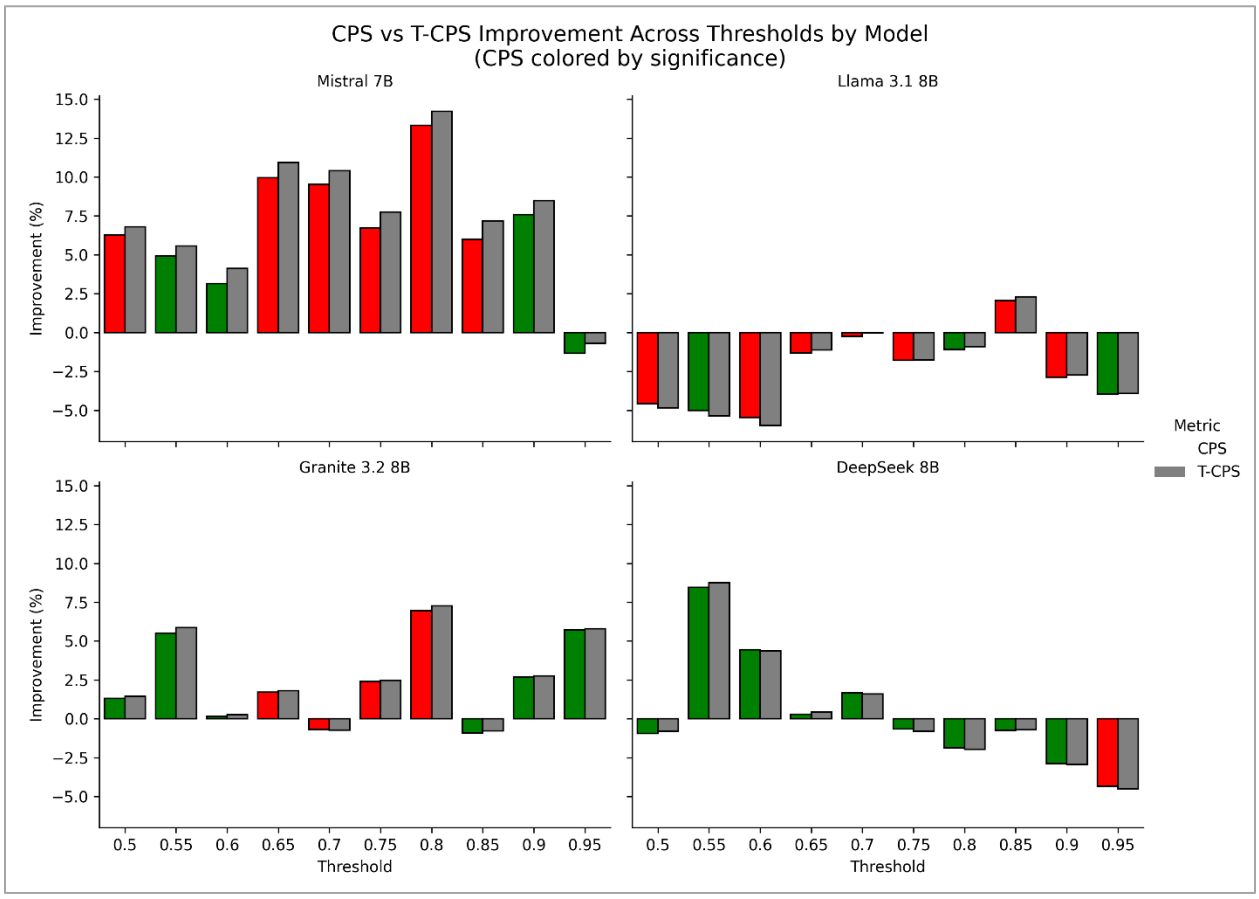


***Figure 4.6*** *Phase IV (Biodiversity, N = 426): Threshold-wise CPS and T-CPS improvements per model. CPS bars indicate percent improvement relative to baseline and are colored by significance at p < 0.05 (uncorrected). T-CPS bars (gray) indicate stability-aware improvement.*

### 4.4.5 Correlation Analysis

To assess redundancy among component metrics and to examine whether T-CPS provides information beyond mean CPS, correlation analysis was performed. Figure 4.7 presents the Spearman correlation matrix among Phase IV component metrics, computed from pooled per-question results across thresholds 0.50–0.95 plus baseline. Table 4.16 reports Spearman correlations between T-CPS and (i) mean CPS and (ii) the coefficient of variation (CV), computed across all model and threshold configurations.

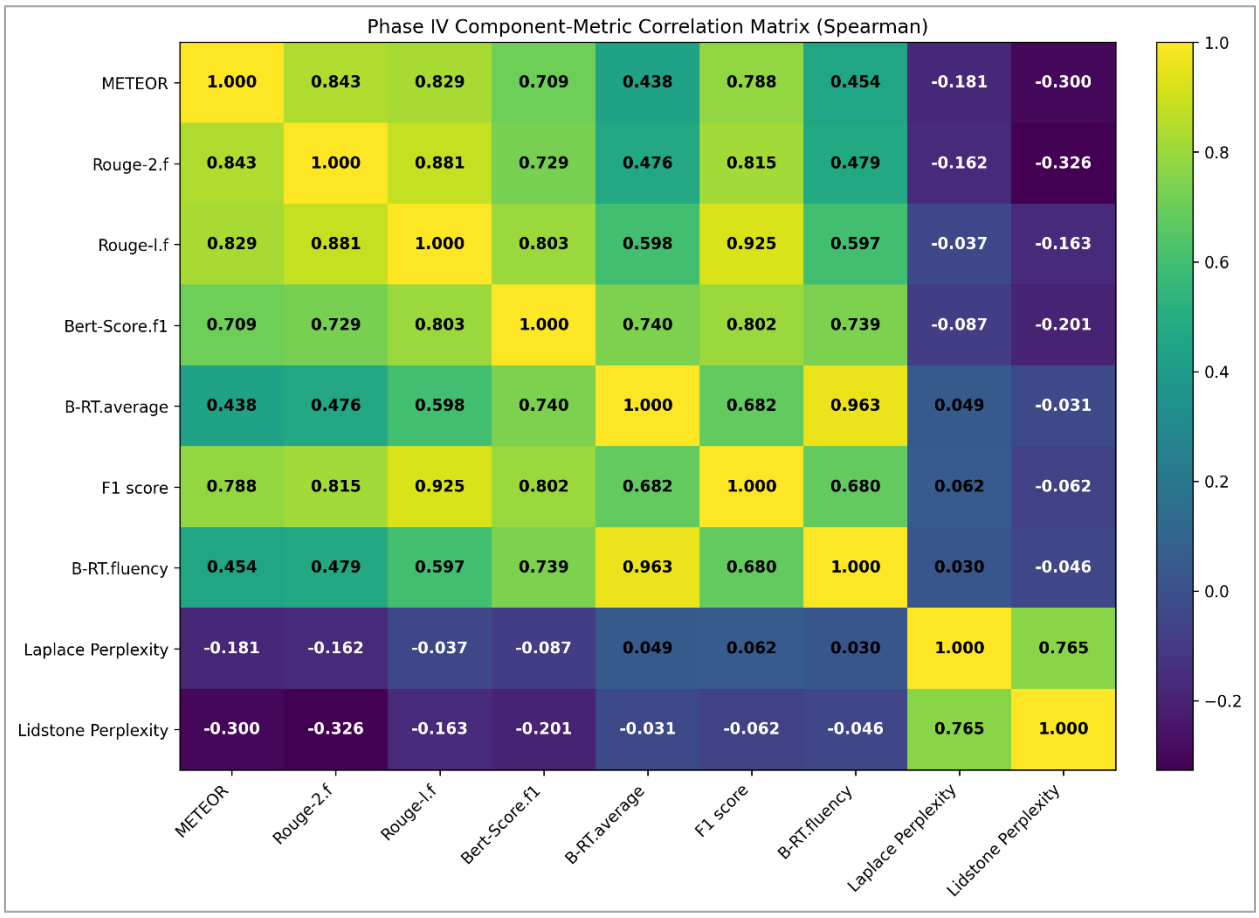


***Figure 4.7 Phase IV component metric correlation matrix (Spearman), pooled across models and similarity thresholds 0.50–0.95 plus baseline using per-question results.***

Strong clustering is observed among overlap-based metrics: ROUGE-2.f and ROUGE-L.f (ρ ≈ 0.881), METEOR and ROUGE-2.f (ρ ≈ 0.843), and METEOR and ROUGE-L.f (ρ ≈ 0.829). The B-RT metrics remain tightly coupled (B-RT.average with B-RT.fluency: ρ ≈ 0.963). Laplace and Lidstone perplexity are strongly correlated (ρ ≈ 0.765) and show negative associations with all other metrics, indicating that lower perplexity aligns with higher overlap and semantic similarity scores in the biodiversity domain.

Compared with Phase III, the lexical metrics form a tighter cluster in Phase IV: METEOR–ROUGE correlations increased from ρ ≈ 0.01–0.23 (agriculture) to ρ ≈ 0.82–0.84 (biodiversity). The perplexity–quality relationship also reversed: Phase III exhibited positive correlations between perplexity and quality metrics (ρ ≈ 0.2–0.5), whereas Phase IV shows negative correlations (ρ ≈ −0.3 to −0.06). These domain-dependent patterns suggest that metric interdependencies are corpus-sensitive.

*Table 4.16 Phase IV associations between stability-aware scoring T-CPS, mean CPS, and variability CV (Spearman; N = 44 model and threshold configurations).*

| Relationship | Spearman ρ |
|---|---|
| ρ(T-CPS, CPS) | 0.992 |
| ρ(T-CPS, CV) | -0.724 |

T-CPS remains strongly aligned with CPS (ρ ≈ 0.992), indicating that stability-aware scoring largely preserves mean-performance ordering. The strong negative correlation with CV (ρ ≈ −0.724) indicates that variability meaningfully influences T-CPS ranking and can alter configuration ordering when mean CPS values are close. Compared with Phase III (ρ(T-CPS, CPS) ≈ 0.999; ρ(T-CPS, CV) ≈ 0.392), Phase IV exhibits a stronger stability signal, suggesting that the biodiversity domain introduces a more pronounced performance–variability trade-off.

The effect of stability-aware scoring in Phase IV is illustrated through two observations.

First, comparing DeepSeek R1 8B across domains: in Phase III, the moderate CV correlation (ρ ≈ 0.392) allowed variability to shift highest-scoring threshold from 0.90 (CPS) to 0.65 (T-CPS); in Phase IV, despite the stronger negative CV correlation (ρ ≈ −0.724), DeepSeek's consistently low variability (CV ≈ 0.129–0.158) meant CPS and T-CPS agreed on threshold 0.55. This convergence reflects DeepSeek's stable behavior in the biodiversity domain rather than an absence of the stability penalty.

Second, even when CPS and T-CPS agree on the peak threshold, improvement percentages differ: for Llama 3.1 8B, CPS improvement at threshold 0.85 was +2.06%, whereas T-CPS improvement was +2.29%. This difference reflects the variability adjustment applied by T-CPS, which rewards configurations with more consistent performance.

Full numeric correlation matrix values and computation notes are provided in Appendix C.

### 4.4.6 Balance Score

Table 4.17 ranks configurations by Balance Score, which quantifies improvement magnitude relative to output variability (Balance Score = (T-CPS improvement % / 100) / CV, as

defined in Section 3.5.4). The ranking is restricted to configurations with statistically significant positive improvements.

***Table 4.17*** *Balance Score Ranking (top 10 statistically significant positive configurations)*

| Rank | Model | Threshold | T-CPS Impr. % | CV | Balance Score | Sig. |
|---|---|---|---|---|---|---|
| 1 | DeepSeek R1 8B | 0.55 | +8.75 | 0.129 | 0.678 | *** |
| 2 | Mistral 7B v0.3 | 0.8 | +14.23 | 0.242 | 0.588 | *** |
| 3 | Mistral 7B v0.3 | 0.65 | +10.93 | 0.233 | 0.469 | *** |
| 4 | Mistral 7B v0.3 | 0.7 | +10.41 | 0.24 | 0.434 | *** |
| 5 | Mistral 7B v0.3 | 0.9 | +8.49 | 0.236 | 0.36 | *** |
| 6 | Mistral 7B v0.3 | 0.75 | +7.74 | 0.228 | 0.34 | *** |
| 7 | Mistral 7B v0.3 | 0.85 | +7.17 | 0.217 | 0.33 | *** |
| 8 | Granite 3.2 8B | 0.8 | +7.25 | 0.239 | 0.303 | *** |
| 9 | DeepSeek R1 8B | 0.6 | +4.38 | 0.158 | 0.277 | *** |
| 10 | Mistral 7B v0.3 | 0.5 | +6.8 | 0.26 | 0.262 | *** |

DeepSeek R1 8B at threshold 0.55 achieves the highest Balance Score (0.678) due to its combination of substantial T-CPS improvement (+8.75%) and notably low CV (0.129). This configuration outranks Mistral 7B v0.3's larger improvement (+14.23%) because the latter's higher CV (0.242) reduces the stability-adjusted ratio. Mistral 7B v0.3 dominates ranks 2–7 and rank 10, reflecting consistent high improvements across multiple thresholds.

Table 4.18 examines threshold alignment across the three selection criteria. All four models exhibit perfect alignment: the threshold maximizing CPS also maximizes T-CPS and Balance Score. This alignment simplifies threshold selection by eliminating trade-off decisions between mean performance and stability—a contrast with Phase III, where DeepSeek R1 8B showed divergent optima.

***Table 4.18*** *Threshold Alignment Across Selection Criteria by Model*

| Model | Best CPS Threshold | Best T-CPS Threshold | Best Balance Score Threshold | Alignment |
|---|---|---|---|---|
| Mistral 7B v0.3 | 0.8 | 0.8 | 0.8 | Perfect |
| Granite 3.2 8B | 0.8 | 0.8 | 0.8 | Perfect |
| Llama 3.1 8B | 0.85 | 0.85 | 0.85 | Perfect |
| DeepSeek R1 8B | 0.55 | 0.55 | 0.55 | Perfect |

Figure 4.8 examines the correspondence between CPS and T-CPS by plotting paired improvements per threshold against the y = x diagonal. Points near the diagonal indicate agreement between raw and stability-adjusted scores; deviations highlight thresholds where variability penalization meaningfully shifts the evaluation.

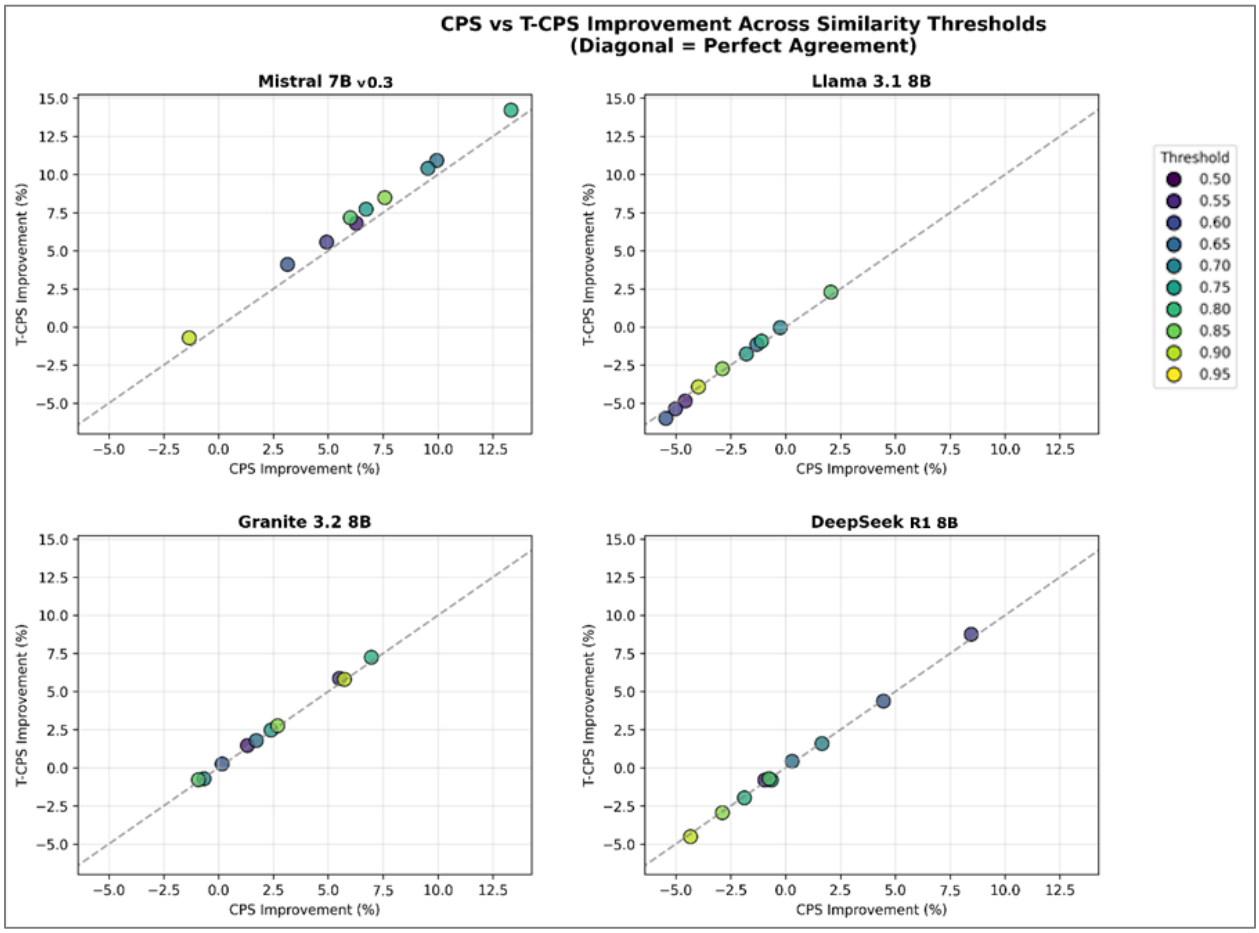


***Figure 4.8*** *Phase IV (Biodiversity, N = 426): CPS–T-CPS Agreement Across Thresholds per Model (Diagonal = Perfect Agreement).*

## 4.4.7 Statistical Significance

Statistical significance was assessed using two-tailed paired t-tests comparing per-question CPS values at each similarity threshold against the baseline. Reported p-values are uncorrected and should be interpreted as descriptive evidence of sensitivity across the similarity threshold sweep, with emphasis on threshold patterns within each model rather than any single comparison. Table 4.19 summarizes model-level significance coverage.

*Table 4.19 Significance Distribution by Model (Biodiversity Domain)*

| Model | Significant Positive | Significant Negative | Not Significant |
|---|---|---|---|
| DeepSeek R1 8B | 2 | 2 | 6 |
| Granite 3.2 8B | 3 | 0 | 7 |
| Llama 3.1 8B | 1 | 5 | 4 |
| Mistral 7B v0.3 | 9 | 0 | 1 |

Per-threshold p-values are visualized in Figure 4.9, underlying the summary in Table 4.15. The statistical evalulation framework is defined in Section 3.5.3. Full statistical tables, effect sizes (Cohen's d), and 95% confidence intervals are reported in Appendix C.

