

WHITEPAPER

Building Trustworthy Enterprise AI with Hybrid Tree-based RAG Architecture

Combining Sparse-dense
Retrieval with Hierarchical
Document Indexing for Enterprise
AI Search, Enterprise Policy and
Compliance Question Answering



Table of Contents

Abstract	3
Introduction	4
Theoretical Foundations	6
Tree-based Document Structure	7
Tree-based Architecture	7
System Architecture Overview	7
End-to-end Query Workflow	8
Hybrid Retrieval: Sparse-dense Fusion	8
Hybrid Retrieval Architecture	9
System Architecture & Implementation	10
Evaluation Framework & Results	10
Evaluation & Metrics Pipeline	10
Threats to Validity & Limitations	11
Enterprise Value & Business Impact	12
Experimental Evaluation and Results	12
Conclusion & Future Work	13
References	13
About the Authors	14

Abstract

Building trustworthy enterprise AI that enables better governance, not merely better search, is essential for modern businesses. The innovation to get this right requires not just retrieval accuracy, but the establishment of a trusted knowledge foundation for enterprise AI. The quality, traceability and reliability of the knowledge which is retrieved to support business outcomes and regulatory compliance is becoming critical, as organizations adopt copilots and AI assistants throughout business functions.

Many businesses face the challenge of balancing retrieval accuracy, response relevance, and trustworthiness while maintaining the efficiency and scalability of traditional RAG methodologies. By blending hierarchical database indexing with hybrid-powered retrieval, organizations can ensure that the availability of their knowledge assets increases in relation to the expected query volume, while the infrastructure costs remain manageable. This hybrid approach to scaling knowledge access helps organizations make enterprise AI more sustainable, as indexing holds true to a hierarchical database structure and hybrid retrieval adjusts the system to meet the query volume.

This whitepaper presents a Hybrid Tree-Based RAG architecture using Reciprocal Rank Fusion of BM25 and dense embeddings to enhance enterprise policy search that serves to build trustworthy AI with better governance. This coarse-to-fine approach improves efficiency while ensuring high recall.



Introduction

The Enterprise Document Challenge

The businesses in today's day and age are faced with a much more complex environment of regulatory needs, internal controls and compliance requirements. There are thousands of governance documents, regulatory guidelines, audit records, standard operating procedures (SOPs) and policy documents across business functions in an organization. These repositories serve as the backbone of efficient access to accurate and relevant information for employees, auditors, and automated compliance systems to make informed decisions and ensure regulatory compliance. The problem is not just about keywords, but about how people ask questions in their natural language, which may differ greatly from the legalistic, policy-specific terms that are often included in documents.

Let's take an actual example: A compliance officer asks, 'What happens if customers request deletion of their data?' The section of the policy may be titled 'Data Subject Rights Under GDPR Article 17' and contain formal language like 'erasure requests' and 'right to be forgotten'. Traditional search systems do not have the ability to solve this vocabulary mismatch problem, while purely semantic approaches can return privacy policies that are tangentially related to the discussion of similar concepts and deal with completely different regulatory contexts.

Limitations of Existing Approaches

Vector-based RAG

- Great at reading between the lines: Understands the meaning and conceptual meaning of a question
- Semantic drift: Can be misled by words that seem to be relevant but are not accurate.

BM25 Retrieval

- Ace is able to fetch a range of keywords, uncommon words and alphanumeric sequences.
- Vocabulary locked: The ability to correlate the words that users use as synonyms for words in the document is not allowed.
- Sensitive to the wording: Responds strongly to rewording the question or paraphrasing it by the user.

Enterprise Impact

None of these approaches is alone reliable enough to be used in an enterprise context, where it may return incomplete, irrelevant or unauthorized information. This can lead to a higher likelihood of sensitive information being leaked, accessed without authorization, not meeting regulatory standards, or inaccuracies in the responses generated by AI. To address these risks, Hybrid RAG fuses lexical and semantic retrieval, enhancing accuracy and governance.

Novel Aspects of the Hybrid Tree-based RAG Architecture

The architecture proposed introduces some innovations that enhance retrieval accuracy, efficiency and reliability in enterprise environments.



Hierarchical Document Indexing

An Index similar to the repository structure that enables coarse-to-fine retrieval by pruning early in the process.



Unified Sparse-dense Fusion

Fusion BM25 and vector retrieval with Reciprocal Rank Fusion (RRF) over categories, documents and chunks to improve retrieval quality.



Enterprise Reliability Controls

Introduces confidence thresholds, source citation enforcement, and the ability to optionally include faithfulness checks to reduce hallucination and increase confidence.



Comprehensive Evaluation

Clearly communicates methodology, what the data represents, and any limitations to the data, allowing for proper interpretation of the results.

Distinctive Features of the Proposed Approach

This work integrates the design of hybrid retrieval, hierarchical indexing and rank fusion into a single design for enterprise-scale retrieval.

- **Hierarchical Fusion:** Sparseness and denseness retrieval signals are fused at category-level, document-level, and chunk-level.
- **Preservation of Retrieval Signals:** Multi-level fusion can achieve a well-preserved information in the early stage of pruning and improve the retrieval efficiency by utilizing the preserved retrieval signal.
- **Reliability-aware Generation:** Utilizes Reciprocal Rank Fusion (RRF) and retrieval-confidence scores to support uncertainty-aware generation for answers and mitigate hallucination errors.

The retrieval confidence signals are directly coupled with our hallucination control signals, rather than merely after generation filtering. The mechanism works in the following way:

Threshold Calibration: Empirically, a held-out test set of 50 queries (not in the 156-query validation set) is used to empirically calibrate a confidence threshold of 0.15. We experimented with thresholds ranging between 0.05 and 0.25 in steps of 0.025 and selected 0.15 as the threshold that gave the best balance between (a) getting uncertain responses to non-answerable queries, and (b) avoiding false rejections of answered queries.

Limitations: This measurement corresponds to performance in our particular test data, and annotation protocol. The absolute rate of hallucination and percentage rate of reduction can depend on the various domains of documents, how the queries are distributed, or even the annotation criteria.

Tree Structure Assumptions

Our deployment categories are built out of the existing metadata on documents (department ownership, regulatory domain, policy type) that have been filtered by domain experts in a period of about two weeks. In the case of organizations that do not have metadata, the architecture enables bootstrapping of initial category structures by embedding-based clustering.

If similarity to category centroids exceeds 0.65, new documents are automatically assigned and otherwise, remain to be reviewed by human. One document (from sample corpus) updates take 15-45 seconds with chunking, embedding generation and index updates. Full corpus reindexing is done when model upgrades happen, and takes about 4 hours on GPU. The statistics are updated by document incrementally and it takes less than 500ms per document.

Theoretical Foundations

Information Retrieval Paradigms

The field of information retrieval research has progressed with a series of paradigms, each with respective strengths. Sparse retrieval, where only terms that are present in documents have non-zero weight. BM25 algorithm was developed in the 1990's and provides scores as probabilities that are easily understood and determined by term frequencies and document length normalization statistics.

Sparse Retrieval: A retrieval method using sparse term-based vectors (usually few nonzero weights in vectors). Uses lexical keyword matching and weighting techniques (BM25 and SPLADE) to retrieve relevant papers

It is unmatched in being open and efficient to access using keywords. With the advancements of neural language models, dense retrieval was obtained, which involved encoding the text into small vectors (typically 384-768 dimensions) that carry semantic relationships learned from large text corpora. Models like Sentence-BERT enable these to calculate similarities efficiently with approximate nearest neighbor algorithms, thereby enabling semantic matching across the enterprise scale.

Hierarchical Information Retrieval

The traditional hierarchical IR study is used as the inspirations for our tree-based design. Document collections are naturally hierarchical, in that documents are grouped into thematic categories, sections are subsections of documents and paragraphs are subsections of sections. This organizational fact is one of the results of the human behavior to find their way in a huge set of documents.

Hierarchical retrieval uses this hierarchy by coarse-to-fine search strategies. The system first identifies pertinent high-level categories and then probes into the wooded parts of the category trees. This is related to faceted search, hierarchical language models and cascade retrieval systems in literature.