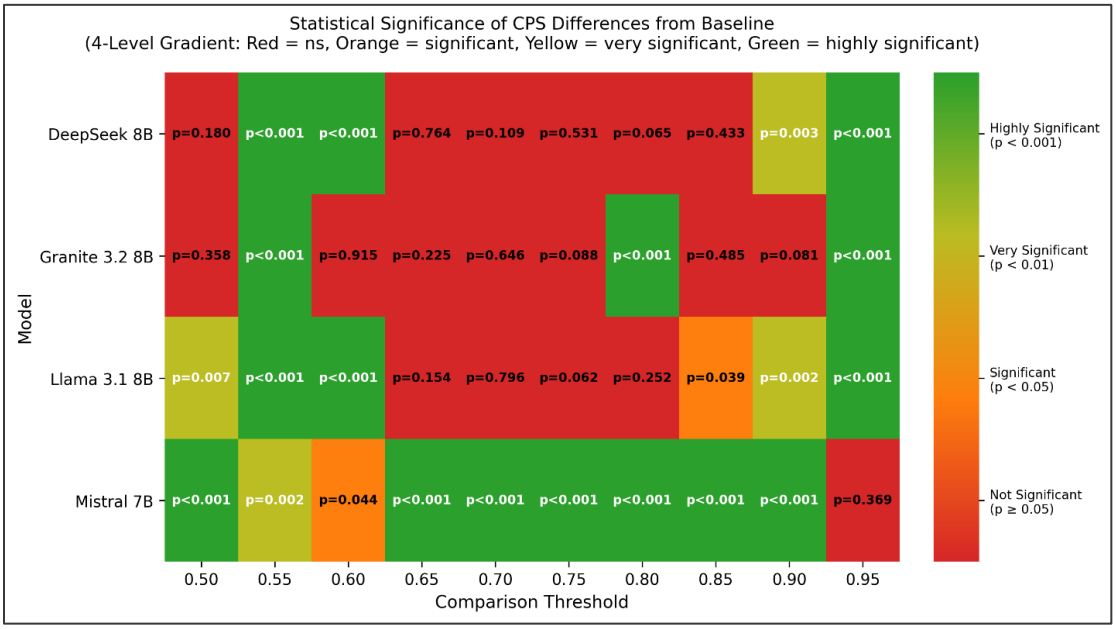


***Figure 4.9 Statistical significance heatmap of CPS differences from baseline across similarity thresholds (Phase IV).*** *P-values are from two-tailed paired t-tests against baseline at the per-question level (uncorrected); values below 0.001 are shown as p<0.001. The four-level gradient encodes significance as not significant (p ≥ 0.05), significant (p < 0.05), very significant (p < 0.01), and highly significant (p < 0.001).*

Mistral 7B v0.3 shows significance at 9 of 10 thresholds (90%), with the strongest evidence concentrated in the 0.65–0.90 range (all p < 0.001). The model's broad significance profile indicates robust threshold tolerance, with only threshold 0.95 failing to reach significance.

Granite 3.2 8B achieves significance at 3 of 10 thresholds (30%), with significant positive results at thresholds 0.55, 0.80, and 0.95 (all p < 0.001). The non-consecutive pattern suggests sensitivity to specific threshold regions rather than a continuous response zone.

Llama 3.1 8B shows a selective significance pattern, with only 1 threshold (0.85, $p < 0.05$) producing statistically significant improvement over baseline, while thresholds 0.50, 0.55, 0.60, 0.90, and 0.95 produced significant decreases in performance.

DeepSeek R1 8B has a constrained significance profile, with meaningful improvements limited to thresholds 0.55 ($p < 0.001$) and 0.60 ($p < 0.001$), and significant negative effects at thresholds 0.90 and 0.95.

### 4.4.8 Model-Specific Similarity Threshold Sensitivity Patterns

Mistral 7B v0.3 exhibits the most favorable sensitivity profile in the biodiversity domain, achieving statistically significant positive improvements at 9 of 10 tested thresholds. Peak performance occurs at similarity threshold 0.80 (+13.32% CPS, +14.23% T-CPS). Balance Score results also identify similarity threshold 0.80 as the most favorable configuration (Balance Score = 0.588). Overall, the effective similarity threshold range is 0.50–0.90, with a peak zone at 0.65–0.85, indicating low sensitivity and high tolerance.

Granite 3.2 8B demonstrates moderate sensitivity in the biodiversity domain, with statistically significant positive improvements at thresholds 0.55, 0.80, and 0.95. Peak performance occurs at similarity threshold 0.80 (+6.95% CPS, +7.25% T-CPS). CV values cluster in a narrow band (0.235–0.258), indicating stable output quality across thresholds. Balance Score identifies similarity threshold 0.80 as the most favorable configuration (Balance Score = 0.303). Accordingly, the effective similarity threshold range is 0.55, 0.80, and 0.95 (non-consecutive), reflecting moderate sensitivity with stable performance.

Llama 3.1 8B shows a constrained sensitivity pattern in the biodiversity domain, with only one statistically significant positive improvement at similarity threshold 0.85 (+2.06% CPS, +2.29% T-CPS) against five significant negative conditions. Performance degradation is observed at several permissive and strict thresholds, with similarity threshold 0.60 yielding a −5.46% CPS change. Balance Score identifies similarity threshold 0.85 as the most favorable configuration (Balance Score = 0.095). Accordingly, the effective similarity threshold range is limited to 0.85, indicating very high sensitivity and a narrow effective range.

DeepSeek R1 8B exhibits the most constrained sensitivity profile in the biodiversity domain, with statistically significant improvements limited to similarity thresholds 0.55 and 0.60. The best configuration occurs at similarity threshold 0.55 (+8.45% CPS, +8.75% T-CPS), which also

achieves the highest Balance Score (Balance Score = 0.678) due to low CV (0.129). In contrast, strict thresholds produce significant negative effects, with similarity threshold 0.95 yielding a −4.51% T-CPS change. Balance Score results identify similarity threshold 0.55 as the most favorable configuration for stability-adjusted performance. Overall, the effective similarity threshold range is 0.55–0.60, indicating high sensitivity and responsiveness limited to low thresholds.

### 4.4.9 Cross-Domain Comparison

Comparing Phase IV results to Phase III reveals both persistent patterns and domain-induced shifts. Table 4.20 summarizes the comparison.

*Table 4.20 Cross-Domain Comparison: Agriculture (Phase III) vs. Biodiversity (Phase IV).*

| Model | Agriculture Most Favorable Threshold | Agriculture CPS improvement % | Bio Most Favorable Threshold | Bio CPS improvement % | Shift |
|---|---|---|---|---|---|
| Mistral 7B v0.3 | 0.95 | +4.58% | 0.80 | +13.32% | -0.15 |
| Granite 3.2 8B | 0.95 | +1.25% | 0.80 | +6.95% | -0.15 |
| Llama 3.1 8B | 0.90 | +1.58% | 0.85 | +2.06% | -0.05 |
| DeepSeek R1 8B | 0.90 | +1.01% | 0.55 | +8.45% | -0.35 |

** Agriculture and biodiversity thresholds reflect CPS peaks. For DeepSeek R1 8B, the stability-aware optimum (T-CPS/Balance Score) in agriculture is 0.65; see Table 4.13.*

Five findings emerge from this comparison.

**First, peak-performing shift downward for all models in the biodiversity domain.** DeepSeek R1 8B shows the largest shift (0.90 to 0.55), followed by Mistral 7B v0.3 and Granite 3.2 8B (both 0.95 to 0.80) and Llama 3.1 8B (0.90 to 0.85). This pattern suggests that the biodiversity corpus benefits from more permissive filtering, possibly because relevant evidence is dispersed across passages with varied similarity scores.

**Second, improvement magnitudes increase substantially.** Mistral 7B v0.3's peak improvement in the biodiversity domain (+13.32%) nearly triples its Phase III maximum (+4.58%), while Granite 3.2 8B's peak (+6.95%) exceeds its Phase III best (+1.25%) by more than fivefold.

This indicates greater threshold sensitivity in the biodiversity domain, possibly due to higher variance in embedding similarity distributions for taxonomic content.

**Third, significance patterns reverse between domains.** In Phase III (agriculture), Granite 3.2 8B showed the most consistent significance (7/10 thresholds), while in Phase IV (biodiversity), Mistral 7B v0.3 achieves the highest rate (9/10). This reversal illustrates the domain-dependent nature of threshold sensitivity and suggests that a model's responsiveness to threshold tuning cannot be assumed to transfer across corpora.

**Fourth, threshold alignment differs between domains.** In Phase III, DeepSeek R1 8B showed divergent CPS and T-CPS optima, requiring trade-off decisions between maximizing mean performance and minimizing variability. In Phase IV, all four models exhibit perfect alignment across CPS, T-CPS, and Balance Score, simplifying threshold selection in the biodiversity domain.

**Fifth, model-specific behaviors show both persistence and inversion.** Granite 3.2 8B maintains consistent stability behavior across domains, with CV values clustering in narrow bands in both phases (0.120–0.130 in agriculture; 0.235–0.258 in biodiversity), suggesting this is a model-intrinsic property rather than domain-dependent. In contrast, Llama 3.1 8B shows a partial inversion: Phase III yielded selective improvement at threshold 0.90, while Phase IV produces significant positive results only at threshold 0.85, with permissive thresholds that were neutral in Phase III now causing significant degradation.

These shifts confirm that best-performing thresholds cannot transfer directly across domains (R3). A practitioner deploying a RAG system with an agricultural corpus and later extending it to biodiversity would find that previously peak-performing thresholds underperform or produce negative effects. The evidence supports maintaining a calibration protocol that includes threshold sweeps when new corpora are introduced.

### 4.4.10 Phase IV Summary

Similarity threshold sensitivity was evaluated across four open-source language models using 426 question-answer pairs from the biodiversity domain. The 9-metric CPS and T-CPS framework, combined with Balance Score analysis, enabled comparison of both absolute performance and output stability.

Based on CPS, T-CPS, and Balance Score alignment, the best-performing biodiversity thresholds (summarized in Table 4.21) are:

- **Mistral 7B v0.3**: similarity threshold 0.80 (CPS +13.32%, T-CPS +14.23%, CV 0.242, Balance Score 0.588).
- **Granite 3.2 8B**: similarity threshold 0.80 (CPS +6.95%, T-CPS +7.25%, CV 0.239, Balance Score 0.303).
- **Llama 3.1 8B**: similarity threshold 0.85 (CPS +2.06%, T-CPS +2.29%, CV 0.240, Balance Score 0.095).
- **DeepSeek R1 8B**: similarity threshold 0.55 (CPS +8.45%, T-CPS +8.75%, CV 0.129, Balance Score 0.678).

*Table 4.21 Best-Performing Configurations Summary (Biodiversity Domain). Significance markers summarize paired t-test results; full statistical tables are reported in Appendix C.*

| Model | Peak Threshold | CPS Impr. % | T-CPS Impr. % | CV | Balance Score | Sig. |
|---|---|---|---|---|---|---|
| Mistral 7B v0.3 | 0.8 | 13.32 | 14.23 | 0.242 | 0.588 | *** |
| Granite 3.2 8B | 0.8 | 6.95 | 7.25 | 0.239 | 0.303 | *** |
| Llama 3.1 8B | 0.85 | 2.06 | 2.29 | 0.24 | 0.095 | * |
| DeepSeek R1 8B | 0.55 | 8.45 | 8.75 | 0.129 | 0.678 | *** |

Peak-performing similarity thresholds shifted downward compared to the agricultural domain: Mistral 7B v0.3 and Granite 3.2 8B peaked at 0.80 (vs. 0.95 in Phase III), Llama 3.1 8B at 0.85 (vs. 0.90), and DeepSeek R1 8B at 0.55 (vs. 0.90). Improvement magnitudes increased substantially, with Mistral 7B v0.3 nearly tripling (+13.32% vs. +4.58%) and Granite 3.2 8B increasing more than fivefold (+6.95% vs. +1.25%). DeepSeek R1 8B achieved the highest Balance Score (0.678) due to its combination of strong improvement magnitude and notably low CV.

All four models exhibited perfect threshold alignment across CPS, T-CPS, and Balance Score, contrasting with the divergent optima observed for DeepSeek R1 8B in Phase III. Significance patterns shifted substantially: Mistral 7B v0.3 achieved the broadest positive response (9/10 thresholds with significant improvement), whereas Granite 3.2 8B showed a narrower response (3/10 significant). Llama 3.1 8B exhibited sensitivity to threshold miscalibration, with 4 thresholds producing significant degradation against a single threshold

(0.85) yielding significant improvement. Model-specific threshold selection produced CPS improvements ranging from 2.06% to 13.32% over baseline configurations without requiring architectural modifications.

These results are bounded by the tested configuration: four models, 426 questions, biodiversity domain, and standardized M1 Mac Mini hardware with 16,000-token context buffer. The controlled hardware environment eliminates context buffer variation as a confounding factor but limits generalizability to other execution environments. As demonstrated in Section 4.4.9, best-performing thresholds require domain-specific calibration; the downward shift in peak-performing thresholds and the increased improvement magnitudes indicate that corpus characteristics materially affect threshold sensitivity patterns. These results provide evidence for RQ1 (similarity threshold effects on generation quality) and RQ2 (model-dependent threshold sensitivity) in the biodiversity domain, and for RQ3 (whether comparable similarity threshold ranges hold across knowledge domains) through comparison with Phase III agricultural results.

## 4.5 Chapter Summary

The experimental evaluation examined similarity threshold effects on RAG generation quality across seven open-source language models and two knowledge domains. Four phases of increasing scope addressed the research questions defined in the Introduction.

Phase I assessed the PaSSER platform under fixed top-k retrieval, establishing baseline system functionality and runtime profiles for three 7B-parameter models (Mistral 7B, Llama 2 7B, Orca 2 7B) across two hardware environments.

Phase II introduced threshold-aware retrieval (0.50–0.80) as a pilot study, demonstrating model-specific CPS variation and confirming the composite scoring approach.

Phase III expanded the evaluation to four models (Mistral 7B v0.3, Granite 3.2 8B, Llama 3.1 8B, DeepSeek R1 8B) and extended the threshold range to 0.95, addressing RQ1 (similarity threshold effects on generation quality) and RQ2 (model-dependent threshold sensitivity). Peak-performing thresholds were model-dependent, ranging from 0.90 to 0.95 (CPS criterion), with CPS improvements between 1.01% and 4.58% over baseline. The T-CPS and Balance Score framework enabled identification of configurations balancing performance improvement with output stability.

Phase IV repeated the same experimental procedure on a biodiversity corpus (N = 426), addressing RQ3 (whether comparable similarity threshold ranges hold across knowledge domains). Peak-performing thresholds shifted lower (0.55–0.85) compared to the agricultural domain (0.90–0.95), with larger CPS improvements (2.06%–13.32%) and higher output variability. Domain-specific threshold calibration appears necessary for open-source deployment decisions.

Across all phases, the evaluation procedure (a), infrastructure (b), and experimental design (c) components of Objective 1 were applied consistently. The controlled evaluations fulfill Objective 4 and provide empirical evidence addressing Deficiency 1 (threshold-aware evaluation), Deficiency 2 (reproducibility infrastructure), and Deficiency 3 (practical guidance), with best-performing thresholds per model and domain (Tables 4.13 and 4.21) enabling evidence-based configuration decisions for open-source RAG deployments. Cross-domain findings indicate that best-performing thresholds are not universally transferable and require corpus-specific calibration.

# CHAPTER 5: DISCUSSION AND FUTURE WORK

The dissertation is guided by one research aim and four objectives. Its aim is to develop an evaluation framework for RAG that supports evidence-based retrieval configuration decisions for RAG systems with open-source LLMs, with particular focus on similarity threshold configuration. Objective 1 defines and implements the core components of this framework: a threshold-aware evaluation procedure with composite scoring, the PaSSER platform as reproducibility infrastructure with blockchain-based provenance logging, and a controlled experimental design producing comparative evidence across models and domains. Objective 2 establishes model selection criteria aligned with local deployment feasibility, licensing constraints, and computational requirements. Objective 3 defines the selection of evaluation metrics and the procedures for their consistent computation across models and experimental conditions. Objective 4 conducts controlled testing and analysis through the preparation of domain corpora and question-answer datasets, systematic parameter variation, including similarity threshold sweeps, and comparative interpretation of the results. Chapter 1 identifies the research gaps and formulates the three deficiencies addressed by the framework. Chapter 2 presents the PaSSER platform and the supporting infrastructure. Chapter 3 defines the model selection rationale, evaluation metrics, and computation procedures. Chapter 4 reports the controlled experiments across four phases, applying the evaluation procedure, infrastructure, and experimental design components of the framework to produce empirical evidence for the research questions. The findings are synthesized below: Section 5.1 summarizes the evidence for each research question, Section 5.2 outlines the three scientific-applied contributions, Section 5.3 discusses limitations, and Section 5.4 identifies directions for future research.

## 5.1 Answers to Research Questions

Subsections 5.1.1–5.1.3 synthesize experimental evidence from Phases II–IV in relation to the three research questions.

### 5.1.1 Threshold Effects on Generation Quality (RQ1)

Similarity threshold configuration produced statistically significant effects on generation quality. Across agricultural and biodiversity corpora, threshold variation within 0.50–0.95 yielded

CPS improvements of up to +4.58% (Phase III, agriculture; Table 4.7) and +13.32% (Phase IV, biodiversity; Table 4.14) relative to the baseline.

Phase II pilot experiments (101 question-answer pairs, three 7B-parameter models, threshold range 0.50–0.80) established that CPS varied across threshold configurations even within a narrow range, motivating extended evaluation in subsequent phases. Peak-performing thresholds differed by model: Mistral 7B and Llama 2 7B at 0.55; Orca 2 7B at 0.65.

Phase III extended the analysis with statistical significance testing across 369 questions, four models, and an expanded threshold range (0.50–0.95). Statistically significant differences relative to the baseline were observed across multiple threshold configurations: Granite 3.2 8B (7/10 thresholds) and Mistral 7B v0.3 (6/10) exceeded chance expectation under uncorrected multiple testing (Section 5.3.3), suggesting systematic threshold sensitivity. Llama 3.1 8B (3/10) and DeepSeek R1 8B (2/10) approached the chance floor at the individual-comparison level; however, the direction and magnitude of their effects are contextualized in Sections 5.1.2.

Phase IV applied the same evaluation procedure to the biodiversity corpus (426 questions) under a different domain corpus and hardware configuration (Section 5.3.2). Threshold effects were consistent in direction with Phase III but differed in magnitude and variability. Mistral 7B v0.3 showed statistically significant improvements at 9/10 thresholds. Granite 3.2 8B contracted from 7/10 significant thresholds in Phase III to 3/10. DeepSeek R1 8B exhibited bifurcated behavior, with significant improvements at lower thresholds (0.55–0.60) and significant degradation at higher thresholds (0.90–0.95). Llama 3.1 8B exhibited predominantly negative threshold effects (5 significant negative, 1 significant positive). Output variability increased in Phase IV relative to Phase III across all models, as reflected in higher coefficient of variation values (Section 4.4.9).

Taken together, these findings provide an affirmative answer to RQ1: similarity threshold configuration produced measurable and, in Phases III–IV, statistically significant effects on generation quality. CPS improvements of up to +4.58% (agriculture) and +13.32% (biodiversity) were achieved through threshold calibration alone, without architectural modifications or model retraining. However, peak-performing threshold values were model-specific and domain-dependent (Sections 5.1.2 and 5.1.3), indicating that configuration-specific testing is necessary rather than reliance on fixed threshold settings.

### 5.1.2 Model-Dependent Similarity Threshold Sensitivity (RQ2)

Substantial model-dependent variation was observed in threshold sensitivity. Models differed in peak-performing threshold values, breadth of effective threshold ranges, magnitude of CPS improvements, and output consistency across threshold configurations. The per-model patterns reported in Section 5.1.1 are compared below to characterize the nature and extent of this variation.

Mistral 7B v0.3 exhibited the broadest and most consistent threshold sensitivity. In Phase III (agriculture), statistically significant improvements were observed at 6/10 thresholds spanning 0.55–0.95. In Phase IV (biodiversity), responsiveness increased to 9/10 significant thresholds, with peak improvement of +13.32% CPS at threshold 0.80. Among the four models evaluated in Phases III–IV, Mistral 7B v0.3 was the only model to maintain broad positive sensitivity across both domains.

Granite 3.2 8B exhibited plateau behavior in Phase III (7/10 significant thresholds across 0.65–0.95) but contracted to 3/10 significant thresholds in Phase IV. This contraction — from the highest significance count in Phase III to a level near chance — represents the largest cross-domain shift in sensitivity breadth among the evaluated models, though Phase IV confounds (Section 5.3.2) complicate interpretation.

Llama 3.1 8B showed constrained positive threshold sensitivity in Phase III (3/10 significant thresholds, concentrated at high values) and predominantly negative effects in Phase IV (6/10 significant: 5 negative, 1 positive). This pattern inversion — from limited positive to predominantly negative — was unique among the evaluated models and suggests that threshold calibration for Llama 3.1 8B is particularly sensitive to corpus characteristics or experimental conditions.

DeepSeek R1 8B showed limited responsiveness in Phase III (2/10 significant thresholds) but stronger effects in Phase IV with bifurcated behavior: significant improvements at lower thresholds (0.55–0.60) and significant degradation at higher thresholds (0.90–0.95). Low CV values (0.129–0.166) relative to other models contributed to the highest Balance Score among all model-threshold combinations in Phase IV (0.678 at threshold 0.55; Table 4.21), indicating that when stability-weighted scoring is applied, DeepSeek R1 8B emerges as a competitive option despite limited mean CPS improvement.