Tree-based Document Structure

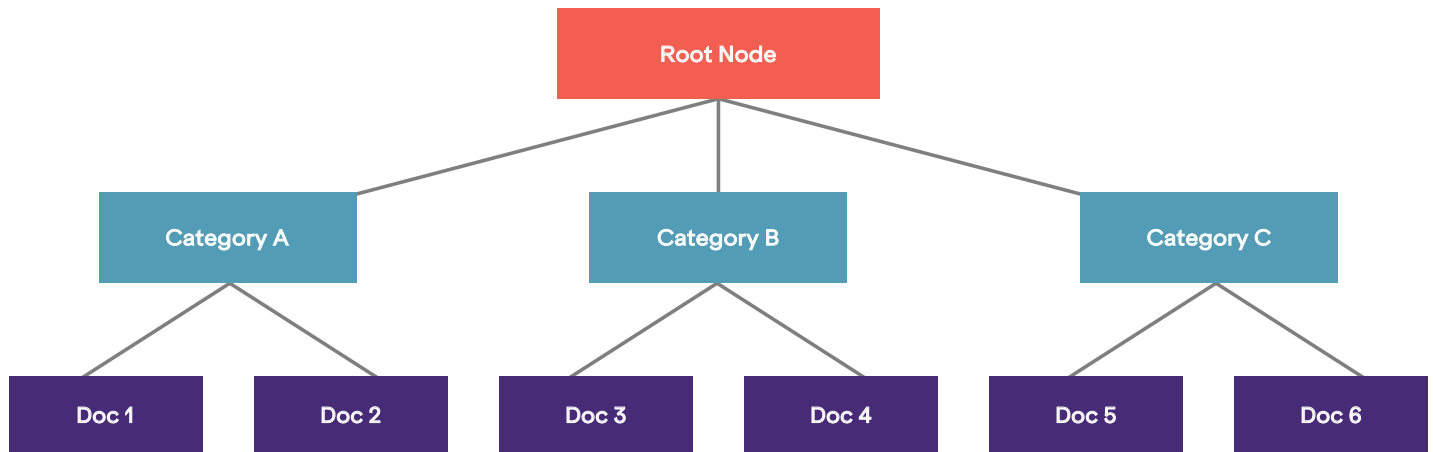


Figure 1: Hierarchical document index structure with root node, category nodes, and document leaf nodes

Retrieval-augmented Generation (RAG)

RAG is a shift in paradigm of the use of purely parametric language models to systems that are explicitly based on retrieved evidence. This is a direct method of dealing with hallucination by training generation towards verifiable source material. We provide a structure-conscious extension of standard RAG with structure-aware retrieval, that maintains and uses hierarchical relationships both in indexing and retrieval.

Tree-based Architecture

Hierarchical Document Indexing

The five levels of the architecture are based on a natural organization of policy repositories: Root, Categories (Data Privacy, Financial Compliance, HR, IT Security, Legal), Documents, Sections, and Chunks (256-512 tokens). BM25 and vector indices are stored on each node, allowing one to use hybrid retrieval anywhere in the cascade. This dual index design makes sure that both the retrieval paradigms do not take charge at any point of decision.