Taken together, these patterns provide an affirmative answer to RQ2: threshold sensitivity differed substantially across models in both magnitude and character. Responsiveness ranged from broad and cross-domain stable (Mistral 7B v0.3) to narrow and domain-sensitive (Granite 3.2 8B), to inverted across domains (Llama 3.1 8B), to bifurcated with high output consistency (DeepSeek R1 8B). These differences indicate that threshold selection cannot be generalized across model architectures; configurations effective for one model may degrade generation quality for another. The extent to which these model-dependent patterns interact with corpus characteristics is examined in Section 5.1.3.

### 5.1.3 Cross-Domain Similarity Threshold Comparison (RQ3)

Taken together, these findings provide an answer to RQ3 indicating that comparable similarity threshold ranges did not hold across the evaluated knowledge domains under the tested setup. Systematic differences were observed across multiple dimensions, though interpretation is complicated by simultaneous changes in question generation procedure, hardware configuration, and corpus characteristics (Section 5.3.2).

Peak-performing thresholds shifted systematically lower in Phase IV. All four models achieved peak CPS at lower thresholds in the biodiversity domain than in agriculture, with shifts ranging from −0.05 (Llama 3.1 8B) to −0.35 (DeepSeek R1 8B; Table 4.20). This consistent downward pattern suggests that the biodiversity corpus or experimental conditions favored less selective retrieval, though the specific factors driving this shift were not isolated experimentally.

Improvement magnitudes differed substantially between phases. Three models achieved larger CPS improvements in Phase IV: Mistral 7B v0.3 (+13.32% vs. +4.58%), Granite 3.2 8B (+6.95% vs. +1.25%), and DeepSeek R1 8B (+8.45% vs. +1.01%). Llama 3.1 8B showed comparable improvement magnitudes (+2.06% vs. +1.58%), though its Phase IV results were predominantly negative threshold effects.

Sensitivity profiles also shifted between phases, as detailed in Section 5.1.2. Model rankings by peak CPS improvement changed: Mistral 7B v0.3 remained first, DeepSeek R1 8B rose from fourth to second, and Llama 3.1 8B dropped from second to fourth.

Metric-level correlation patterns differed between phases. Lexical metrics clustered more tightly in Phase IV: METEOR–ROUGE correlations increased from 0.01–0.23 (Phase III; Figure 4.3)

to 0.83–0.84 (Phase IV; Figure 4.7). Perplexity-quality correlations reversed sign from positive (ρ = 0.2–0.5, Phase III) to negative (ρ = −0.3 to −0.06, Phase IV). These shifts suggest that metric interdependencies are corpus-sensitive, though whether this reflects domain characteristics or other experimental differences remains unclear.

Taken together, these findings provide an answer to RQ3 indicating that threshold ranges were not transferable across the evaluated domains under the tested setup. Peak-performing thresholds, improvement magnitudes, sensitivity profiles, and metric correlation structures all shifted between corpora. The systematic direction of threshold shifts across all four models suggests interactions between corpus characteristics (or experimental conditions) and model behavior. However, the confounds documented in Section 5.3.2 prevent attribution of these shifts specifically to domain properties.

## 5.2 Scientific-applied Contributions

Three scientific-applied contributions, organized as layers of an integrated evaluation framework, address the deficiencies identified in the Introduction: a threshold-aware evaluation procedure with composite scoring (C1, Evaluation Procedure layer, addressing D1), reproducibility infrastructure with blockchain-based provenance logging (C2, Infrastructure layer, addressing D2), and practical guidance for open-source deployments (C3, Evidence layer, addressing D3). Subsections 5.2.1–5.2.3 characterize each.

### 5.2.1 Threshold-aware Evaluation Procedure

Section 1.4 identified that existing evaluation frameworks (RAGAS, RGB, TREC RAG Track, TruLens) evaluate outputs under fixed retrieval configurations and do not treat similarity threshold as a first-class experimental variable (Section 1.4.6). In many frameworks, threshold configuration is treated as an implementation detail rather than as a parameter requiring systematic characterization. Frameworks typically report mean performance without quantifying stability across queries — a gap that Threshold-aware Composite Performance Score (T-CPS) and Balance Score address through explicit variability assessment. The framework therefore treats similarity threshold as an explicit independent variable and reports threshold–response evidence rather than single-point performance.

Section 1.3.8 noted that RAG implementations typically rely on heuristic threshold selection, with some systems adopting 0.7 as a default without systematic justification. Peak-performing thresholds varied substantially across evaluated models (0.90–0.95 by CPS criterion, 0.65–0.95 when T-CPS and Balance Score optima are included; Phase III) and domains (0.55–0.85 in Phase IV), consistent with concerns about fixed defaults. Llama 3.1 8B exhibited performance degradation at both low (0.60) and high (0.95) thresholds while achieving peak performance at 0.90, illustrating model-specific sensitivity patterns that cannot be captured without threshold variation.

A threshold-aware evaluation procedure was developed, addressing the threshold characterization gap (Deficiency 1). Controlled threshold variation was operationalized through the PaSSER platform (Chapter 2), while composite scoring (Composite Performance Score [CPS], T-CPS, Balance Score) was implemented through separate Python scripts. Performance differences across thresholds ranged up to +4.58% CPS improvement in the agricultural domain (Phase III; Table 4.7) and +13.32% in the biodiversity domain (Phase IV; Table 4.14), indicating that evaluation results obtained under single threshold configurations may not characterize system behavior across the retrieval selectivity range.

The threshold-aware evaluation procedure was applied across three experimental phases (Phases II–IV, Chapter 4), producing systematic evidence for model-specific and domain-specific threshold sensitivity patterns. Phase I established baseline system functionality, Phase II piloted threshold variation (0.50–0.80), Phase III extended the threshold range and introduced statistical significance testing (0.50–0.95, agricultural domain), and Phase IV assessed cross-domain generalization (biodiversity domain). The threshold sensitivity findings constitute exploratory results; confirmatory studies with appropriate statistical correction would strengthen generalizability (Section 5.3.3).

The CPS formulation, introduced in Phase II and refined in Phase III, aggregates heterogeneous metrics through weighted summation after normalization. The 9-metric weighting scheme allocates lexical overlap (30%), semantic similarity (25%), fluency and accuracy (25%), and language modeling (20%), providing coverage across multiple quality dimensions.

T-CPS incorporates coefficient of variation (CV) to penalize high output variability, reflecting the concern that configurations with high mean performance but inconsistent outputs

may be less suitable for deployment. Balance Score quantifies improvement relative to variability: *Balance Score = (T-CPS improvement % / 100) / CV.* These metrics provide information for deployment decisions where output consistency matters alongside mean performance.

Across Phases III and IV, T-CPS and CPS exhibited very high correlation ($\rho > 0.99$; Tables 4.9 and 4.16). For the models and thresholds evaluated, configurations with higher mean performance tended to exhibit lower output variability, producing alignment between CPS and T-CPS rankings.

This high correlation reflects characteristics of the evaluated model set and threshold range rather than redundancy in the T-CPS formulation. T-CPS incorporates stability considerations through explicit CV-based penalization, enabling identification of configurations where mean performance and output consistency diverge. When models exhibit similar variability profiles across thresholds, CPS and T-CPS rankings converge; when variability profiles differ, T-CPS identifies different optima.

DeepSeek R1 8B in Phase III provides an example of such divergence: peak CPS occurred at threshold 0.90, while peak T-CPS occurred at threshold 0.65 (Section 4.3.4, Table 4.11). DeepSeek's low CV values in Phase III (0.085–0.108; Table 4.13) meant that stability penalization reduced its T-CPS less than for other models, shifting the optimum toward a threshold with lower mean performance but greater output consistency. Across the broader evaluation, such divergences were infrequent, though the potential for divergence motivated retention of both metrics in results reporting.

Overall, this contribution addresses Deficiency 1 by replacing fixed-threshold evaluation with a threshold-aware procedure that captures both performance level and performance stability. The reported findings remain exploratory, and confirmatory studies with stronger statistical control would further strengthen generalizability.

### 5.2.2 Reproducibility Infrastructure

The PaSSER platform (Performance Assessment System for Similarity Evaluation and Retrieval) was developed to address reproducibility challenges in RAG evaluation (Chapter 2, Objective 1(b)). The platform implements controlled threshold variation, multi-metric

assessment, and blockchain-based provenance logging through a browser-based interface with backend services coordinating vector storage, model inference, and evaluation computation.

Blockchain-based provenance logging was integrated as a core component of the evaluation workflow. Each experimental run records evaluation metrics, timing data, and run identifiers on the Antelope blockchain, while the backend captures the associated configuration context (model identifier, retrieval parameters, decoding settings, and dataset identifiers). Provenance logging is implemented as default behavior in PaSSER rather than an optional feature, ensuring that reported results are linked to fully specified configurations.

Three properties relevant to reproducibility are provided: immutability (logged configurations cannot be retroactively modified), timestamping (the sequence of runs is verifiable), and accessibility (logged data can be retrieved for independent analysis). While blockchain logging does not guarantee identical replication of computational environments, it records experimental conditions in a form that supports verification. This infrastructure responds to the reproducibility gap identified in Section 1.4.6, where configuration complexity across interacting layers (corpus preprocessing, embedding, retrieval, generation, evaluation) complicates independent replication of RAG experiments.

The materials associated with this dissertation are publicly available in two GitHub locations. The phase-organized thesis archive, containing the experimental results and supporting research materials, is available at https://github.com/M33rschaum/passer-thesis-archive.

The current PaSSER implementations, including the original PaSSER platform and its related repositories such as maPaSSER and the PaSSER-SR, are available through the GitHub organization at https://github.com/scpdxtest.

This open-source release enables independent verification of reported results.

### 5.2.3 Practical Guidance for Open-Source Deployments

Comparative evidence characterizing threshold sensitivity across seven open-source LLMs (7–8 billion parameters) was collected under controlled conditions, addressing the practical guidance gap for open-source deployments (Section 1.4.6). Model selection criteria and metric computation procedures (Chapter 3, Objectives 2–3) enabled systematic comparison, while the

controlled experimental design (Objective 1(c), Objective 4) made threshold-aware performance visible and comparable across deployment candidates in resource-constrained settings.

Over 38,000 evaluations across four experimental phases, two domains, and multiple hardware configurations (Chapter 4) generated evidence linking threshold sensitivity, generation quality, and deployment feasibility. Four patterns inform practical guidance. First, model-specific threshold responsiveness patterns — ranging from broad and consistent (Mistral 7B v0.3, significant across 6–9 of 10 thresholds) to constrained and bifurcated (DeepSeek R1 8B, significant at 2 thresholds in Phase III) — inform model selection. Second, effective threshold ranges varied by model and phase (0.55–0.95 for Mistral 7B v0.3 in Phase III; 0.65–0.95 for Granite 3.2 8B in Phase III), providing starting configurations for threshold calibration. Third, domain-specific sensitivity patterns, including systematic downward threshold shifts and changes in improvement magnitude and direction across corpora, characterize the extent of cross-domain variation. Fourth, improvement-consistency trade-offs quantified through Balance Score rankings support deployment planning under stability constraints.

These findings provide practical guidance for threshold selection and model configuration within the documented scope: exploratory statistical approach, hardware heterogeneity, domain-specific confounds, and evaluation limited to the 7–8B parameter range in agricultural and biodiversity domains.

Section 1.5 reviewed seven RAG failure points documented in prior work [88]. Threshold sensitivity patterns observed in Phases II–IV relate primarily to two context quality failure points: missed top-ranked documents (when thresholds exclude relevant passages that fall below the similarity cutoff) and retrieval of weakly related content (when permissive thresholds admit passages that distract generation). Model-specific sensitivity profiles suggest that susceptibility to these failure points may be model-dependent, though isolating causal mechanisms would require controlled ablation studies beyond the documented experimental scope.

## 5.3 Limitations

Several factors constrain the generalizability of the findings and should be considered when interpreting results beyond the evaluated corpora, models, and configurations.

### 5.3.1 Scope Constraints

The evaluation scope was limited to specific domains and model types.

***Domain Scope***

Two knowledge domains were evaluated: agriculture (Phase III) and biodiversity (Phase IV). Both domains contain technical and regulatory content with formal language and well-defined terminology. Whether similar threshold sensitivity patterns extend to other domain types—conversational content, creative writing, code generation, medical or legal text—remains untested. Threshold optima differed between agriculture and biodiversity evaluations (Section 4.3 and Section 4.4), suggesting that domain-specific factors influence retrieval-generation trade-offs. However, Phase IV introduced simultaneous changes to hardware, question generation method, and domain (Section 5.3.2), making it difficult to isolate domain effects from experimental confounds. Specific domain characteristics (e.g., terminology density, question complexity, document structure) that drive threshold sensitivity were not experimentally manipulated or measured.

***Model Scope***

The evaluation focused on open-source models with approximately 7–8 billion parameters to balance capability with computational feasibility on mid-range hardware. Models were selected according to criteria detailed in Section 3.1, prioritizing open-source availability and instruction-tuning capability. This scope excludes proprietary models, larger open-source models, and smaller models (<7B parameters). Threshold sensitivity may differ outside this range. Larger models with greater parametric knowledge may rely less on retrieved context, potentially reducing threshold sensitivity. Smaller models may exhibit different trade-offs between retrieval selectivity and generation quality. Proprietary models with different training procedures and architectural choices may respond differently to threshold variation. Within the evaluated parameter range, architectural differences across model families (Mistral, Llama, Orca, Granite, DeepSeek) were not systematically analyzed as independent variables.

### 5.3.2 Experimental Design Limitations

***Hardware Heterogeneity***

Experiments were executed across heterogeneous hardware configurations due to practical constraints and resource availability. Phase I and Phase II employed two platforms: Mac M1 with GPU acceleration (8 CPU cores, 10 GPU cores, 16 GB RAM) and Intel Xeon CPU-only (32 cores, 128 GB RAM; Section 4.1). Phase III introduced additional variability with M1 Mac Mini, M2 Mac Mini, and CPU-only configurations, with context buffer sizes varying by model (2048–10,000 tokens) based on memory constraints (Section 4.3). Phase IV (biodiversity domain) standardized hardware to a single Mac M1 configuration with fixed 16,000-token context buffer for all models, eliminating hardware and context buffer variation but introducing simultaneous changes to domain and question generation method (Section 4.4).

This hardware heterogeneity introduces confounds when comparing absolute performance metrics across models and phases. Statistical comparisons in Phase III were performed within each model relative to its baseline configuration to reduce hardware-related confounds, but minor hardware-specific effects on inference behavior cannot be fully excluded. The mixed-hardware design in Phases I–III reflects practical deployment realities but limits conclusions about absolute performance levels. The analysis emphasizes relative differences across threshold configurations within each phase, under the assumption—not empirically verified—that threshold-induced performance patterns are more stable across hardware than absolute metric values.

***Corpus Construction***

Vector stores were constructed using Mistral 7B embeddings via the Ollama embedding endpoint, with the same model used for query embedding at retrieval time. Source documents were segmented into fixed-length character chunks of 1,024 characters with 50-character overlap prior to embedding and insertion into ChromaDB. Character-based chunking was selected over sentence- or paragraph-level segmentation to produce uniform chunk lengths, simplifying threshold calibration analysis by reducing variability attributable to uneven passage granularity. The 1,024-character length balances retrieval specificity against context completeness: shorter chunks risk fragmenting coherent passages, while longer chunks dilute relevance signals when

only a portion addresses the query. The 50-character overlap mitigates hard boundary effects by preserving continuity across adjacent chunks. However, chunk size and overlap were held constant across all experimental phases. Whether threshold sensitivity patterns depend on passage granularity remains untested; this limitation is addressed through a proposed chunking ablation in Section 5.4.3.

This single-embedding-model design ensures consistent retrieval behavior across all evaluated generators: given identical queries and threshold settings, all models receive the same retrieved context. However, because Mistral 7B serves as both the embedding model and one of the evaluated generators, a potential confound exists. Retrieval may be systematically better aligned with the embedding model's learned representations, potentially favoring passage selections that suit its generation patterns. Peak CPS improvements (Sections 4.3.3 and 4.4.3) show Mistral 7B v0.3 achieving the largest gains in both the agricultural domain (+4.58%) and the biodiversity domain (+13.32%). Because Phase IV questions were generated by Claude Opus rather than Mistral 7B, the question-generation confound documented in above does not account for this pattern; the embedding-alignment confound remains the more plausible candidate explanation. However, this observation cannot distinguish a confound effect from genuinely superior generation capability — both may co-occur. Isolating the embedding confound would require controlled ablation with independent embedding models, as outlined in Section 5.4.3.

***Question Generation***

The agricultural domain questions (Phases I–III) were generated using Mistral 7B, which is also one of the evaluated models. This creates a potential confound: Mistral 7B may exhibit inflated performance because the questions were formulated in patterns that align with its generation tendencies (acknowledged in Section 4.1.1). Several considerations partially mitigate this concern. First, questions were generated from source documents and reviewed for factual accuracy, reducing the influence of model-specific phrasing preferences. Second, the relative ranking of models is more robust than absolute scores—if Mistral 7B benefits from question familiarity, this affects its absolute CPS but not necessarily the threshold sensitivity patterns observed across models. Third, Phase IV uses a different question set (biodiversity domain with Claude Opus generation), providing partial cross-validation. Mistral 7B achieved higher peak performance in Phase IV (13.32% CPS improvement at threshold 0.80) than in Phase III (Table

4.20), suggesting the question generation method did not systematically inflate agricultural domain results. Nevertheless, the question generation procedure represents a limitation, and whether the mitigation strategies fully eliminate bias was not empirically assessed through controlled experimentation with human-authored questions.

***Cross-Domain Comparison Confounds***

Phase IV (biodiversity domain) differs from Phase III (agricultural domain) in multiple respects beyond domain content. Question sets were generated using different models: Mistral 7B for agriculture (Phases I–III) versus Claude Opus for biodiversity (Phase IV). Corpus characteristics vary between domains, including vocabulary density, document length distributions, and terminology patterns. Hardware configuration changed: Phase III employed heterogeneous platforms (M1, M2, CPU-only) with variable context buffers (2048–10,000 tokens), while Phase IV standardized to M1 hardware with fixed 16,000-token context across all models.

These simultaneous changes make it difficult to attribute observed threshold shifts specifically to domain properties. Peak-performing thresholds shifted downward for all models in Phase IV relative to Phase III: DeepSeek R1 8B (0.90 → 0.55), Mistral 7B v0.3 (0.95 → 0.80), Granite 3.2 8B (0.95 → 0.80), and Llama 3.1 8B (0.90 → 0.85; Section 4.4.9). These shifts could reflect genuine domain-dependent threshold sensitivity, question generation procedure differences, corpus structural differences, hardware-induced variance, or interactions among these factors.

Isolating domain effects would require controlled experiments where only the corpus changes while question generation procedure, hardware, and other factors remain constant. The cross-domain findings should therefore be interpreted as preliminary evidence suggesting domain-dependent calibration may be necessary, rather than as definitive characterization of domain-specific threshold effects.

## 5.3.3 Measurement and Analysis Limitations

***Perplexity Measurement Approach***

The perplexity metrics (Laplace Perplexity and Lidstone Perplexity) are computed using classical n-gram language models implemented in NLTK (nltk.lm) rather than from the evaluated transformer models' token-level probabilities (Section 3.4.3). Laplace Perplexity uses a bigram

model (order=2) with Laplace (add-one) smoothing, while Lidstone Perplexity uses a trigram model (n=3) with Lidstone smoothing (λ=0.1). The n-gram model is trained on evaluation reference text, and perplexity is computed on the generated candidate text under the resulting n-gram distribution. This approach was adopted for computational efficiency and to avoid introducing evaluation bias from using one LLM to assess another. These perplexity values are model-independent proxies under the fitted n-gram distribution and should not be interpreted as the evaluated models' internal confidence.

The practical implication is that perplexity scores reflect deviation from standard English n-gram distributions rather than model confidence in generated outputs, functioning as linguistic typicality indicators rather than direct generation quality measures. The relatively low weights assigned to perplexity in the CPS formulation (7.5% each) reflect this proxy status.