System Architecture Overview

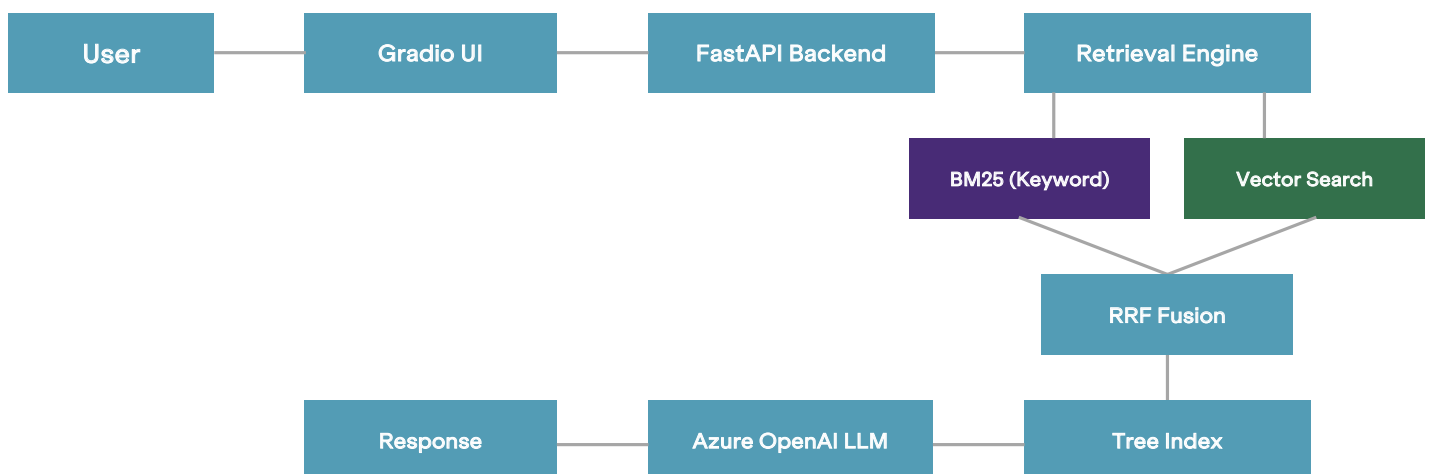


Figure 2: High-level system architecture showing the flow from user query through Gradio UI, FastAPI backend, hybrid retrieval engine (BM25 + Vector), RRF fusion, tree index, and LLM generation

Top-down Pruning Strategy

All queries are scored with hybrid matching and the top 3 of these are kept:

- Score all categories with hybrid matching, keep top 3.
- Take 10 of each of top 10 of mark papers for chosen categories.
- Retrieve chunks in surviving documents (full hybrid).

With this pruning, the number of searches is limited by 85-95%. If you have a 100,000 chunked tree, and reduce that to 5,000 to 8,000 chunks that actually provide a search candidate, then you will have a 10-20x reduction in computation, but coverage is still accomplished in an intelligent way.

End-to-End Query Workflow



Figure 3: Query processing workflow illustrating the eight-stage pipeline from user query to final response

Fan-out Management

Parallel hybrid search is at risk of exponential fan-out up the hierarchy. We enable early pruning to remove 70-80% of the candidates to perform the chunk search, shared index structures to efficiently merge results, asynchronous parallel processing of BM25 and vector search operations, and hard budget constraints to guarantee that latency would not get out of control in adversarial conditions.

Hybrid Retrieval: Sparse-dense Fusion

BM25 Sparse Retrieval

This principled probabilistic formulation is what is used to compute relevance scores in BM25:

$$\text{BM25}(D,Q) = \sum \text{IDF}(q_i) \cdot [f(q_i,D) \cdot (k_1+1)] / [f(q_i,D) + k_1 \cdot (1-b+b \cdot |D|/\text{avgdl})]$$

The frequency term $f(q_i,D)$ is assigned by the frequency of terms in document D , $k_1=1.5$ controls the saturation of terms, $b=0.75$ controls the normalization of length and the inverse document frequency term IDF weights the terms by the inverse document frequency. This base provides interpretable scores especially useful in the exact terminology and regulatory citation matching.

Dense Vector Retrieval

The queries and documents are encoded to 384-dimensional embeddings using All-MiniLM-L6-v2. Relevance is measured by the cosine similarity of the encoded representations. Semantic similarity matching between synonyms and paraphrases without a lexical match is possible by the encoder that has been pre-trained with semantic relationships, but it is impossible for BM25.

Reciprocal Rank Fusion

RRF ranks the lists from several retrieval systems together into a single ranking:

$$\text{RRF_score}(d) = \sum_i R [w_i / (k + r_i(d))]$$

$d = \text{document}$

We tested the set of ranks ($k \in \{10, 30, 60, 90\}$) of document d in system i (1-indexed), to find the optimum smoothing factor.

We noticed that the results were stable for $(k) = 60$ was 60 and w_i (Optional System Weighting). Documents that are seen in one list would have one-system contributions. The rank-based aggregation provided by RRF is scale-invariant, which is important when fusion of unbounded BM25 scores and limited cosine similarity is applied across levels of a hierarchy that have vastly different numbers of candidates.

The system weighting is optional, and it is referred to as “ w_i ,” and it is used for effective usage of rank-based aggregation.

Hybrid Retrieval Architecture

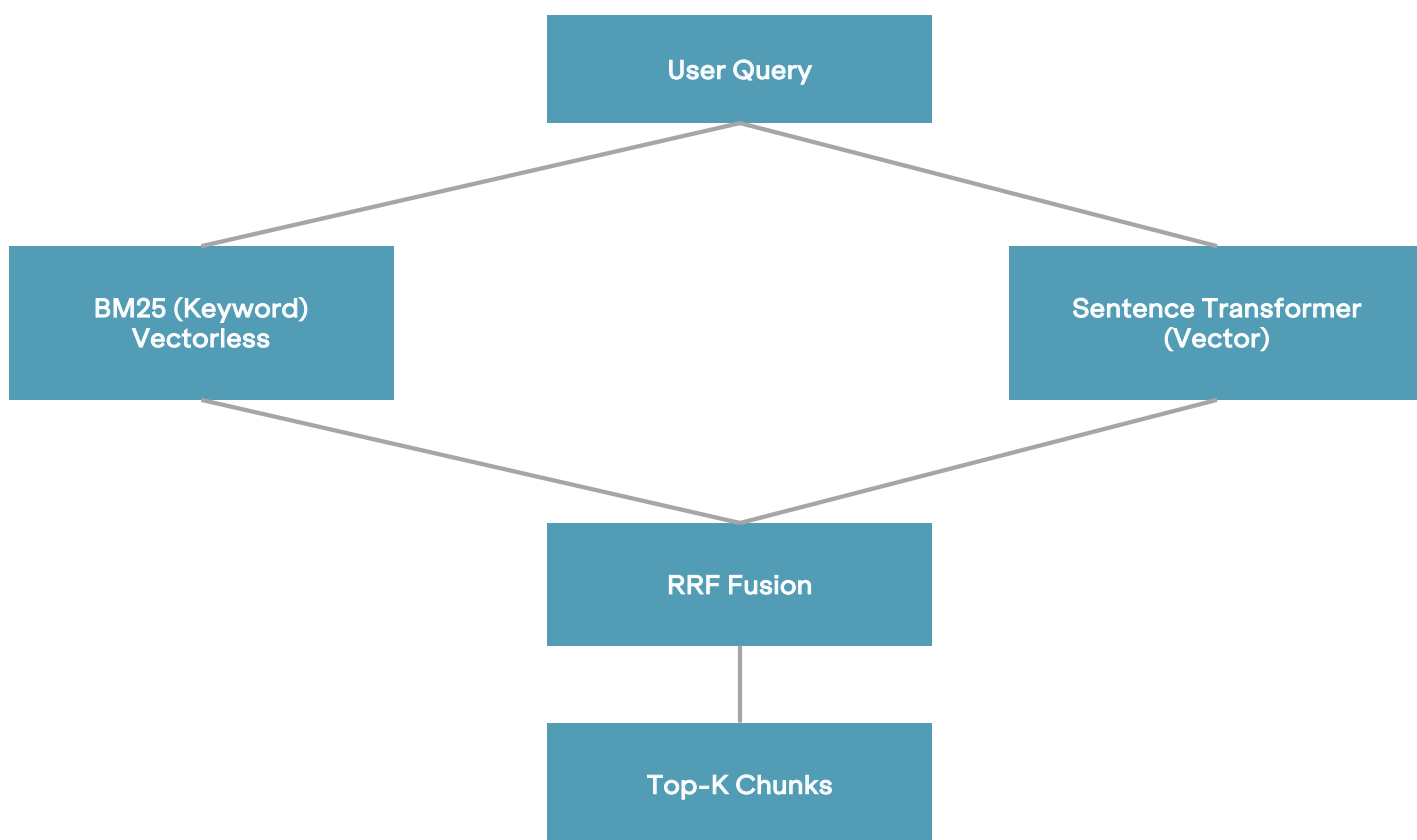


Figure 4: Hybrid retrieval architecture depicting parallel BM25 (vectorless) and Sentence Transformer (vector) paths converging at the RRF fusion layer

Why RRF Over Alternatives?

There is a need to carefully calibrate weighted score fusion, which does not generalize across query type and hierarchy level. Learned fusion is expensive to obtain relevance judgments (labeled ones) and can suffer from overfitting on some document collections. The max-pooling ignores the important agreement information across systems. RRF is a parameter-less, training-free, decades-long benchmark, and documents with good retrieval results in both sparse and dense retrieval are intrinsically rewarded.

System Architecture & Implementation

The entire system consists of 5 major components that are connected through a horizontally scalable and reliable FastAPI backend:

- **Document Ingestion Pipeline:** PDF/DOCX extraction with structure, hierarchical chunking, respecting section boundaries and parallel BM25 index building and generation of embedding.
- **Query Processing Engine:** Query preprocessing, embedding generation using the same encoder as documents, query expansion (Optional), acronym and synonym handling.
- **Hybrid Retrieval Engine:** Tree-based category and document-level pruning, BM25 and vector search, RRF combination with parameterizable parameters, top-K selection with diversity filtering.
- **Hallucination Control Module:** Hallucination is averted by generating explicit uncertainty response prior to the generation when the maximum RRF score drops below 0.15- avoiding hallucination at the source, as opposed to filtering after the fact.