***Coherence Proxy Limitations***

The B-RT coherence and fluency metrics rely on [CLS] embedding projections from a pretrained BERT-base encoder rather than sentence-level discourse analysis (Section 3.4.5). This implementation has three limitations.

First, B-RT coherence cannot diagnose specific discourse faults such as dangling references or abrupt topic shifts within long generations; it functions as a stable scalar proxy rather than a fine-grained discourse analyzer.

Second, the projection-based fluency signal substitutes for token-level perplexity, reducing sensitivity to subtle grammatical errors in otherwise strong outputs.

Third, correlation between B-RT scores and actual human judgments of readability has not been empirically validated. The B-RT suite was adopted as a practical approximation enabling scalable evaluation across large experimental matrices (369–426 question-answer pairs per phase, 10 thresholds, 7 models). B-RT scores should be interpreted as comparative signals within the evaluation pipeline rather than as direct measures of perceived quality. Deployments requiring validated human-alignment guarantees should complement B-RT with direct human evaluation or employ alternative implementations with established human correlation benchmarks.

***Statistical Considerations***

Phases III and IV employ formal statistical significance testing with reported p-values uncorrected for multiple comparisons to preserve statistical power (Section 3.5.3). Each phase includes 4 models × 10 thresholds = 40 pairwise comparisons against baseline. At $\alpha = 0.05$, approximately 2 significant results per phase would be expected by chance alone under the null hypothesis. Readers should interpret the overall pattern of results—the consistency of significant effects across thresholds within a model, the coherence of findings across phases, and the alignment of statistical significance with effect size magnitudes—rather than relying on individual p-values in isolation.

The threshold sensitivity findings constitute exploratory results that identify threshold variation as a phenomenon warranting further investigation rather than definitive threshold prescriptions for deployment.

## 5.3.4 Causal Interpretation

The experimental design established empirical associations between similarity thresholds and generation performance, including model-dependent threshold optima (ranging from 0.55 to 0.95), domain-dependent threshold shifts (agriculture vs. biodiversity), and stability-performance trade-offs quantified through Balance Score analysis.

However, causal mechanisms underlying these patterns—such as whether threshold effects arise from embedding space geometry, attention mechanism constraints, parametric knowledge reliance, or retrieval dependency—were not experimentally isolated. Multiple factors vary across experimental conditions (corpus characteristics, question formulation, hardware configuration, model architecture), preventing attribution of observed effects to single causal mechanisms. Controlled ablation studies that systematically manipulate individual architectural or algorithmic components while holding other factors constant would be required to distinguish between competing mechanistic explanations.

The threshold sensitivity findings provide empirical evidence for deployment optimization and identify mechanistic investigation as a valuable direction for extending the analysis beyond descriptive characterization toward causal understanding of threshold effects in retrieval-augmented generation.

## 5.4 Future Work

The evaluation framework and empirical findings open several directions for future research. Extensions are organized into three categories: scope extensions addressing the domain and model constraints documented in Section 5.3.1, procedural extensions improving evaluation procedures and infrastructure, and validation efforts establishing empirical grounding for measurement assumptions and causal claims. Scope extensions (5.4.1) propose evaluation across additional domains and model parameter ranges. Procedural extensions (5.4.2) include adaptive threshold selection mechanisms, platform infrastructure enhancements integrating blockchain and IPFS capabilities beyond provenance logging, and systematic characterization of efficiency-quality trade-offs. Validation and verification efforts (5.4.3) address measurement limitations documented in Section 5.3.3 through human validation studies and alternative measurement approaches, and enable causal investigation of threshold effects outlined in Section 5.3.4 through controlled experimental designs.

### 5.4.1 Scope Extensions

***Extended Domain Coverage***

Threshold sensitivity analysis could be extended to additional domains beyond the agricultural and biodiversity corpora evaluated in Phases III and IV. Candidate domains include conversational and educational content, general knowledge retrieval, and technical documentation. These domains differ in vocabulary density, document structure, question complexity, and terminology patterns—factors hypothesized to influence effective threshold configurations but not systematically measured in Phases III–IV.

Controlled domain variation, where corpus content changes while question generation, hardware configuration, and evaluation procedures remain constant, could isolate domain-specific threshold effects from the confounds documented in Section 5.3.2. Such evidence would clarify whether the threshold shifts observed between agriculture and biodiversity (−0.05 to −0.35 across models, Section 4.4.9) reflect general domain-dependent calibration requirements or are artifacts of simultaneous experimental changes. Comparative analysis across domains with measured characteristics could support predictive approaches for threshold selection in untested domains.

***Extended Model Coverage***

Threshold sensitivity characterization could be extended to models outside the 7–8 billion parameter range (Section 5.3.1). Smaller models (<7B parameters) and larger models (>8B parameters) could be evaluated to determine whether threshold sensitivity scales with model capacity, though such extension would require infrastructure beyond the mid-range hardware configurations used in Phases I–IV (Section 5.3.2). Comparison with proprietary models under matched evaluation conditions could clarify whether observed patterns reflect general RAG behavior or are specific to open-source architectures, though API access costs, rate limits, and limited configuration transparency present practical barriers. Analysis of fine-tuned model variants, including domain-adapted and instruction-tuned versions with different training procedures, could isolate the influence of training procedure on threshold sensitivity independently of base architecture.

## 5.4.2 Procedural extensions

***Adaptive Threshold Selection***

The evaluation applied fixed similarity thresholds uniformly across all queries within each experimental phase (Sections 4.1–4.4). Adaptive threshold selection mechanisms could be investigated, where retrieval selectivity varies by query characteristics such as specificity, ambiguity, or complexity. For instance, highly specific queries (e.g., "What is the nitrogen content requirement for organic wheat certification?") may benefit from higher thresholds that prioritize precision, while broad exploratory queries (e.g., "What are sustainable farming practices?") may require lower thresholds to ensure adequate retrieval coverage. Predictive approaches could estimate suitable thresholds from extractable query features—including query length, named entity density, question type (factual vs exploratory), and lexical overlap with corpus vocabulary—enabling dynamic adjustment without manual per-query calibration.

Such mechanisms could improve performance by matching retrieval behavior to query requirements, though effectiveness would depend on accurate feature extraction and threshold prediction.

***Platform Infrastructure Extensions***

The PaSSER platform implemented blockchain-based provenance logging to record evaluation results and metadata as tamper-evident transactions via the Antelope blockchain (Section 2.2.3). Currently, blockchain integration logs completed evaluation metrics (accuracy and timing) but does not capture intermediate artifacts such as retrieved passages, similarity scores, or query-specific retrieval decisions.

Platform infrastructure could be extended to record retrieval-level provenance, logging which passages were retrieved for each query, at what similarity scores, and how context composition influenced generation. Such fine-grained logging would create a complete audit trail from query through retrieval to generation, enabling verification of individual retrieval decisions rather than only aggregate evaluation outcomes.

IPFS integration, mentioned in Section 2.2.3 as supporting future fine-tuning artifact storage, could be extended to enable content-addressed storage of corpora and evaluation datasets, where documents and question sets are referenced by cryptographic content hashes rather than file paths. Combined blockchain-IPFS integration could support dataset provenance alongside configuration provenance, strengthening reproducibility guarantees by ensuring that reported results correspond to verifiable, immutable corpus versions. These extensions would address the reproducibility challenges documented in Section 5.3.2 by providing infrastructure for fine-grained verification beyond the current configuration-level logging.

***Efficiency-Quality Trade-offs***

Generation quality was analyzed without systematic treatment of computational efficiency. The relationship between threshold configuration and inference cost could be characterized to identify configurations that offer favorable quality-efficiency trade-offs. Higher similarity thresholds reduce the number of retrieved passages, decreasing context length and thereby reducing generation inference time and memory consumption. Lower thresholds increase retrieval coverage but impose computational overhead through longer context processing.

Systematic measurement of inference latency, memory usage, and throughput across threshold configurations could quantify these trade-offs, enabling deployment decisions that

balance quality improvements against resource constraints. This analysis is particularly relevant for resource-constrained deployments where quality gains may need to be weighed against computational overhead, including scenarios where hardware limitations (Section 5.3.2) constrain the feasible operating range. Efficiency characterization could inform threshold selection when computational budget, rather than quality maximization, is the primary constraint.

### 5.4.3 Validation and Verification

***Human Validation Studies***

Automated metrics aggregated through CPS and T-CPS provided systematic performance assessment across threshold configurations (Sections 4.2-4.4). The B-RT coherence metric, introduced in Phase III (Section 4.3), relies on [CLS] embedding projections rather than sentence-level discourse analysis, and correlation with human readability judgments has not been validated (Section 5.3.3).

Human validation studies could assess whether threshold configurations identified as favorable by CPS/T-CPS align with human-perceived quality differences. Validation protocols could include pairwise preference judgments (presenting responses generated at different thresholds for the same query and eliciting human preferences), Likert-scale quality ratings across multiple dimensions (accuracy, coherence, relevance, conciseness), and correlation analysis between automated metric rankings and aggregated human preference orderings. Such assessment would determine whether metric-based threshold selection translates to practical improvements in user-perceived response quality, which is particularly important for deployment decisions where user satisfaction rather than metric optimization is the primary objective. Validation findings could also inform metric refinement, identifying which automated measures best predict human judgments and whether composite weighting schemes require adjustment to improve human-metric alignment.

***Controlled Causal Studies***

The experimental design established statistical associations between threshold configurations and generation performance but did not isolate causal mechanisms underlying observed effects (Section 5.3.4). Multiple confounds—including corpus construction procedure,

question generation procedures, hardware heterogeneity, and simultaneous parameter changes in cross-domain comparison—prevent attribution of threshold effects to specific architectural or algorithmic mechanisms.

Controlled ablation studies could isolate individual confounds through systematic experimental manipulation. Corpus construction ablations could compare retrieval performance when embedding models differ from generation models versus when they match, testing whether the Mistral-for-both-embedding-and-generation design (Section 5.3.2) introduces alignment bias. Chunking strategy ablations could vary segment length and overlap parameters to assess whether threshold sensitivity patterns depend on passage granularity, testing whether the fixed 1,024-character chunks with 50-character overlap used across all phases constrain or bias retrieval behavior at specific threshold ranges. Question generation ablations could hold domain and hardware constant while varying question source (Mistral 7B vs Claude Opus vs human-authored), isolating question generation effects from domain and infrastructure confounds.

Hardware standardization studies could eliminate the Phase I/II/III/IV heterogeneity documented in Section 5.3.2, establishing whether threshold sensitivity patterns remain stable across configurations or reflect hardware-specific inference behavior. Cross-domain ablations could vary corpus domain only while controlling question generation procedure and hardware, enabling isolation of domain-specific threshold effects from the simultaneous changes that confound Phase IV interpretation (Section 5.3.2).

Such controlled designs could test competing mechanistic explanations for threshold effects, including whether performance changes arise from retrieval coverage versus context dilution trade-offs, embedding space geometry versus attention mechanism constraints, or parametric knowledge reliance versus retrieval dependency patterns.

***Alternative Measurement Approaches***

Current measurement approaches impose limitations on interpretability and validation (Section 5.3.3). Perplexity metrics (Laplace and Lidstone) rely on NLTK n-gram language models rather than the evaluated transformers' token-level probabilities, functioning as model-independent proxies under fitted n-gram distributions (Section 3.4.3).

Alternative perplexity measurement could use transformer-native token probabilities, computing log-likelihood under the actual generation model's distribution rather than approximating with n-gram proxies. Such model-native measurement would capture generation confidence more directly, though it requires access to token-level logits that may be unavailable from API-based models. B-RT coherence measurement uses [CLS] embedding projections rather than sentence-level discourse analysis and has not been validated against human judgments (Section 5.3.3). Alternative coherence assessment could employ sentence-level discourse coherence models that detect specific discourse faults (dangling references, topic shifts, logical inconsistencies) or human annotation protocols that establish correlation between automated scores and perceived readability.

Statistical rigor could be improved through multiple comparison correction methods applied to Phases III-IV exploratory analyses (Section 5.3.3). Bonferroni correction or Benjamini-Hochberg false discovery rate (FDR) control could adjust p-value interpretation for the 40 pairwise comparisons per phase (4 models × 10 thresholds), reducing Type I error risk at the cost of decreased statistical power. Empirical validation of proxy metric assumptions—including whether n-gram perplexity correlates with transformer perplexity, whether B-RT coherence predicts human readability judgments, and whether CPS weighting schemes generalize across domains—could inform metric selection and aggregation procedures for future threshold-aware evaluations.

## 5.5 Chapter Summary

Experimental findings were synthesized across three research questions. Three contributions were characterized, mapping each to the deficiencies and prior work reviewed in Chapter 1. Scope constraints, experimental design limitations, measurement considerations, and causal interpretation boundaries were documented. Future work was organized into scope extensions, procedural extensions, and validation efforts addressing the documented limitations.

# CONCLUSION – RESUME OF THE OBTAINED RESULTS

An evaluation framework for retrieval-augmented generation (RAG) was developed, integrating three layers: an Evaluation Procedure layer introducing a threshold-aware evaluation procedure, an Infrastructure layer implementing reproducibility infrastructure, and an Evidence layer producing practical guidance for open-source deployments. The framework addresses three deficiencies identified in current RAG evaluation practice: the absence of threshold-aware evaluation, insufficient reproducibility infrastructure, and the lack of practical guidance for open-source RAG deployments.

The PaSSER platform was designed and implemented as a browser-accessible, open-source application supporting configurable retrieval parameter testing, multi-metric evaluation with composite scoring, and blockchain-based provenance logging via the Antelope ledger. Three composite scoring instruments were developed: the Composite Performance Score (CPS) for unified threshold comparison, the Threshold-aware Composite Performance Score (T-CPS) incorporating output consistency through a coefficient-of-variation-based reward-penalty structure, and the Balance Score quantifying the stability-performance trade-off.

Seven open-source language models in the 7–8 billion parameter range were evaluated across four experimental phases, two application domains (agriculture and biodiversity), and over 38,000 individual evaluations. Phase I validated end-to-end platform functionality under fixed top-k retrieval. Phase II introduced threshold-aware evaluation and provided initial evidence that threshold sensitivity varies across models. Phase III extended the analysis to four newer models across a broader threshold range (0.50–0.95) with statistical validation, revealing CPS improvements of up to 4.58% in the agricultural domain and identifying two distinct sensitivity profiles: broad improvement zones and narrow effective ranges. Phase IV replicated the evaluation in the biodiversity domain, where substantially larger threshold effects were observed — peak CPS improvements reached 13.32%. Peak-performing threshold configurations shifted downward for all four models when moving from agriculture to biodiversity, with shifts ranging from −0.05 (Llama 3.1 8B) to −0.35 (DeepSeek R1 8B). Output variability increased by 67–105% (coefficient of variation) relative to agriculture, confirming that threshold sensitivity is domain-dependent and not solely a model-intrinsic property.

CPS and T-CPS alignment analysis demonstrated that mean performance and consistency-aware assessment can yield divergent threshold recommendations, with two of four models showing different optima under the two scoring instruments.

All four objectives were addressed. Objective 1 was realized through three components: the reproducibility infrastructure with blockchain-based provenance logging (b, Chapter 2), the threshold-aware evaluation procedure with composite scoring (a, Chapters 3–4), and the controlled experimental design producing comparative evidence across models and domains (c, Chapter 4). Objective 2 was addressed through definition of model selection criteria aligned with deployment feasibility, licensing, and computational requirements (Chapter 3). Objective 3 was fulfilled through definition and consistent implementation of metric computation procedures across five evaluation constructs, with aggregation into three composite scoring instruments (Chapter 3). Objective 4 was achieved through controlled experimentation across four phases, two domains, and over 38,000 evaluations under systematic threshold variation (Chapter 4).

Three practical conclusions emerge from the experimental evidence. First, threshold calibration produces measurable and statistically significant improvements, but effective configurations depend on both model architecture and knowledge domain - no single threshold setting generalizes across all conditions. Second, consistency-aware scoring is recommended over mean-performance-only assessment, as it prevents selection of high-performing but unstable configurations. Third, systematic threshold evaluation should be repeated when changing the application domain, as peak-performing thresholds shifted downward for all four tested models when moving from agriculture to biodiversity.

Three scientific-applied contributions — together constituting the evaluation framework — result from the research. The end-to-end traceability from identified deficiencies through research questions and objectives to contributions is shown in Figure C.1.

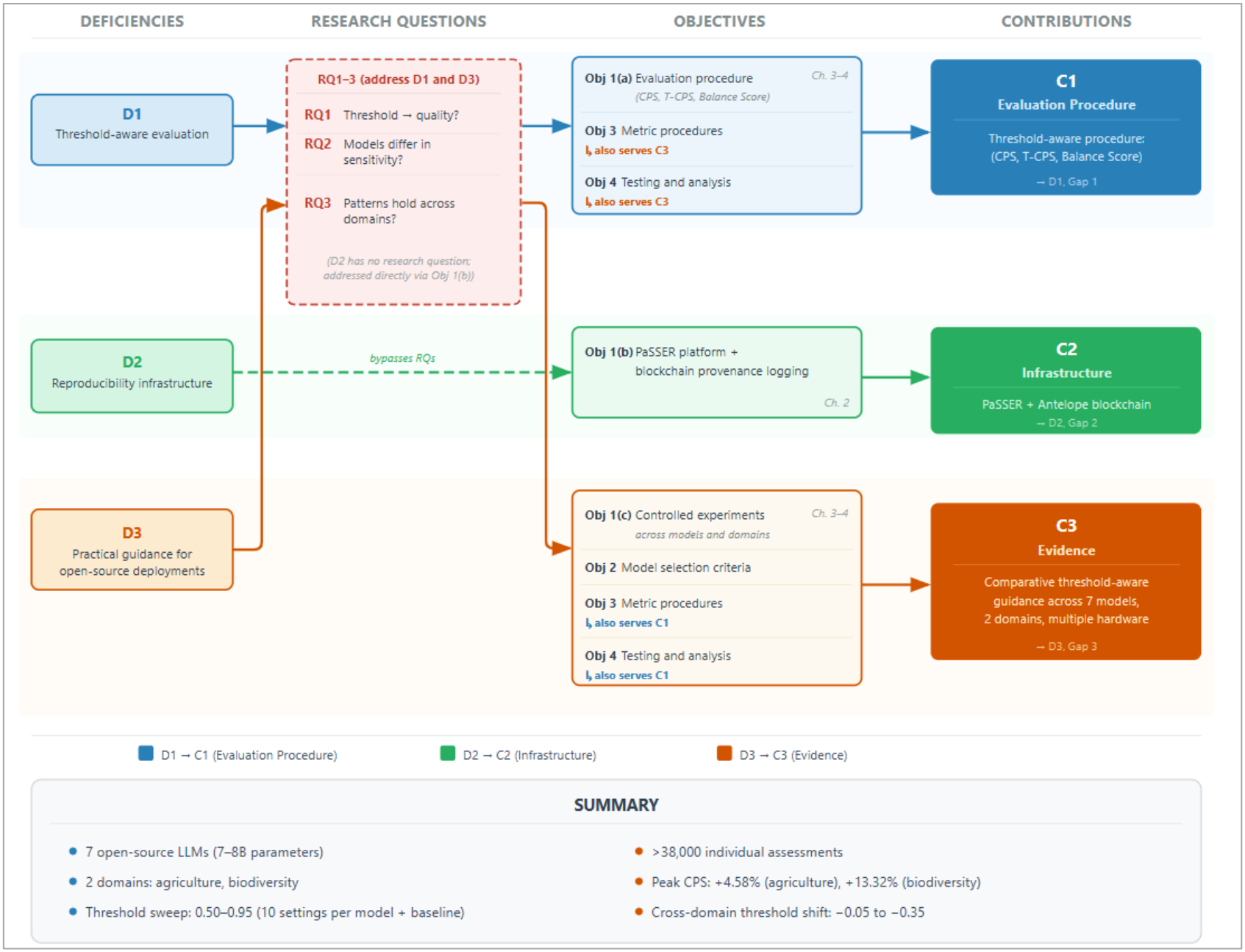


*Figure C.1 End-to-End Traceability Map*

# APPENDIX A. Key Algorithm Specifications

*Formal algorithmic specifications for the three core methodological components introduced in Chapter 3: the Composite Performance Score (CPS), the Threshold-aware Composite Performance Score (T-CPS), and the paired t-test procedure used for statistical significance testing. These specifications complement the conceptual descriptions in Sections 3.5.1–3.5.3. Complete source code implementations are available in the GitHub repository referenced in Section 5.2.*

## Algorithm A.1: Composite Performance Score (CPS)

```
Input:
    metrics     : dictionary {metric_name → raw_score}
    min_vals    : dictionary {metric_name → global_minimum}
    max_vals    : dictionary {metric_name → global_maximum}
    weights     : dictionary {metric_name → weight}, Σ weights = 1
    polarity    : dictionary {metric_name → +1 or −1}
                  (+1 = higher is better, −1 = lower is better)

Output:
    CPS score ∈ [0, 1]

Procedure:
    score ← 0

    FOR EACH metric m IN metrics DO
        d ← polarity[m]

        IF max_vals[m] = min_vals[m] THEN
            normalized ← 1.0
        ELSE
            normalized ← d × (metrics[m] − min_vals[m]) / (max_vals[m] − min_vals[m])
                                + (1 − d) / 2
        END IF

         score ← score + (weights[m] × normalized)
    END FOR

    RETURN score
```

*Note: The normalization formula ensures that for positive polarity (d = +1), higher values yield higher scores, while for negative polarity (d = −1), lower values yield higher scores. The term (1 − d)/2 adjusts the baseline for negative polarity metrics.*

## Algorithm A.2: Threshold-aware Composite Performance Score (T-CPS)

```
Input:
    CPS_scores  : array of CPS values across questions for a
                  given model-threshold configuration
    α           : reward coefficient (default = 0.1)
    β           : penalty coefficient (default = 0.05)

Output:
    T-CPS score

Procedure:
    μ ← mean(CPS_scores)
    σ ← standard_deviation(CPS_scores)

    IF μ ≠ 0 THEN
        CV ← σ / μ                          // Coefficient of Variation
    ELSE
        CV ← 0
    END IF

    consistency_factor ← max(0, 1 – CV)
    variance_penalty ← CV²

    T-CPS ← μ × (1 + α × consistency_factor) – β × variance_penalty

    RETURN T-CPS
```

*Note: The consistency factor rewards configurations with low variability (CV close to 0), while the variance penalty discourages high variability. The max(0, 1 − CV) ensures the consistency factor remains non-negative.*

## Algorithm A.3: Paired t-test with Effect Size

```
Input:
    baseline_scores : array of CPS values at baseline threshold
    test_scores     : array of CPS values at test threshold
    α_level         : significance level (default = 0.05)

Output:
    t_statistic, p_value, Cohen's_d, confidence_interval

Procedure:
    // Ensure equal sample sizes
    n ← min(length(baseline_scores), length(test_scores))
    baseline ← baseline_scores[1..n]
    test ← test_scores[1..n]

    // Calculate differences
    differences ← test − baseline

    mean_diff ← mean(differences)
    std_diff  ← sample_standard_deviation(differences)
    SE        ← std_diff / √n

    // t-statistic
    IF SE ≠ 0 THEN
        t ← mean_diff / SE
    ELSE
        t ← 0
    END IF

    // Degrees of freedom
    df ← n − 1

    // Two-tailed p-value
    p_value ← 2 × (1 − CDF_t(|t|, df))

    // Effect size (Cohen's d for paired samples)
    IF std_diff ≠ 0 THEN
        Cohen's_d ← mean_diff / std_diff
    ELSE
        Cohen's_d ← 0
    END IF

    // 95% Confidence Interval
    t_critical ← inverse_CDF_t(0.975, df)
    CI_lower ← mean_diff − (t_critical × SE)
    CI_upper ← mean_diff + (t_critical × SE)

    RETURN t, p_value, Cohen's_d, (CI_lower, CI_upper)
```

*Interpretation guidelines:*

- *Statistical significance: $p < 0.05$*
- *Effect size (|Cohen's d|): - negligible: < 0.2 - small: 0.2–0.5 - medium: 0.5–0.8 - large: ≥ 0.8*

# APPENDIX B. Phase III Supplementary Tables

*P-values are from two-tailed paired t-tests against baseline at the per-question level (uncorrected). Values below 0.001 are reported as p<0.001.*

**Table B.1** Statistical Analysis Results for Mistral 7B v.0.3 (Phase III, Agriculture Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.5215. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 369 | 0.5246 | 0.59 | 0.0047 | 0.663 | 368 | 0.508 | 0.034 | −0.0061 | 0.0122 | negl. | ns |
| 0.55 | 369 | 0.5306 | 1.75 | 0.0044 | 2.056 | 368 | 0.04 | 0.107 | 0.0004 | 0.0178 | negl. | * |
| 0.6 | 369 | 0.5316 | 1.94 | 0.0045 | 2.274 | 368 | 0.024 | 0.118 | 0.0014 | 0.0189 | negl. | * |
| 0.65 | 369 | 0.5231 | 0.32 | 0.0045 | 0.375 | 368 | 0.708 | 0.02 | −0.0071 | 0.0105 | negl. | ns |
| 0.7 | 369 | 0.5325 | 2.11 | 0.0044 | 2.47 | 368 | 0.014 | 0.129 | 0.0022 | 0.0197 | negl. | * |
| 0.75 | 369 | 0.5319 | 2 | 0.0045 | 2.317 | 368 | 0.021 | 0.121 | 0.0016 | 0.0192 | negl. | * |
| 0.8 | 369 | 0.5259 | 0.85 | 0.0046 | 0.974 | 368 | 0.331 | 0.051 | −0.0045 | 0.0134 | negl. | ns |
| 0.85 | 369 | 0.5264 | 0.95 | 0.0046 | 1.085 | 368 | 0.279 | 0.056 | −0.0040 | 0.0139 | negl. | ns |
| 0.9 | 369 | 0.5338 | 2.37 | 0.0045 | 2.738 | 368 | 0.006 | 0.143 | 0.0035 | 0.0212 | negl. | ** |
| 0.95 | 369 | 0.5454 | 4.58 | 0.0051 | 4.679 | 368 | <0.001 | 0.244 | 0.0138 | 0.0339 | small | *** |

**Table B.2** T-CPS Descriptive Metrics for Mistral 7B

| Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|
| 0.5 | 0.568607848 | 0.468654087 | 0.1412033 | Minimal change |
| 0.55 | 0.575985903 | 1.772299893 | 0.1315082 | Moderate improvement |
| 0.6 | 0.576810139 | 1.917936152 | 0.1350739 | Moderate improvement |
| 0.65 | 0.567720038 | 0.31178478 | 0.1329377 | Minimal change |
| 0.7 | 0.578261406 | 2.174363964 | 0.1283225 | Moderate improvement |
| 0.75 | 0.577604644 | 2.058319107 | 0.1276619 | Moderate improvement |
| 0.8 | 0.570535008 | 0.809168483 | 0.1363856 | Small improvement |
| 0.85 | 0.570667385 | 0.832558599 | 0.142685 | Small improvement |
| 0.9 | 0.57933651 | 2.36432659 | 0.1312421 | Moderate improvement |
| 0.95 | 0.591622007 | 4.535079808 | 0.1338613 | Large improvement |

**Table B.3** Statistical Analysis Results for Granite 3.2 8B (Phase III, Agriculture Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.5118. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 369 | 0.5142 | 0.48 | 0.0021 | 1.166 | 368 | 0.244 | 0.061 | −0.0017 | 0.0066 | negl. | ns |
| 0.55 | 369 | 0.5152 | 0.66 | 0.0019 | 1.786 | 368 | 0.075 | 0.093 | −0.0003 | 0.0071 | negl. | ns |
| 0.6 | 369 | 0.5145 | 0.53 | 0.0022 | 1.239 | 368 | 0.216 | 0.064 | −0.0016 | 0.0071 | negl. | ns |
| 0.65 | 369 | 0.5164 | 0.9 | 0.0021 | 2.212 | 368 | 0.028 | 0.115 | 0.0005 | 0.0087 | negl. | * |
| 0.7 | 369 | 0.5179 | 1.2 | 0.0021 | 2.877 | 368 | 0.004 | 0.149 | 0.0019 | 0.0103 | negl. | ** |
| 0.75 | 369 | 0.5174 | 1.1 | 0.0021 | 2.742 | 368 | 0.006 | 0.142 | 0.0016 | 0.0097 | negl. | ** |
| 0.8 | 369 | 0.5178 | 1.17 | 0.0019 | 3.101 | 368 | 0.002 | 0.161 | 0.0022 | 0.0098 | negl. | ** |
| 0.85 | 369 | 0.517 | 1.02 | 0.002 | 2.612 | 368 | 0.009 | 0.136 | 0.0013 | 0.0091 | negl. | ** |
| 0.9 | 369 | 0.5167 | 0.96 | 0.002 | 2.46 | 368 | 0.014 | 0.128 | 0.001 | 0.0088 | negl. | * |
| 0.95 | 369 | 0.5182 | 1.25 | 0.002 | 3.123 | 368 | 0.002 | 0.162 | 0.0024 | 0.0104 | negl. | ** |

**Table B.4** T-CPS Descriptive Metrics for Granite 3.2 8B

| Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|
| 0.5 | 0.558285774 | 0.439923113 | 0.1276133 | Minimal change |
| 0.55 | 0.559447145 | 0.648862756 | 0.1251651 | Small improvement |
| 0.6 | 0.558784596 | 0.529665128 | 0.1246806 | Small improvement |
| 0.65 | 0.561117794 | 0.949425401 | 0.1204028 | Small improvement |
| 0.7 | 0.562219649 | 1.147657751 | 0.1287082 | Small improvement |
| 0.75 | 0.561976552 | 1.103922604 | 0.1239542 | Small improvement |
| 0.8 | 0.562517765 | 1.201291071 | 0.1220778 | Small improvement |
| 0.85 | 0.561554319 | 1.027959704 | 0.1235255 | Small improvement |
| 0.9 | 0.560802013 | 0.892614034 | 0.1299823 | Small improvement |
| 0.95 | 0.562812965 | 1.254399823 | 0.1239809 | Small improvement |

**Table B.5** Statistical Analysis Results for Llama 3.1 8B (Phase III, Agriculture Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.5001. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 369 | 0.4974 | −0.55 | 0.0044 | −0.632 | 368 | 0.528 | −0.033 | −0.0113 | 0.0058 | negl. | ns |
| 0.55 | 369 | 0.5052 | 1.01 | 0.0039 | 1.285 | 368 | 0.2 | 0.067 | −0.0027 | 0.0128 | negl. | ns |
| 0.6 | 369 | 0.4757 | −4.88 | 0.0064 | −3.817 | 368 | <0.001 | −0.198 | −0.0369 | −0.0118 | negl. | *** |
| 0.65 | 369 | 0.4986 | −0.29 | 0.0037 | −0.398 | 368 | 0.691 | −0.021 | −0.0087 | 0.0058 | negl. | ns |
| 0.7 | 369 | 0.5065 | 1.33 | 0.0047 | 1.412 | 368 | 0.159 | 0.074 | −0.0026 | 0.0159 | negl. | ns |
| 0.75 | 369 | 0.5016 | 0.3 | 0.004 | 0.375 | 368 | 0.708 | 0.02 | −0.0064 | 0.0094 | negl. | ns |
| 0.8 | 369 | 0.5045 | 0.87 | 0.0046 | 0.944 | 368 | 0.346 | 0.049 | −0.0047 | 0.0135 | negl. | ns |
| 0.85 | 369 | 0.5034 | 0.65 | 0.0046 | 0.712 | 368 | 0.477 | 0.037 | −0.0057 | 0.0122 | negl. | ns |
| 0.9 | 369 | 0.508 | 1.58 | 0.0039 | 2.051 | 368 | 0.041 | 0.107 | 0.0003 | 0.0155 | negl. | * |
| 0.95 | 369 | 0.4742 | −5.18 | 0.0056 | −4.648 | 368 | <0.001 | −0.242 | −0.0369 | −0.0149 | small | *** |

**Table B.6** T-CPS Descriptive Metrics for Llama 3.1 8B

| Model | Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|---|
| Llama 3.1 8B | 0.5 | 0.538902832 | -0.594313846 | 0.1472357 | Minimal change |
| Llama 3.1 8B | 0.55 | 0.548420913 | 1.161385588 | 0.1285758 | Small improvement |
| Llama 3.1 8B | 0.6 | 0.509539316 | -6.010689916 | 0.2332831 | Minimal change |
| Llama 3.1 8B | 0.65 | 0.540624943 | -0.276654371 | 0.1383009 | Minimal change |
| Llama 3.1 8B | 0.7 | 0.548977378 | 1.264030752 | 0.1417794 | Small improvement |
| Llama 3.1 8B | 0.75 | 0.543137024 | 0.186722539 | 0.1500943 | Minimal change |
| Llama 3.1 8B | 0.8 | 0.545679803 | 0.65576217 | 0.1589308 | Small improvement |
| Llama 3.1 8B | 0.85 | 0.544033734 | 0.352129152 | 0.1546914 | Minimal change |
| Llama 3.1 8B | 0.9 | 0.550146432 | 1.479673741 | 0.1479137 | Small improvement |
| Llama 3.1 8B | 0.95 | 0.510584003 | -5.81798773 | 0.1939108 | Minimal change |

**Table B.7** Statistical Analysis Results for DeepSeek R1 8B (Phase III, Agriculture Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.4514. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 369 | 0.4515 | 0.04 | 0.0013 | 0.126 | 368 | 0.9 | 0.007 | −0.0024 | 0.0027 | negl. | ns |
| 0.55 | 369 | 0.453 | 0.36 | 0.0012 | 1.307 | 368 | 0.192 | 0.068 | −0.0008 | 0.004 | negl. | ns |
| 0.6 | 369 | 0.4523 | 0.19 | 0.0015 | 0.572 | 368 | 0.568 | 0.03 | −0.0021 | 0.0039 | negl. | ns |
| 0.65 | 369 | 0.4548 | 0.77 | 0.0013 | 2.661 | 368 | 0.008 | 0.138 | 0.0009 | 0.006 | negl. | ** |
| 0.7 | 369 | 0.4525 | 0.26 | 0.0014 | 0.859 | 368 | 0.391 | 0.045 | −0.0015 | 0.0039 | negl. | ns |
| 0.75 | 369 | 0.4538 | 0.54 | 0.0013 | 1.823 | 368 | 0.069 | 0.095 | −0.0002 | 0.005 | negl. | ns |
| 0.8 | 369 | 0.4531 | 0.38 | 0.0014 | 1.267 | 368 | 0.206 | 0.066 | −0.0009 | 0.0044 | negl. | ns |
| 0.85 | 369 | 0.4515 | 0.02 | 0.0015 | 0.07 | 368 | 0.944 | 0.004 | −0.0028 | 0.003 | negl. | ns |
| 0.9 | 369 | 0.4559 | 1.01 | 0.002 | 2.301 | 368 | 0.022 | 0.119 | 0.0007 | 0.0085 | negl. | * |
| 0.95 | 369 | 0.455 | 0.8 | 0.0024 | 1.482 | 368 | 0.139 | 0.077 | −0.0012 | 0.0084 | negl. | ns |

**Table B.8** T-CPS Descriptive Metrics for DeepSeek 8B

| Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|
| 0.5 | 0.492188101 | -0.001528152 | 0.0902093 | Minimal change |
| 0.55 | 0.494035356 | 0.373781011 | 0.0851082 | Minimal change |
| 0.6 | 0.492870203 | 0.137055412 | 0.0926327 | Minimal change |
| 0.65 | 0.496089412 | 0.791106121 | 0.0847496 | Small improvement |
| 0.7 | 0.493299299 | 0.224235452 | 0.0902052 | Minimal change |
| 0.75 | 0.494706442 | 0.510126432 | 0.0892088 | Small improvement |
| 0.8 | 0.49403267 | 0.373235233 | 0.0872966 | Minimal change |
| 0.85 | 0.492121897 | -0.01497885 | 0.090172 | Minimal change |
| 0.9 | 0.496019969 | 0.776997311 | 0.1075279 | Small improvement |
| 0.95 | 0.495849291 | 0.742320504 | 0.0931026 | Small improvement |

**Table B.9.** Phase III Spearman correlation matrix (rounded to three decimals)

| | METEOR | Rouge-2.f | Rouge-l.f | Bert-Score.f1 | B-RT.average | F1 score | B-RT.fluency | Laplace Perplexity | Lidstone Perplexity |
|---|---|---|---|---|---|---|---|---|---|
| METEOR | 1 | 0.23 | 0.011 | 0.72 | 0.534 | 0.813 | 0.544 | 0.372 | 0.207 |
| Rouge-2.f | 0.23 | 1 | 0.768 | 0.123 | -0.128 | -0.067 | -0.128 | -0.173 | -0.236 |
| Rouge-l.f | 0.011 | 0.768 | 1 | -0.05 | -0.241 | -0.301 | -0.259 | -0.265 | -0.235 |
| Bert-Score.f1 | 0.72 | 0.123 | -0.05 | 1 | 0.741 | 0.793 | 0.728 | 0.491 | 0.347 |
| B-RT.average | 0.534 | -0.128 | -0.241 | 0.741 | 1 | 0.713 | 0.95 | 0.486 | 0.357 |
| F1 score | 0.813 | -0.067 | -0.301 | 0.793 | 0.713 | 1 | 0.704 | 0.561 | 0.392 |
| B-RT.fluency | 0.544 | -0.128 | -0.259 | 0.728 | 0.95 | 0.704 | 1 | 0.378 | 0.246 |
| Laplace Perplexity | 0.372 | -0.173 | -0.265 | 0.491 | 0.486 | 0.561 | 0.378 | 1 | 0.828 |
| Lidstone Perplexity | 0.207 | -0.236 | -0.235 | 0.347 | 0.357 | 0.392 | 0.246 | 0.828 | 1 |

*Spearman correlations were computed on pooled per-question rows across thresholds and baseline; values are rounded to three decimals.*

**Reproducibility notes (Phase III correlation)**

Per-question Phase III results were pooled across all thresholds 0.50–0.95 (step 0.05) and the baseline configuration (threshold_0.00) for all four models (Mistral, Granite, Llama, DeepSeek). Each per-threshold export file was loaded and the trailing aggregate row (id = 'Average') was removed when present. The pooled dataset contained N = 16,324 per-question rows (44 files: 4 models × 11 configurations). Spearman rank correlations were computed pairwise over the Phase III component-metric panel: METEOR, ROUGE-2.f, ROUGE-L.f, BERTScore.f1, B-RT.average, token-overlap F1, B-RT.fluency, Laplace Perplexity, and Lidstone Perplexity. No threshold-level averaging (N = 11) was used for correlation estimation. The resulting 9×9 correlation matrix is reported below (rounded to three decimals).