Evaluation Framework & Results

Evaluation Context

We record full assessment context, so that it can be reproduced and interpreted appropriately. Corpus: Total of 847 policy documents with an average of 312 tokens per document, with chunks in four different domains: Data Privacy, Financial Compliance, HR and IT Security, and Legal. Test dataset: 156 question–answer pairs that were constructed by experts, who had the latest knowledge about the document collection. Evaluation period: April 2026.

Evaluation & Metrics Pipeline

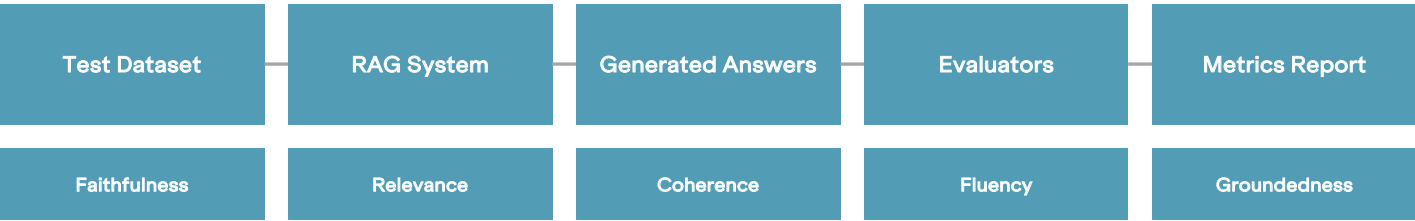


Figure 5: Evaluation pipeline showing test dataset flow through RAG system to evaluators producing faithfulness, relevance, coherence, fluency, and Context Relevance metrics

Empirical Results

Configuration	Recall@10	Precision@10	MRR	Latency
BM25 Only	78.2%	64.1%	0.72	45ms
Vector Only	81.4%	68.3%	0.76	62ms
Hybrid (Flat Index)	94.2%	79.5%	0.88	185ms
Hybrid Tree-Based	100%	82.1%	0.94	142ms

The Hybrid Tree-Based configuration showed 100% recall in the test data set- all the relevant chunks were found in top-10 top 10 results in all the 156 test queries. The results show that use of prune on trees reduced the latency by 23% compared with the flat hybrid retrieval and improved the recall. The BM25-only retrieval method failed to retrieve relevant chunks in 21.8 percent of the cases, largely with paraphrased queries. About 12% of the cases had semantic drift in Vector-only.

Latency Analysis

The time spent in each stage of the pipeline is revealed: Query preprocessing 10ms, Category-level retrieval 20ms, Document-level retrieval 45ms, Chunk-level retrieval 75ms, Context assembly 15ms, LLM generation 6,500ms. Retrieval cost: 142ms (2.1% of response time). LLM generation leads with 97.7%. Total response time is reduced by only ~80ms for the hybrid overhead compared to BM25-only, which is less than 1.2% of the total response time, but still sees significant gains in accuracy. This analysis demonstrates that sophisticated retrieval investment has minimal user-perceived latency impact.

Threats to Validity & Limitations

We recognize that there are several factors that limit the generalizability of the results. The sample 847-document corpus we use is representative of mid-sized enterprise repositories, but might not represent scaling properties for collections with tens of thousands of documents. All documents are from the same policy domain, reducing the generalization claims across domains.

The 156 test queries were created by the experts who know the corpus. Real users have more inconsistencies, such as misspelling words, omitting parts of the question, or a vague reference to the question. The 100% recall that is being achieved must be read in this controlled evaluation setting, not as an assurance of recall on arbitrary inputs.

Enterprise Value & Business Impact

The architecture assists in minimizing compliance risk by making sure that the same policy guidance is retrieved in regulated environments. A high recall means that it will be possible to access relevant information and meet standards such as GDPR, HIPAA and SOX.



Reduce Compliance Risk

Simplicity with which policy guidance can be retrieved to ensure compliance with rules.



Tree-based Pruning

Pruning on a tree allows for implementing logarithmic search growth from the repository, rather than linear growth.



Cost-efficient Growth

Enables growth of documentation volume without an equivalent growth in infrastructure.