# APPENDIX C. Phase IV Supplementary Tables

*P-values are from two-tailed paired t-tests against baseline at the per-question level (uncorrected). Values below 0.001 are reported as p<0.001.*

**Table C.1.** Statistical Analysis Results for Mistral 7B v0.3 (Phase IV, Biodiversity Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.4334. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 426 | 0.4605 | 6.27 | 0.0063 | 4.343 | 425 | <0.001 | 0.21 | 0.0149 | 0.0395 | small | *** |
| 0.55 | 426 | 0.4547 | 4.92 | 0.0067 | 3.189 | 425 | 0.002 | 0.154 | 0.0082 | 0.0345 | negl. | ** |
| 0.6 | 426 | 0.447 | 3.14 | 0.0067 | 2.018 | 425 | 0.044 | 0.098 | 0.0004 | 0.0269 | negl. | * |
| 0.65 | 426 | 0.4764 | 9.94 | 0.0068 | 6.352 | 425 | <0.001 | 0.308 | 0.0297 | 0.0564 | small | *** |
| 0.7 | 426 | 0.4747 | 9.53 | 0.0071 | 5.791 | 425 | <0.001 | 0.281 | 0.0273 | 0.0553 | small | *** |
| 0.75 | 426 | 0.4625 | 6.72 | 0.0067 | 4.34 | 425 | <0.001 | 0.21 | 0.0159 | 0.0423 | small | *** |
| 0.8 | 426 | 0.4911 | 13.32 | 0.0071 | 8.095 | 425 | <0.001 | 0.392 | 0.0437 | 0.0717 | small | *** |
| 0.85 | 426 | 0.4593 | 5.99 | 0.0068 | 3.791 | 425 | <0.001 | 0.184 | 0.0125 | 0.0394 | negl. | *** |
| 0.9 | 426 | 0.4662 | 7.57 | 0.0068 | 4.798 | 425 | <0.001 | 0.232 | 0.0194 | 0.0463 | small | *** |
| 0.95 | 426 | 0.4275 | -1.35 | 0.0065 | -0.899 | 425 | 0.369 | -0.044 | -0.0186 | 0.0069 | negl. | ns |

**Table C.2.** T-CPS Descriptive Metrics for Mistral 7B

| Model | Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|---|
| Mistral 7B | 0.5 | 0.491239221 | 6.802074408 | 0.2599705 | Large improvement |
| Mistral 7B | 0.55 | 0.485594086 | 5.574745476 | 0.2511963 | Large improvement |
| Mistral 7B | 0.6 | 0.478879033 | 4.114801877 | 0.2279347 | Large improvement |
| Mistral 7B | 0.65 | 0.510229438 | 10.93080537 | 0.2334984 | Large improvement |
| Mistral 7B | 0.7 | 0.507839323 | 10.41116201 | 0.2404051 | Large improvement |
| Mistral 7B | 0.75 | 0.495573821 | 7.744475408 | 0.2281887 | Large improvement |
| Mistral 7B | 0.8 | 0.525400296 | 14.22915587 | 0.2417808 | Large improvement |
| Mistral 7B | 0.85 | 0.49293772 | 7.171351324 | 0.2169546 | Large improvement |
| Mistral 7B | 0.9 | 0.498990605 | 8.487330541 | 0.2363135 | Large improvement |
| Mistral 7B | 0.95 | 0.456707505 | -0.705589188 | 0.2464171 | Minimal change |

**Table C.3.** Statistical Analysis Results for Granite 3.2 8B (Phase IV, Biodiversity Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.4183. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 426 | 0.4238 | 1.31 | 0.006 | 0.919 | 425 | 0.358 | 0.045 | -0.0063 | 0.0172 | negl. | ns |
| 0.55 | 426 | 0.4413 | 5.51 | 0.0059 | 3.928 | 425 | <0.001 | 0.19 | 0.0115 | 0.0346 | negl. | *** |
| 0.6 | 426 | 0.4189 | 0.15 | 0.0059 | 0.107 | 425 | 0.915 | 0.005 | -0.011 | 0.0123 | negl. | ns |
| 0.65 | 426 | 0.4254 | 1.71 | 0.0059 | 1.216 | 425 | 0.225 | 0.059 | -0.0044 | 0.0187 | negl. | ns |
| 0.7 | 426 | 0.4155 | -0.67 | 0.0061 | -0.46 | 425 | 0.646 | -0.022 | -0.0148 | 0.0092 | negl. | ns |
| 0.75 | 426 | 0.4283 | 2.39 | 0.0059 | 1.71 | 425 | 0.088 | 0.083 | -0.0015 | 0.0215 | negl. | ns |
| 0.8 | 426 | 0.4473 | 6.95 | 0.0068 | 4.274 | 425 | <0.001 | 0.207 | 0.0157 | 0.0424 | small | *** |
| 0.85 | 426 | 0.4144 | -0.92 | 0.0055 | -0.698 | 425 | 0.485 | -0.034 | -0.0147 | 0.007 | negl. | ns |
| 0.9 | 426 | 0.4295 | 2.69 | 0.0064 | 1.751 | 425 | 0.081 | 0.085 | -0.0014 | 0.0239 | negl. | ns |
| 0.95 | 426 | 0.4422 | 5.73 | 0.0058 | 4.158 | 425 | <0.001 | 0.201 | 0.0126 | 0.0353 | small | *** |

**Table C.4.** Statistical Analysis Results for Granite 3.2 8B (Phase IV, Biodiversity Domain).

| Model | Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|---|
| Granite 3.2 8B | 0.5 | 0.452633114 | 1.458538168 | 0.2466563 | Small improvement |
| Granite 3.2 8B | 0.55 | 0.472294833 | 5.865748287 | 0.2348773 | Large improvement |
| Granite 3.2 8B | 0.6 | 0.447277365 | 0.25803729 | 0.2485925 | Minimal change |
| Granite 3.2 8B | 0.65 | 0.454157722 | 1.800281722 | 0.2502454 | Moderate improvement |
| Granite 3.2 8B | 0.7 | 0.442922901 | -0.718023891 | 0.258421 | Minimal change |
| Granite 3.2 8B | 0.75 | 0.457170267 | 2.475549315 | 0.2512337 | Moderate improvement |
| Granite 3.2 8B | 0.8 | 0.478482352 | 7.252691927 | 0.2394132 | Large improvement |
| Granite 3.2 8B | 0.85 | 0.442670351 | -0.774633449 | 0.2454523 | Minimal change |
| Granite 3.2 8B | 0.9 | 0.458412399 | 2.753975386 | 0.2526316 | Moderate improvement |
| Granite 3.2 8B | 0.95 | 0.471980265 | 5.795237382 | 0.2542331 | Large improvement |

**Table C.5.** Statistical Analysis Results for Llama 3.1 8B (Phase IV, Biodiversity Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.4618. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 426 | 0.4407 | -4.57 | 0.0077 | -2.732 | 425 | 0.007 | -0.132 | -0.0363 | -0.0059 | negl. | ** |
| 0.55 | 426 | 0.4386 | -5.02 | 0.0045 | -5.144 | 425 | <0.001 | -0.249 | -0.032 | -0.0143 | small | *** |
| 0.6 | 426 | 0.4366 | -5.46 | 0.0045 | -5.555 | 425 | <0.001 | -0.269 | -0.0342 | -0.0163 | small | *** |
| 0.65 | 426 | 0.4557 | -1.31 | 0.0042 | -1.428 | 425 | 0.154 | -0.069 | -0.0144 | 0.0023 | negl. | ns |
| 0.7 | 426 | 0.4606 | -0.25 | 0.0045 | -0.258 | 425 | 0.796 | -0.013 | -0.01 | 0.0077 | negl. | ns |
| 0.75 | 426 | 0.4535 | -1.79 | 0.0044 | -1.869 | 425 | 0.062 | -0.091 | -0.017 | 0.0004 | negl. | ns |
| 0.8 | 426 | 0.4567 | -1.11 | 0.0045 | -1.147 | 425 | 0.252 | -0.056 | -0.0139 | 0.0036 | negl. | ns |
| 0.85 | 426 | 0.4713 | 2.06 | 0.0046 | 2.072 | 425 | 0.039 | 0.1 | 0.0005 | 0.0185 | negl. | * |
| 0.9 | 426 | 0.4485 | -2.88 | 0.0042 | -3.136 | 425 | 0.002 | -0.152 | -0.0216 | -0.005 | negl. | ** |
| 0.95 | 426 | 0.4434 | -3.98 | 0.0046 | -4.031 | 425 | <0.001 | -0.195 | -0.0273 | -0.0094 | negl. | *** |

**Table C.6.** T-CPS Descriptive Metrics for Llama 3.1 8B

| Model | Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|---|
| Llama 3.1 8B | 0.5 | 0.469024632 | -4.853076496 | 0.2724771 | Minimal change |
| Llama 3.1 8B | 0.55 | 0.466490745 | -5.367103965 | 0.2768902 | Minimal change |
| Llama 3.1 8B | 0.6 | 0.463459338 | -5.982058909 | 0.2886381 | Minimal change |
| Llama 3.1 8B | 0.65 | 0.487374612 | -1.130576489 | 0.2415567 | Minimal change |
| Llama 3.1 8B | 0.7 | 0.49279859 | -0.030261516 | 0.239508 | Minimal change |
| Llama 3.1 8B | 0.75 | 0.484268668 | -1.760652187 | 0.2520119 | Minimal change |
| Llama 3.1 8B | 0.8 | 0.488431012 | -0.916273921 | 0.2411207 | Minimal change |
| Llama 3.1 8B | 0.85 | 0.504249178 | 2.292619025 | 0.2400511 | Moderate improvement |
| Llama 3.1 8B | 0.9 | 0.479515056 | -2.724975871 | 0.2427271 | Minimal change |
| Llama 3.1 8B | 0.95 | 0.473605491 | -3.923797566 | 0.249452 | Minimal change |

**Table C.7.** Statistical Analysis Results for DeepSeek 8B (Phase IV, Biodiversity Domain).

*Full statistical results including standard error, degrees of freedom, and 95% confidence intervals. Baseline Mean CPS = 0.4697. Significance: ns = not significant, * p < 0.05, ** p < 0.01, *** p < 0.001. Effect sizes: negl. = negligible (d < 0.2), small (0.2 ≤ d < 0.5).*

| Thresh. | N | Mean CPS | CPS Impr. % | SE | t_statistic | df | p | Cohens_d | CI Low | CI Up | Effect | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 426 | 0.4652 | -0.94 | 0.0033 | -1.343 | 425 | 0.18 | -0.065 | -0.0109 | 0.0021 | negl. | ns |
| 0.55 | 426 | 0.5094 | 8.45 | 0.0044 | 8.975 | 425 | <0.001 | 0.435 | 0.031 | 0.0484 | small | *** |
| 0.6 | 426 | 0.4906 | 4.45 | 0.0048 | 4.316 | 425 | <0.001 | 0.209 | 0.0114 | 0.0304 | small | *** |
| 0.65 | 426 | 0.4711 | 0.3 | 0.0046 | 0.301 | 425 | 0.764 | 0.015 | -0.0077 | 0.0105 | negl. | ns |
| 0.7 | 426 | 0.4775 | 1.66 | 0.0049 | 1.604 | 425 | 0.109 | 0.078 | -0.0018 | 0.0174 | negl. | ns |
| 0.75 | 426 | 0.4666 | -0.65 | 0.0048 | -0.627 | 425 | 0.531 | -0.03 | -0.0126 | 0.0065 | negl. | ns |
| 0.8 | 426 | 0.4609 | -1.88 | 0.0048 | -1.852 | 425 | 0.065 | -0.09 | -0.0182 | 0.0005 | negl. | ns |
| 0.85 | 426 | 0.4661 | -0.75 | 0.0045 | -0.785 | 425 | 0.433 | -0.038 | -0.0124 | 0.0053 | negl. | ns |
| 0.9 | 426 | 0.4561 | -2.88 | 0.0046 | -2.951 | 425 | 0.003 | -0.143 | -0.0226 | -0.0045 | negl. | ** |
| 0.95 | 426 | 0.4493 | -4.33 | 0.0046 | -4.425 | 425 | <0.001 | -0.214 | -0.0294 | -0.0113 | small | *** |

**Table C.8.** T-CPS Descriptive Metrics for DeepSeek 8B

| Model | Threshold | Test_T-CPS | T-CPS_Improvement_% | CV | Interpretation |
|---|---|---|---|---|---|
| DeepSeek 8B | 0.5 | 0.504290427 | -0.803487748 | 0.1395878 | Minimal change |
| DeepSeek 8B | 0.55 | 0.552873232 | 8.752999236 | 0.1291463 | Large improvement |
| DeepSeek 8B | 0.6 | 0.53062035 | 4.375743186 | 0.1579591 | Large improvement |
| DeepSeek 8B | 0.65 | 0.510544163 | 0.426654189 | 0.1408425 | Minimal change |
| DeepSeek 8B | 0.7 | 0.516483305 | 1.594914032 | 0.157077 | Moderate improvement |
| DeepSeek 8B | 0.75 | 0.504312225 | -0.799199893 | 0.1637613 | Minimal change |
| DeepSeek 8B | 0.8 | 0.498412106 | -1.959783522 | 0.158142 | Minimal change |
| DeepSeek 8B | 0.85 | 0.50474855 | -0.713372577 | 0.1480727 | Minimal change |
| DeepSeek 8B | 0.9 | 0.493476197 | -2.930702089 | 0.1548192 | Minimal change |
| DeepSeek 8B | 0.95 | 0.48544806 | -4.509877821 | 0.1656804 | Minimal change |

**Table C.9.** Phase III Spearman correlation matrix (rounded to three decimals)

| | METEOR | Rouge-2.f | Rouge-l.f | Bert-Score.f1 | B-RT.average | F1 score | B-RT.fluency | Laplace Perplexity | Lidstone Perplexity |
|---|---|---|---|---|---|---|---|---|---|
| METEOR | 1 | 0.843 | 0.829 | 0.709 | 0.438 | 0.788 | 0.454 | -0.181 | -0.3 |
| Rouge-2.f | 0.843 | 1 | 0.881 | 0.729 | 0.476 | 0.815 | 0.479 | -0.162 | -0.326 |
| Rouge-l.f | 0.829 | 0.881 | 1 | 0.803 | 0.598 | 0.925 | 0.597 | -0.037 | -0.163 |
| Bert-Score.f1 | 0.709 | 0.729 | 0.803 | 1 | 0.74 | 0.802 | 0.739 | -0.087 | -0.201 |
| B-RT.average | 0.438 | 0.476 | 0.598 | 0.74 | 1 | 0.682 | 0.963 | 0.049 | -0.031 |
| F1 score | 0.788 | 0.815 | 0.925 | 0.802 | 0.682 | 1 | 0.68 | 0.062 | -0.062 |
| B-RT.fluency | 0.454 | 0.479 | 0.597 | 0.739 | 0.963 | 0.68 | 1 | 0.03 | -0.046 |
| Laplace Perplexity | -0.181 | -0.162 | -0.037 | -0.087 | 0.049 | 0.062 | 0.03 | 1 | 0.765 |
| Lidstone Perplexity | -0.3 | -0.326 | -0.163 | -0.201 | -0.031 | -0.062 | -0.046 | 0.765 | 1 |

***Note:*** *Spearman correlations were computed on pooled per-question rows across thresholds and baseline; values are rounded to three decimals.*

**Reproducibility notes (Phase IV correlation)**

Per-question Phase IV results were pooled across all thresholds 0.50–0.95 (step 0.05) and the baseline configuration (threshold_0.00) for all four models (Mistral, Granite, Llama, DeepSeek). Each per-threshold export file was loaded and the trailing aggregate row (id = 'Average') was removed when present. The pooled dataset contained N = 18,699 per-question rows (44 files: 4 models × 11 configurations). Spearman rank correlations were computed pairwise over the Phase IV component-metric panel: METEOR, ROUGE-2.f, ROUGE-L.f, BERTScore.f1, B-RT.average, token-overlap F1, B-RT.fluency, Laplace Perplexity, and Lidstone Perplexity. No threshold-level averaging (N = 11) was used for correlation estimation. The resulting 9×9 correlation matrix is reported below (rounded to three decimals).

# BIBLIOGRAPHY

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[3] Z. Ji, N. Qiu, S. Xu, D. Young, F. Tao, L. Lyu, C. Chen, C. Gu, R. Li, L. Yang, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, Dec. 2023, doi: 10.1145/3571730.

[4] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3784–3803, Nov. 2021, doi: 10.18653/v1/2021.findings-emnlp.320.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459–9474, 2020.

[6] N. Rossi, G. M. Gupta, S. Agarwal, S. Srinivasan, J. Liu, S. Han, and Y. Gao, "Relevance filtering for embedding-based retrieval," in *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*, 2024, doi: 10.1145/3627673.3680095.

[7] A. Pal, L. K. Umapathi, and M. Sankarasubbu, "Med-HALT: Medical domain hallucination test for large language models," in *Proc. 27th Conf. on Computational Natural Language Learning (CoNLL)*, pp. 314–334, 2023, doi: 10.18653/v1/2023.conll-1.21.

[8] J. Cui, Z. Li, Y. Yan, B. Chen, and L. Yuan, "ChatLaw: Open-source legal large language model with integrated external knowledge bases," *arXiv preprint arXiv:2306.16092*, Jun. 2023, doi: 10.48550/arXiv.2306.16092.

[9] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, S. Barezi, P. Pascual, H. Li, R. Shick, S. Joty, B. Shin, and P. Fung, "A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity,"

in *Proc. Int. Joint Conf. on Natural Language Processing and the Asia-Pacific Chapter of the ACL (IJCNLP-AACL)*, 2023.

[10] Y. Gao, Y. Xiong, X. Wang, J. Wang, Z. Jiang, H. Li, Y. Wen, K. Jiang, N. Meng, L. Shao, and P. Sethi, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, Dec. 2023, doi: 10.48550/arXiv.2312.10997.

[11] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[12] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature*, vol. 533, no. 7604, pp. 452–454, May 2016, doi: 10.1038/533452a.

[13] R. Pradeep, N. Thakur, S. Upadhyay, D. Campos, N. Craswell, and J. Lin, "Initial nugget evaluation results for the TREC 2024 RAG track with the AutoNuggetizer framework," *arXiv preprint arXiv:2411.09607*, Nov. 2024, doi: 10.48550/arXiv.2411.09607.

[14] J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking large language models in retrieval-augmented generation," in *Proc. AAAI Conf. on Artificial Intelligence (AAAI)*, vol. 38, no. 16, pp. 17754–17762, 2024.

[15] M. Dimitrova, I. Popchev, and I. Radeva, "PaSSER: A platform for evaluating LLMs in RAG," in *2025 IEEE BdKCSE*, 2025, p. 7, doi: 10.1109/BdKCSE67969.2025.11300500.

[16] I. Radeva, I. Popchev, and M. Dimitrova, "Similarity thresholds in retrieval-augmented generation," in *2024 IEEE 12th Int. Conf. on Intelligent Systems (IS)*, Aug. 2024, pp. 1–7, doi: 10.1109/IS61756.2024.10705214.

[17] I. Radeva, I. Popchev, L. Doukovska, and M. Dimitrova, "Web application for retrieval-augmented generation: Implementation and testing," *Electronics*, vol. 13, no. 7, p. 1361, Apr. 2024, doi: 10.3390/electronics13071361.

[18] I. Radeva, I. Popchev, L. Doukovska, and M. Dimitrova, "Multi-agent coordination strategies vs. retrieval-augmented generation in LLMs: A comparative evaluation," *Electronics*, vol. 14, no. 24, p. 4883, Jan. 2025, doi: 10.3390/electronics14244883.

[19] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. draft. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/ (accessed Jan. 20, 2026).

[20] M. Dimitrova, "Retrieval-augmented generation (RAG): Advances and challenges," *Problems of Engineering Cybernetics and Robotics (PECR)*, vol. 83, Jul. 2025, doi: 10.7546/PECR.83.25.03.

[21] V. Bush, "As we may think," *The Atlantic*, Jul. 1945. [Online]. Available: https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/ (accessed Dec. 10, 2024).

[22] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of Research and Development*, vol. 1, no. 4, pp. 309–317, Oct. 1957, doi: 10.1147/rd.14.0309.

[23] "Keyword-in-context index for technical literature (KWIC index)," *HathiTrust*. [Online]. Available: https://hdl.handle.net/2027/mdp.39015005511467?urlappend=%3Bseq=16 (accessed Mar. 18, 2025).

[24] J. Rees and A. Kent, "Mechanized searching experiments using the WRU searching selector," *American Documentation*, vol. 9, no. 4, pp. 277–303, 1958, doi: 10.1002/asi.5090090404.

[25] C. N. Mooers, "Zatocoding applied to mechanical organization of knowledge," *American Documentation*, vol. 2, no. 1, pp. 20–32, 1951, doi: 10.1002/asi.5090020107.

[26] B. F. Green, A. K. Wolf, C. Chomsky, and K. Laughery, "Baseball: An automatic question-answerer," in *Proc. Western Joint IRE-AIEE-ACM Computer Conf. (Western)*, Los Angeles, CA, USA, 1961, p. 219, doi: 10.1145/1460690.1460714.

[27] C. W. Cleverdon and M. Keen, "Aslib Cranfield research project: Factors determining the performance of indexing systems; Volume 2, Test results," 1966. [Online]. Available: http://hdl.handle.net/1826/863

[28] K. Sparck Jones and K. Reeves, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, Jan. 1972, doi: 10.1108/eb026526.

[29] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975, doi: 10.1145/361219.361220.

[30] W. A. Woods, R. M. Kaplan, and B. Nash-Webber, "The Lunar Sciences Natural Language Information System: Final report," BBN Technologies, Cambridge, MA, USA, Tech. Rep., Jun. 1972.

[31] W. A. Woods, "Progress in natural language understanding: An application to lunar geology," in *Proc. Nat. Comput. Conf. Expo. (AFIPS '73)*, Jun. 1973, pp. 441–450, doi: 10.1145/1499586.1499695.

[32] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR '98)*, Aug. 1998, pp. 275–281, doi: 10.1145/290941.291008.

[33] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, "Using latent semantic analysis to improve access to textual information," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI '88)*, May 1988, pp. 281–285, doi: 10.1145/57167.57214.

[34] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, C. Schlaefer, and C. Welty, "Building Watson: An overview of the DeepQA project," *AI Mag.*, vol. 31, no. 3, pp. 59–79, 2010, doi: 10.1609/aimag.v31i3.2303.

[35] D. A. Ferrucci, "Introduction to 'This is Watson,'" *IBM J. Res. Dev.*, vol. 56, no. 3.4, pp. 1:1–1:15, May 2012, doi: 10.1147/JRD.2012.2184356.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, Jan. 2013, doi: 10.48550/arXiv.1301.3781.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[38] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," in *Proc. EMNLP*, Nov. 2020, pp. 6769–6781, doi: 10.18653/v1/2020.emnlp-main.550.

[39] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, Tech. Rep., 2019.

[40] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, Mar. 2023, doi: 10.48550/arXiv.2303.08774.

[41] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, Mar. 2025, doi: 10.1145/3703155.

[42] J. Huang, X. Chen, S. Mishra, H. S. Liao, J. J. Chung, H. G. Song, and D. Zhou, "Large language models cannot self-correct reasoning yet," *arXiv preprint arXiv:2310.01798*, Oct. 2023, doi: 10.48550/arXiv.2310.01798.

[43] A. Shrivastava and P. Li, "Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS)," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014.

[44] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Sep. 2021, doi: 10.1109/TBDATA.2019.2921572.

[45] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, Apr. 2020, doi: 10.1109/TPAMI.2018.2889473.

[46] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2020, doi: 10.18653/v1/2020.acl-main.703.

[47] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014.

[48] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, J. Polosukhin, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M. Kelcey, M. W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguist.*, vol. 7, pp. 452–466, 2019, doi: 10.1162/tacl_a_00276.

[49] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2017, pp. 1601–1611, doi: 10.18653/v1/P17-1147.

[50] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A large-scale dataset for fact extraction and VERification," in *Proc. NAACL-HLT*, 2018, pp. 809–819, doi: 10.18653/v1/N18-1074.

[51] P. Bajaj, D. Campos, N. Craswell, L. Deng, C. Majumder, X. Qu, B. de Rossi, A. Rodriguez, B. Bhaskar, R. Lin, S. Sayyaparaju, and J. Shao, "MS MARCO: A human generated MAchine reading COmprehension dataset," *arXiv preprint arXiv:1611.09268*, Nov. 2016, doi: 10.48550/arXiv.1611.09268.

[52] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. EACL*, 2021.

[53] S. Hofstätter, J. Chen, K. Raman, and H. Zamani, "FiD-Light: Efficient and effective retrieval-augmented text generation," *arXiv preprint arXiv:2209.14290*, Sep. 2022, doi: 10.48550/arXiv.2209.14290.

[54] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. S. Majid, S. Edunov, V. Karpukhin, G. T. Usunier, H. Schick, P. Akter-Syed, D. Hazen, G. Szilard, A. Fan, M. Lewis, S. Riedel, and S. J. Pan, "KILT: A

benchmark for knowledge intensive language tasks," in *Proc. NAACL-HLT*, 2021, doi: 10.18653/v1/2021.naacl-main.200.

[55] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "LightRAG: Simple and fast retrieval-augmented generation," *arXiv preprint arXiv:2410.05779*, Oct. 2024, doi: 10.48550/arXiv.2410.05779.

[56] D. Edge, H. Trinh, B. Cheng, J. Bradley, N. Chao, A. Mody, S. Truitt, and J. Larson, "From local to global: A graph RAG approach to query-focused summarization," *arXiv preprint arXiv:2404.16130*, Apr. 2024, doi: 10.48550/arXiv.2404.16130.

[57] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: Guaranteeing well-connected communities," *Sci. Rep.*, vol. 9, Art. no. 5233, Mar. 2019, doi: 10.1038/s41598-019-41695-z.

[58] T. Yu, S. Zhang, and Y. Feng, "Auto-RAG: Autonomous retrieval-augmented generation for large language models," *arXiv preprint arXiv:2411.19443*, Nov. 2024, doi: 10.48550/arXiv.2411.19443.

[59] Z. Wang, J. Cho, S. S. Kim, S. J. Hwang, S. Lee, and J. G. Park, "Speculative RAG: Enhancing retrieval augmented generation through drafting," *arXiv preprint arXiv:2407.08223*, Jul. 2024, doi: 10.48550/arXiv.2407.08223.

[60] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," *arXiv preprint arXiv:2310.11511*, Oct. 2023, doi: 10.48550/arXiv.2310.11511.

[61] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective retrieval augmented generation," *arXiv preprint arXiv:2401.15884*, Jan. 2024, doi: 10.48550/arXiv.2401.15884.

[62] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity," in *Proc. NAACL-HLT (Long Papers)*, 2024, pp. 7036–7050.

[63] Z. Jiang, F. F. Xu, L. Gao, J. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, C. Callison-Burch, and G. Neubig, "Active retrieval augmented generation," in *Proc. EMNLP*, 2023, pp. 7969–7992, doi: 10.18653/v1/2023.emnlp-main.495.

[64] J. Huang, W. Ping, P. Xu, M. Shoeybi, K. C.-C. Chang, and B. Catanzaro, "RAVEN: In-context learning with retrieval-augmented encoder-decoder language models," *arXiv preprint arXiv:2308.07922*, Aug. 2023, doi: 10.48550/arXiv.2308.07922.

[65] P. Mandikal and R. Mooney, "Sparse meets dense: A hybrid approach to enhance scientific document retrieval," *arXiv preprint arXiv:2401.04055*, Jan. 2024, doi: 10.48550/arXiv.2401.04055.

[66] W3C, "SPARQL 1.1 Query Language," W3C Recommendation. Accessed: Mar. 23, 2025. [Online]. Available: https://www.w3.org/TR/sparql11-query/

[67] S. Jeong, K. Kim, J. Baek, and S. J. Hwang, "VideoRAG: Retrieval-augmented generation over video corpus," *arXiv preprint arXiv:2501.05874*, Jan. 2025, doi: 10.48550/arXiv.2501.05874.

[68] W. Chen, H. Hu, X. Chen, P. Verga, and W. W. Cohen, "MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text," *arXiv preprint arXiv:2210.02928*, Oct. 2022, doi: 10.48550/arXiv.2210.02928.

[69] J. Lála, O. O'Donoghue, A. Shtedritski, S. Cox, S. G. Rodriques, and A. D. White, "PaperQA: Retrieval-augmented generative agent for scientific research," *arXiv preprint arXiv:2312.07559*, Dec. 2023, doi: 10.48550/arXiv.2312.07559.

[70] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, "PubMedQA: A dataset for biomedical research question answering," in *Proc. EMNLP-IJCNLP*, 2019, pp. 2567–2577, doi: 10.18653/v1/D19-1259.

[71] FutureHouse, "Future-House/LitQA," GitHub repository. Accessed: Jan. 07, 2026. [Online]. Available: https://github.com/Future-House/LitQA

[72] Y. Hoshi, D. Miyawaki, K. Suzuki, K. Sakaida, S. Nakayama, and Y. Taguchi, "RaLLe: A framework for developing and evaluating retrieval-augmented large language models," *arXiv preprint arXiv:2308.10633*, Aug. 2023, doi: 10.48550/arXiv.2308.10633.

[73] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, and J. E. Gonzalez, "RAFT: Adapting language model to domain specific RAG," *arXiv preprint arXiv:2403.10131*, Mar. 2024, doi: 10.48550/arXiv.2403.10131.

[74] S. Xia, A. Kumar, Z. Dai, and S. Gupta, "Ground every sentence: Improving retrieval-augmented LLMs with interleaved reference-claim generation," *arXiv preprint arXiv:2407.01796*, Jul. 2024, doi: 10.48550/arXiv.2407.01796.

[75] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, "Atlas: Few-shot learning with retrieval augmented language models," *J. Mach. Learn. Res.*, vol. 24, pp. 251:1–251:43, 2023.

[76] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020.

[77] O. Ram, Y. Levine, I. Dalmedigos, A. D. Kishore, S. Shashua, K. Leyton-Brown, and Y. Shoham, "In-context retrieval-augmented language models," *Trans. Assoc. Comput. Linguist.*, vol. 11, pp. 1316–1331, 2023, doi: 10.1162/tacl_a_00605.

[78] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," in *Proc. ICLR*, 2020.

[79] "Reducing false positives in retrieval-augmented generation (RAG) semantic caching: A banking case study," *InfoQ*. Accessed: Jan. 21, 2026. [Online]. Available: https://www.infoq.com/articles/reducing-false-positives-retrieval-augmented-generation/

[80] A. Celikyilmaz, E. Clark, and J. Gao, "Evaluation of text generation: A survey," *arXiv preprint arXiv:2006.14799*, Jun. 2020, doi: 10.48550/arXiv.2006.14799.

[81] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated evaluation of retrieval augmented generation," in *Proc. EACL (Demos)*, 2024.

[82] R. Pradeep, N. Thakur, S. Upadhyay, D. Campos, N. Craswell, and J. Lin, "Ragnarök: A reusable RAG framework and baselines for TREC 2024 retrieval-augmented generation track," *arXiv preprint arXiv:2406.16828*, Jun. 2024, doi: 10.48550/arXiv.2406.16828.

[83] "Getting Started," *TruLens*. Accessed: Jan. 31, 2026. [Online]. Available: https://www.trulens.org/getting_started/

[84] "LangSmith Evaluation," *LangChain Docs*. Accessed: Jan. 31, 2026. [Online]. Available:https://docs.langchain.com/langsmith/evaluation

[85] "Observability concepts," *LangChain Docs*. Accessed: Jan. 31, 2026. [Online]. Available: https://docs.langchain.com/langsmith/observability-concepts

[86] "Home," *Arize Phoenix*. Accessed: Jan. 31, 2026. [Online]. Available: https://phoenix.arize.com/

[87] "DeepEval," *Confident AI*. Accessed: Jan. 31, 2026. [Online]. Available: https://deepeval.com/

[88] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, "Seven failure points when engineering a retrieval augmented generation system," *arXiv preprint arXiv:2401.05856*, Jan. 2024, doi: 10.48550/arXiv.2401.05856.

[89] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *Proc. ICLR*, 2020.

[90] I. Popchev, L. Doukovska, and I. Radeva, "A prototype of blockchain/distributed file system platform," in *2022 IEEE 11th Int. Conf. Intell. Syst. (IS)*, Oct. 2022, pp. 1–7, doi: 10.1109/IS57118.2022.10019715.

[91] I. Popchev, L. Doukovska, and I. Radeva, "A framework of blockchain/IPFS-based platform for smart crop production," in *2022 Int. Conf. Automatics and Informatics (ICAI)*, Oct. 2022, pp. 265–270, doi: 10.1109/ICAI55857.2022.9960070.

[92] I. Popchev and I. Radeva, "Decentralized application (dApp) development and implementation," *Cybernetics and Information Technologies*, vol. 24, no. 2, pp. 122–141, Jun. 2024, doi: 10.2478/cait-2024-0019.

[93] AntelopeIO, "Antelope," GitHub repository. Accessed: Jun. 14, 2025. [Online]. Available: https://github.com/AntelopeIO

[94] IPFS, "IPFS Documentation," Accessed: Jun. 14, 2025. [Online]. Available: https://docs.ipfs.tech/

[95] I. Popchev, I. Radeva, and L. Doukovska, "Oracles integration in blockchain-based platform for smart crop production data exchange," *Electronics*, vol. 12, no. 10, Art. no. 2244, Jan. 2023, doi: 10.3390/electronics12102244.

[96] Greymass, "greymass/anchor: Antelope Desktop Wallet and Authenticator," GitHub repository. Accessed: Jan. 08, 2026. [Online]. Available: https://github.com/greymass/anchor

[97] EOSio Support, "Anchor Wallet Overview," Accessed: Mar. 15, 2023. [Online]. Available: https://eosio.support/anchor-wallet-overview/

[98] PrimeTek, "PrimeReact: React UI Component Library," Accessed: Jun. 14, 2025. [Online]. Available:https://primereact.org

[99] NGINX, "nginx," Accessed: Jan. 08, 2026. [Online]. Available: https://nginx.org/

[100] Ollama, "Ollama," Accessed: Jan. 21, 2026. [Online]. Available: https://ollama.com

[101] Chroma, "Chroma," Accessed: Jan. 21, 2026. [Online]. Available: https://www.trychroma.com

[102] Chroma Research, "Evaluating chunking strategies for retrieval," Accessed: Feb. 04, 2026. [Online]. Available: https://research.trychroma.com/evaluating-chunking

[103] LangChain, "LangChain," Accessed: Jun. 14, 2025. [Online]. Available: https://www.langchain.com

[104] PyPI, "pyntelope," Accessed: Jun. 14, 2025. [Online]. Available: https://pypi.org/project/pyntelope/

[105] Modal, "How much VRAM do I need for LLM inference?," Accessed: Feb. 04, 2026. [Online]. Available: https://modal.com/blog/how-much-vram-need-inference

[106] J. Manchanda, L. Boettcher, M. Westphalen, and J. Jasser, "The open source advantage in large language models (LLMs)," *arXiv preprint arXiv:2412.12004*, Dec. 2024, doi: 10.48550/arXiv.2412.12004.

[107] AI21, "What is a long context window? Benefits & use cases," Accessed: Feb. 04, 2026. [Online]. Available: https://www.ai21.com/knowledge/long-context-window/

[108] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.

[109] Mistral AI, "Announcing Mistral 7B," Accessed: Jan. 21, 2026. [Online]. Available: https://mistral.ai/news/announcing-mistral-7b/

[110] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Itani, K. Jiotomo, M. Levy, W. Luo, T. Scialom, G. Sun, K. S. Balaji, A. Sagun, E. Grave, S. Goyal, T. Izacard, A. Kushman, P. Luc, S. Iyer, A. Lomeli, Y. Low, J. Martin, P. Bhargava, M. Sastry, S. Singh, M. Singh, T. Majid, R. Williams, T. Scialom, and J. Zettlemoyer, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, Jul. 2023, doi: 10.48550/arXiv.2307.09288.

[111] A. Mitra, H. S. Liao, M. Moussawi, A. S. Atanasova, A. S. Sestari, H. Song, J. G. Park, J. J. Chung, and J. Huang, "Orca 2: Teaching small language models how to reason," *arXiv preprint arXiv:2311.11045*, Nov. 2023, doi: 10.48550/arXiv.2311.11045.

[112] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Castro, M. S. Lauw, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," *arXiv preprint arXiv:2310.06825*, Oct. 2023, doi: 10.48550/arXiv.2310.06825.

[113] R. Rastogi, "Papers explained: Mistral 7B," DAIR.AI (Medium). Accessed: Mar. 06, 2024. [Online]. Available: https://medium.com/dair-ai/papers-explained-mistral-7b-b9632dedf580

[114] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2021.

[115] GSM8K, "openai/grade-school-math," GitHub repository. Accessed: Feb. 04, 2026. [Online]. Available: https://github.com/openai/grade-school-math

[116] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP Demo)*, Nov. 2018, pp. 66–71, doi: 10.18653/v1/D18-2012.

[117] M. Suzgun, N. S. Abid, A. Adam, E. Ahumada, A. Bansal, T. B. Brown, W. J. Child, E. Choi, D. S. Weld, and L. Zettlemoyer, "Challenging BIG-Bench tasks and whether chain-of-thought can solve them," *arXiv preprint arXiv:2210.09261*, Oct. 2022, doi: 10.48550/arXiv.2210.09261.

[118] IBM, "IBM Granite 3.2: open source reasoning and vision," Accessed: Jan. 21, 2026. [Online]. Available: https://www.ibm.com/new/announcements/ibm-granite-3-2-open-source-reasoning-and-vision

[119] DeepSeek-AI, "deepseek-ai/DeepSeek-R1," GitHub repository. Accessed: Jan. 21, 2026. [Online]. Available: https://github.com/deepseek-ai/DeepSeek-R1

[120] Meta AI, "Introducing Llama 3.1: Our most capable models to date," Accessed: Jan. 21, 2026. [Online]. Available: https://ai.meta.com/blog/meta-llama-3-1/

[121] Mistral AI, "Mistral 7B," Accessed: Jan. 21, 2026. [Online]. Available: https://docs.mistral.ai/models/mistral-7b-0-3

[122] DeepSeek-AI, C. Guo, M. Yang, Z. Bi, K. Zhou, F. Wang, W. Liu, Z. Shao, D. Wang, and G. Dai, "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," *arXiv preprint arXiv:2501.12948*, Jan. 2025, doi: 10.48550/arXiv.2501.12948.

[123] A. Grattafiori, J. Santua, K. Stone, P. Albert, S. Batra, K. J. Chen, A. Chou, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Itani, K. Jiotomo, A. Kushman, P. Luc, M. Martin, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, B. Fuller, C. Gao, V. Goswami, and N. Goyal, "The Llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, Jul. 2024, doi: 10.48550/arXiv.2407.21783.

[124] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[125] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, MI, USA, Jun. 2005, pp. 65–72.

[126] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain, Jul. 2004, pp. 74–81.

[127] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Philadelphia, PA, USA, Jul. 2002, pp. 311–318, doi: 10.3115/1073083.1073135.

[128] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," in *Intell. Data Eng. Autom. Learn. (IDEAL 2013)*, *Lecture Notes in Computer Science*, vol. 8206, pp. 611–618, 2013, doi: 10.1007/978-3-642-41278-3_74.

[129] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2020.

[130] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. 34th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 1996.

[131] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation," in *Advances in Artificial Intelligence (AI 2006)*, *Lecture Notes in Computer Science*, vol. 4304, pp. 1015–1021, 2006, doi: 10.1007/11941439_114.

[132] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, Oct. 2020, doi: 10.48550/arXiv.2010.16061.

[133] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *Amer. Stat.*, vol. 42, no. 1, pp. 59–66, 1988, doi: 10.1080/00031305.1988.10475524.

[134] W. Kirch (ed.), "Pearson's correlation coefficient," in *Encyclopedia of Public Health*. Dordrecht, The Netherlands: Springer, 2008, doi: 10.1007/978-1-4020-5614-7_2569.

[135] H. Kane, M. Y. Kocyigit, A. Abdalla, P. Ajanoh, and M. Coulibali, "NUBIA: NeUral based interchangeability assessor for text generation," in *Proc. 1st Workshop on Evaluating NLG Evaluation*, Dec. 2020, pp. 28–37.

[136] T. Ito, K. van Deemter, and J. Suzuki, "Reference-free evaluation metrics for text generation: A survey," *arXiv preprint arXiv:2501.12011*, Jan. 2025, doi: 10.48550/arXiv.2501.12011.

[137] D. C. Montgomery, *Statistical Quality Control: A Modern Introduction*, 6th ed. Hoboken, NJ, USA: Wiley, 2010.

[138] W. F. Sharpe, "Mutual fund performance," *J. Bus.*, vol. 39, no. 1, pp. 119–138, 1966.

[139] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. New York, NY, USA: Psychology Press, 2009.

[140] Regulation (EU) 2018/848 of the European Parliament and of the Council of 30 May 2018 on organic production and labelling of organic products and repealing Council Regulation (EC) No 834/2007. Accessed: Jan. 21, 2026. [Online]. Available: http://data.europa.eu/eli/reg/2018/848/oj

[141] FAO, "Climate Smart Agriculture Sourcebook," Accessed: Jan. 21, 2026. [Online]. Available: https://www.fao.org/climate-smart-agriculture-sourcebook/en/

[142] scpdxtest, "scpdxtest/PaSSER," GitHub repository, May 30, 2025. Accessed: Jan. 23, 2026. [Online]. Available: https://github.com/scpdxtest/PaSSER

[143] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003, doi: 10.1023/A:1022859003006.

[144] Convention on Biological Diversity, "The Convention on Biological Diversity," Accessed: Jan. 07, 2026. [Online]. Available: https://www.cbd.int/convention

[145] European Commission, "Biodiversity Strategy for 2030," Accessed: Jan. 07, 2026. [Online]. Available: https://environment.ec.europa.eu/strategy/biodiversity-strategy-2030_en

# SUPPORTING PUBLICATIONS

Five peer-reviewed publications underpin the reported research:

**[15]** M. Dimitrova, I. Popchev, and I. Radeva, **PaSSER: A Platform for Evaluating LLMs in RAG.** Proceedings of the 9th IEEE International Conference on Big Data, Knowledge and Control Systems Engineering – BdKCSE'2025, 06-07 November 2025, Bankya, Bulgaria, IEEE Xplore, 2025, ISSN:979-8-3315-8712-3, DOI:10.1109/BdKCSE67969.2025.11300500, 1-7.
This work describes the PaSSER platform architecture and functionalities detailed in Chapter 2.

**[16]** I. Radeva, I. Popchev, and M. Dimitrova, **Similarity Thresholds in Retrieval-Augmented Generation.** Proceedings of the 12th IEEE International Conference on Intelligent Systems - IS'24, 29-31 August 2024, Varna, Bulgaria, IEEE Xplore, 2024, ISBN:979-8-3503-5098-2, ISSN:2832-4145, DOI:10.1109/IS61756.2024.10705214, 1-7.
This work supports the CPS formulation and threshold sensitivity analysis presented in Chapter 4 (Phase II).

**[17]** I. Radeva, I. Popchev, L. Doukovska, and M. Dimitrova, **Web Application for Retrieval-Augmented Generation: Implementation and Testing.** Electronics, 13, 7, MDPI, Basel, Switzerland, 2024, ISSN:2079-9292, DOI:10.3390/electronics13071361, 1-31. SJR (Scopus):0.64, JCR-IF (Web of Science):2.9.
This work presents the PaSSER platform and metrics discussed in Chapter 3.

**[18]** I. Radeva, I. Popchev, L. Doukovska, and M. Dimitrova, **Multi-Agent Coordination Strategies vs Retrieval-Augmented Generation in LLMs: A Comparative Evaluation.** Electronics, 14, 24, MDPI, 2025, ISSN:2079-9292, DOI:10.3390/electronics14244883, SJR (Scopus):0.62.
This work documents the T-CPS and Balanced Score reported in Chapter 4 (Phase IV).

**[20]** M. Dimitrova, **Retrieval-Augmented Generation (RAG): Advances and Challenges.** Problems of Engineering Cybernetics and Robotics, 83, Prof. Marin Drinov Academic Publishing House, 2025, ISSN:2738-7356, DOI:10.7546/PECR.83.25.03, 32-57.

This work provides the RAG literature review and frameworks analysis that form the foundation of Chapter 1.

Publications [17] and [18] are indexed in JCR-IF (Web of Science) and SJR (Scopus). Conference papers [15] and [16] are indexed in IEEE Xplore Digital Library. Article [20] is published by Prof. Marin Drinov Academic Publishing House of the Bulgarian Academy of Sciences.

# CITATION RECORD

**[1] I. Radeva, I. Popchev, L. Doukovska, and M. Dimitrova, Web Application for Retrieval-Augmented Generation: Implementation and Testing. Electronics, 13, 7, MDPI, Basel, Switzerland, 2024, ISSN:2079-9292, DOI:10.3390/electronics13071361, 1-31. SJR (Scopus):0.64, JCR-IF (Web of Science):2.9.**

*Cited by:*

**[1.1]** H. Andersson, "Retrieval-augmented generation with Azure OpenAI," M.S. thesis, Malardalen Univ., 2024.

**[1.2]** S. D'Urso, B. Martini, and F. Sciarrone, "A Novel LLM Architecture for Intelligent System Configuration," in Proc. Int. Conf. Information Visualisation (IV), Coimbra, Portugal, 2024, pp. 326-331, doi:10.1109/IV64223.2024.00063.

**[1.3]** D. Firdaus, I. Sumardi, and Y. Kulsum, "Integrating Retrieval-Augmented Generation With Large Language Model Mistral 7B for Indonesian Medical Herb," JISKA, vol. 9, no. 3, pp. 230-243, 2024, doi:10.14421/jiska.2024.9.3.230-243.

**[1.4]** H. Zhang, Z. Li, F. Liu, Y. He, Z. Cao, and Y. Zheng, "Design and Implementation of LangChain-based Chatbot," in Proc. Int. Seminar on AI, Computer Technology and Control Engineering (ACTCE), Wuhan, China, 2024, pp. 226-229, doi:10.1109/ACTCE65085.2024.00053.

**[1.5]** J. G. Ongris, E. Tjitrahardja, F. Darari, and F. J. Ekaputra, "Towards an Open NLI LLM-based System for KGs: A Case Study of Wikidata," in Proc. 7th Int. Seminar on Research of IT and Intelligent Systems (ISRITI), 2024, pp. 44-49, doi:10.1109/ISRITI64779.2024.10963661.

**[1.6]** C. K. Kitengera and M. K. Kasambya, "Developpement d'une plateforme web d'evaluation des enseignements...," Revue Internationale Multidisciplinaire Etincelle, vol. 25, no. 2, pp. 1-22, 2024, doi:10.61532/rime252117.

**[1.7]** B. Lu, "Evaluating LLMs on large contexts: a RAG approach on text comprehension," Master's thesis, Univ. de Liege, 2024.

**[1.8]** M. M. Li, I. Nikishina, O. Sevgili, and M. Semman, "Wiping out the limitations of Large Language Models: A Taxonomy for Retrieval Augmented Generation," arXiv:2408.02854, 2024, doi:10.48550/arXiv.2408.02854.

**[1.9]** M. Olaosegba, "Next-Gen AI Optimization Tools for AWS Cloud Cost Control," IJFMR, 2024.

**[1.10]** P. Phukon, Y. Lokhar, and P. P. Ray, "Localized Open-Source LLM Aware RAG of Legal Documents...," in Proc. Int. BIT Conf. (BITCON), 2024, pp. 1-6, doi:10.1109/BITCON63716.2024.10985396.

**[1.11]** S. Rani, S. G. Deepika, D. Devdharshini, and H. Ravindran, "Augmenting Code Sequencing with RAG...," in Proc. SSITCON, 2024, pp. 1-7, doi:10.1109/SSITCON62437.2024.10796587.

**[1.12]** S. Dudhmande et al., "Textual Compression Using Lamini-LM," IRJAEM, vol. 2, no. 5, pp. 1536-1540, 2024, doi:10.47392/IRJAEM.2024.0208.

**[1.13]** S. Bouzid and L. Piron, "Leveraging Generative AI in Short Document Indexing," Electronics, vol. 13, no. 17, 2024, doi:10.3390/electronics13173563.

**[1.14]** K. Traykov, "A Framework for Security Testing of Large Language Models," in Proc. 12th IEEE Int. Conf. on Intelligent Systems (IS), Varna, Bulgaria, 2024, pp. 1-7, doi:10.1109/IS61756.2024.10705238.

**[1.15]** L. Werkman, "Assessing the potential of leveraging LLaMA-2...," thesis, Lulea Univ. of Technology, 2024.

**[1.16]** W. Wilmi and N. Roslund, "Implementering av RAG for automatiserad analys av hallbarhetsrapportering...," thesis, KTH Royal Inst. of Technology, 2024.

**[1.17]** Y. Xu et al., "Development of an Enterprise Knowledge Base System Based on Elasticsearch," in Proc. ISPCEM, 2024, pp. 186-190, doi:10.1109/ISPCEM64498.2024.00039.

**[1.18]** Y. Song, Enhancing Classroom Dialogue Productiveness: Exploring the Potential of Artificial Intelligence. London, U.K.: Routledge, 2024, doi:10.4324/9781003543039.

**[1.19]** Z. Zhong et al., "Mix-of-Granularity: Optimize the Chunking Granularity for RAG," arXiv:2406.00456, 2024.

**[1.20]** J. O. Agada et al., "A Systematic Review of Key RAG Systems...," arXiv:2507.18910, 2025, doi:10.48550/arXiv.2507.18910.

**[1.21]** A. Guyyala et al., "RAG-based AI Agents for Multilingual Help Desks...," Int. J. Computer Applications, vol. 187, no. 56, pp. 15-28, 2025, doi:10.5120/ijca2025925964.

**[1.22]** O. Barcelos et al., "Technological Convergence Identification Model (TCIM)...," Revista E-TECH, vol. 18, no. 1, 2025, doi:10.18624/e-tech.v18i1.1444.

**[1.23]** C. Yu et al., "Safety Devolution in AI Agents," 2025, doi:10.48550/arXiv.2505.14215.

**[1.24]** D. Costa et al., "Mycroft: Retrieval Augmented Generation for SDK Documentation," in Proc. NATL, 2025, doi:10.5121/csit.2025.152211.

**[1.25]** R. Dayarathne et al., "Comparing the Performance of LLMs in RAG-Based QA...," in AI in Education Technologies, LNDECT, vol. 228. Singapore: Springer, 2025, doi:10.1007/978-981-97-9255-9_26.

**[1.26]** E. H. Omoush et al., "Advancing Arabic Medical QA Systems with RAG...," in Proc. ICTCS, 2025, pp. 511-516, doi:10.1109/ICTCS65341.2025.10989446.

**[1.27]** Y. Fan et al., "Research on the Online Update Method for RAG Model...," in Proc. NNICE, 2025, pp. 1740-1744, doi:10.1109/NNICE64954.2025.11063821.

**[1.28]** F. Shen et al., "Development of a Convenient Accounting System Based on SpringBoot+Vue," in Proc. CITSC, 2025, pp. 167-171, doi:10.1109/CITSC64390.2025.00038.

**[1.29]** F. Ehrlich-Sommer et al., "ForestGPT and Beyond...," Electronics, vol. 14, no. 18, p. 3583, 2025, doi:10.3390/electronics14183583.

**[1.30]** H. Mahfoud et al., "AI Chatbots for Healthcare Maintenance...," TQM J., 2025, doi:10.1108/TQM-10-2024-0394.

**[1.31]** G. Iieva and G. A. Tsihrintzis, "Editorial Note to Special Issue...," Electronics, vol. 14, no. 10, p. 1925, 2025, doi:10.3390/electronics14101925.

**[1.32]** L. A. Sanjani et al., "Performance Analysis of LLM Models with RAG and Fine-Tuning T5...," in Proc. ICoCSETI, 2025, pp. 152-157, doi:10.1109/ICoCSETI63724.2025.11018908.

**[1.33]** B. T. Mahardika and A. M. Hasan, "Application of GPT in Chatbots...," Eduvest, vol. 5, no. 6, pp. 6235-6247, 2025, doi:10.59188/eduvest.v5i6.51321.

**[1.34]** N. A. Akbar et al., "Novel Approach for Leveraging Agent-Based Experts...," in AIxIA 2024, LNCS, vol. 15450. Cham, Switzerland: Springer, 2025, doi:10.1007/978-3-031-80607-0_2.

**[1.35]** B. M. Praneeth et al., "Optimization of Customer Feedback Summarization...," IEEE Access, vol. 13, pp. 124319-124332, 2025, doi:10.1109/ACCESS.2025.3588337.

**[1.36]** P. Pany, "Reasoning Engine with Pre-Trained LLMs: An Operation GPT," IJRASET, vol. 13, no. 4, pp. 2452-2463, 2025, doi:10.22214/ijraset.2025.68761.

**[1.37]** S. K. Mahjour and S. S. Mahjour, "Intelligent Reservoir Decision Support...," 2025, doi:10.48550/arXiv.2509.11376.

**[1.38]** S. Chen et al., "Customized large-scale model for human-AI collaborative operation...," Appl. Energy, vol. 393, pp. 126-169, 2025, doi:10.1016/j.apenergy.2025.126169.

**[1.39]** T. Jung and I. Joe, "An Intelligent Docent System with a Small Language Model (sLLM) Based on RAG," Appl. Sci., vol. 15, no. 17, p. 9398, 2025, doi:10.3390/app15179398.

[1.40] C.-N. Tirpescu and E. Velescu, "Enhancing Veterinary Education...," Procedia Comput. Sci., vol. 270, pp. 3828-3837, 2025, doi:10.1016/j.procs.2025.09.508.

**[1.41]** W. Ke et al., "Large Language Models in Document Intelligence: A Comprehensive Survey...," ACM Trans. Inf. Syst., vol. 44, no. 1, 2025, doi:10.1145/3768156.

**[1.42]** A. J. Winata et al., "Utilizing Large Language Models for Developing Automatic Question Generation in Education," in Proc. ICADEIS, 2025, doi:10.1109/ICADEIS65852.2025.10933227.

**[1.43]** Y. Benitez-Morejon et al., "Question-Answering Systems for Tourism...," in MISNC 2025, CCIS, vol. 2729. Cham, Switzerland: Springer, doi:10.1007/978-3-032-09945-7_22.

**[1.44]** J. Qi, Mitigating Translation Hallucinations in Large Language Models: A Chain of Thought and RAG-Based Approach, Ph.D. research proposal, The Chinese Univ. of Hong Kong, 2024-2025.

**[1.45]** R. Kumar and Y. Qu, "Utilizing Large Language Model Enabled Agents to Streamline Business Decision Making," Eur. J. Electr. Eng. Comput. Sci., vol. 9, no. 5, pp. 14-21, Sep. 2025, doi:10.24018/ejece.2025.9.5.717.

**[1.46]** I. P. A. E. Pratama, I. M. O. Widyantara, Linawati, and N. Gunantara, "Bibliometric Analysis of AI-Based Prototype Proposal for User Security Awareness in Healthcare," JOIV: Int. J. Informatics Visualization, vol. 9, no. 3, pp. 982-994, May 2025, doi:10.62527/joiv.9.3.3319.

**[1.47]** S. Gandla, Automated Test Code Generation from Textual Descriptions Using Generative AI, Master's thesis, Blekinge Inst. of Technology, 2024.

**[1.48]** N. S. Patil, A. J. Koyande, A. V. Thakur, P. B. Kadam, and P. G. Moholkar, "RAG Chatbots: Implementing Large Language Models in Retrieval-Augmented Generations," in Smart Trends in Computing and Communications, LNNS, vol. 1363, pp. 401-410, 2025, doi:10.1007/978-981-96-2885-8_33.

**[1.49]** I. Balen, Sustav korisnicke podrske temeljen na bazi znanja i koristenju alata umjetne inteligencije za brzo odgovaranje na ponavljajuca pitanja korisnika,

Undergraduate thesis (Zavrsni rad), Faculty of Electrical Engineering and Computing (FER), Univ. of Zagreb, Jun. 2024.

**[1.50]** Y. Jiao, S. Ouyang, M. Zhong, Y. Zhang, L. Ding, S. Zhou, and J. Han, "Retrieval and Structuring Augmented Generation with LLMs for Web Applications," in Companion Proc. ACM Web Conf. 2025 (WWW '25 Companion), pp. 25-28, May 2025, doi:10.1145/3701716.3715870.

**[1.51]** Z. Liu, Design and Implementation of an AI-based Agent to Inform Best Practices on Test Case Execution Routines, Master's thesis, Univ. of Zurich, Jun. 29, 2025, doi:10.5167/uzh-278942.

**[2] I. Radeva, I. Popchev, and M. Dimitrova, Similarity Thresholds in Retrieval-Augmented Generation. Proceedings of the 12th IEEE International Conference on Intelligent Systems - IS'24, 29-31 August 2024, Varna, Bulgaria, IEEE Xplore, 2024, ISBN:979-8-3503-5098-2, ISSN:2832-4145, DOI:10.1109/IS61756.2024.10705214, 1-7.**

*Cited by:*

**[2.1]** D. Ayepah-Mensah et al., "A RAG-Assisted DRL Framework for Microservices Deployment in 6G Vehicular Networks," in Proc. WiMob 2025, Marrakesh, Morocco, 2025, pp. 1-6, doi:10.1109/WiMob66857.2025.11257559.

**[2.2]** Y. Bondalapati and H. N. BM, "Scalable RAG with Kubernetes for Enhanced Document Intelligence," in Proc. CICCS 2025, Bengaluru, India, 2025, pp. 1-6, doi:10.1109/CICCS66437.2025.11280266.

**[2.3]** A. Jadhav et al., "AI-Driven Diagnosis Predictive Chatbot for Healthcare," in Proc. WorldSUAS 2025, 2025, doi:10.1109/WorldSUAS66815.2025.11199219.

**[2.4]** J. Van Nooten et al., "One Size Does Not Fit All: Exploring Variable Thresholds for Distance-Based Multi-Label Text Classification," arXiv:2510.11160, 2025, doi:10.48550/arXiv.2510.11160.

**[2.5]** X. Sun, C. Liang, Q. Wang, et al., "Mesh RAG: Retrieval Augmentation for Autoregressive Mesh Generation," arXiv:2511.16807, 2025.

**[2.6]** K. Traykov and Y. Kolova, "Analysis of Methods for Evaluating Responses of LLMs in Retrieval-Augmented Generation," in Proc. Int. Conf. on Big Data, Knowledge and Control Systems Engineering, 2025, pp. 1-6.

**[2.7]** A. Kosar, W. Daelemans, and G. De Pauw, Dont Make Me Guess: Automatically Detecting and Naming Topics in Large Collections of Text. Antwerp, Belgium: Univ. of Antwerp, Faculty of Arts, 2025.

**[2.8]** J. Van Nooten and W. Daelemans, The Many Faces of a Text: Applications and Enhancements of Multi-Label Text Classification Algorithms. Antwerp, Belgium: Univ. of Antwerp, Faculty of Arts, 2025.

**[2.9]** T. Bosi, Design, Implementation and Benchmarking of a Retrieval-Augmented Chatbot for the Insurance Sector, Master's thesis (Laurea magistrale), Univ. of Bologna, 2025.

**[2.10]** J. Karkoush and M. Ali, Kallgranskning med RAG och smasprakmodeller, Student thesis (Basic level, 15 HE credits), Univ. of Gavle, 2025, 46 pp., URN: urn:nbn:se:hig:diva-47778.

**[3] I. Radeva, I. Popchev, L. Doukovska, and M. Dimitrova, Multi-Agent Coordination Strategies vs Retrieval-Augmented Generation in LLMs: A Comparative Evaluation. Electronics, 14, 24, MDPI, 2025, ISSN:2079-9292, DOI:10.3390/electronics14244883, SJR (Scopus):0.62.**

*Cited by:*

**[3.1]** M. Nababan and G. Simarmata, "Model Matematika Dalam Pemilihan Mekanisme Koordinasi...," Jurnal Ilmiah Matematika (JIMAT), vol. 6, no. 2, pp. 891-902, Dec. 2025, doi:10.63976/jimat.v6i2.1201.

**[3.2]** S. Schmulling and G. Sanrocco, "Ensembles of Small Language Models as an Efficient Alternative to Large Language Models," course report (II2202, Fall 2025, Period 1/Period 1-2), KTH Royal Inst. of Technology, Stockholm, Sweden, Jan. 14, 2026.

**[4] M. Dimitrova, Retrieval-Augmented Generation (RAG): Advances and Challenges. Problems of Engineering Cybernetics and Robotics, 83, Prof. Marin Drinov Academic Publishing House, 2025, ISSN:2738-7356, DOI:10.7546/PECR.83.25.03, 32-57.**

*Cited by:*

**[4.1]** M. E. Koutsiaki, M. Delianidi, C. Mizeli, K. Diamantaras, I. Grigoropoulos, and N. Koutlianos, "From Textbook to Talkbot: A Case Study of a Greek-Language RAG-Based Chatbot in Higher Education," arXiv:2601.14265, 2025

# SUMMARY OF PROJECT PARTICIPATION

The research was conducted with support from the Bulgarian Ministry of Education and Science under:

**1.** The National Research Program "Smart Crop Production" (Decision of the Ministry Council No. 866/26.11.2020);

**2.** The Scientific Research Fund project "BG PLANTNET: Establishment of a National Information Network Genebank—Plant Genetic Resources" (KP-06-N36).

# ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to the colleagues and researchers from the Intelligent Systems Department at the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, for their support, professionalism, and the stimulating academic environment in which this dissertation was developed. The constructive discussions, shared expertise, and collegial atmosphere within the department contributed significantly to the progress and completion of this research.

# DECLARATION OF ORIGINALITY OF THE RESULTS

I declare that this dissertation contains original results obtained through scientific research conducted by me. The results that have been obtained, described, and/or published by other researchers are duly and thoroughly cited in the bibliography.

This dissertation has not been submitted for the acquisition of an academic degree at another higher education institution, university, or research institute.

Signature: …… 

Miroslava Doncheva Dimitrova