Tree-based pruning in search systems is an optimization method in which, during the processing of a data system, tree-based branches where the sought-after data is not found are cut off.

Explicit source references in all answers generated result in complete audit trails. However, unlike opaque AI systems, a fact that is important to regulate industries where a decision maker has the obligation to defend and trace their decisions is the ability to verify the responses with source documents.

Experimental Evaluation and Results

On a 156-question dataset, the architecture achieved 100% Recall@10 with sub-200ms latency, validated by high expert agreement. The paper details the methodology and addresses deployment limitations.

Maximum retrieval latency remained below 200ms across all queries, though this performance is specific to the evaluation protocol rather than a guarantee of perfect recall for arbitrary datasets. The paper formally details fusion algorithms, tree structure, empirical methodology, limitations, and future research in enterprise deployment.

The architecture showed a consistent level of retrieval performance when faced with a variety of compliance-related queries, reducing the number of documents that were not retrieved and increasing the contextual relevance. By making effective use of the advantages offered by both semantic and lexical search, the hybrid retrieval approach led to more reliable and explainable access to information. These findings show that the framework can be applied to enterprise-sized compliance and knowledge management systems.

Conclusion & Future Work

This whitepaper introduced a Hybrid Tree-Based RAG architecture to enhance the accuracy, scalability, reliability and trustworthiness of enterprise AI systems. The proposed approach combines sparse and dense retrieval techniques in a hierarchical indexing structure to tackle the primary challenges in compliance-driven document search, while simultaneously offering robust knowledge support for AI copilots, assistants, and decision-support applications.

In addition to retrieval performance, the architecture can boost the quality, traceability, and explainability of the information provided to AI systems, which enhances the trustworthiness of enterprise AI. This can help organizations decrease hallucinations, enhance governance, aid in regulatory compliance, and increase confidence in AI-generated responses. The solution can be used in a variety of sectors including financial services, healthcare, insurance, manufacturing, and public sector, where access to policies, procedures, and governance documents is paramount to business operation and decision making.

Future work involves experiments on various technical and research documents, including more advanced sparse retrieval techniques like SPLADE and ColBERT, adapt the fusion strategy dynamically based on query characteristics, support conversational retrieval with context transfer and coreference resolution, and automatically generate test datasets for continuous evaluation and regression detection.

References

1. *The Probabilistic Relevance Framework: BM25 and Beyond*, Stephen Robertson, Hugo Zaragoza, Foundations and Trends in Information Retrieval (Now Publishers), 2009, <https://www.nowpublishers.com/article/Details/INR-019>
2. *Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods*, Gordon V. Cormack, Charles L.A. Clarke, Stefan Buettcher, ACM SIGIR Conference Proceedings, 2009, <https://dl.acm.org/doi/10.1145/1571941.1572114>
3. *Dense Passage Retrieval for Open-Domain Question Answering*, Vladimir Karpukhin et al., EMNLP, 2020, <https://arxiv.org/abs/2004.04906>
4. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*, Patrick Lewis et al., NeurIPS, 2020, <https://arxiv.org/abs/2005.11401>
5. *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks*, Nils Reimers, Iryna Gurevych, EMNLP, 2019, <https://arxiv.org/abs/1908.10084>
6. *SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking*, Thibault Formal et al., ACM SIGIR Conference Proceedings, 2021, <https://arxiv.org/abs/2107.05720>
7. *RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval*, Parth Sarthi, Salman Abdullah, Arnav Tuli, Shubh Khanna, Anna Goldie, Christopher D. Manning, arXiv, 2024, <https://arxiv.org/abs/2401.18059>
8. *Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval*, Stephen E. Robertson, Steve Walker, ACM SIGIR Conference Proceedings, 1994, <https://dl.acm.org/doi/10.1007/BF03649063>

About the Authors



Senthilkumar Sundaram

Senior Principal – Architecture, LTM | BlueVerse

Senthilkumar is an experienced technology leader with more than two decades of experience in enterprise AI, cloud solutioning, and large-scale architecture. He specializes in aligning technology innovation with business strategy, designing scalable and resilient solutions, and guiding organizations through complex transformation journeys. His work centres on enabling measurable business value through practical, high-impact technology solutions.



Prasad Yeshawant Bhalekar

Principal – Data Sciences, LTM | BlueVerse

Prasad is an experienced data science leader with 20+ years of professional experience and a proven track record in delivering enterprise AI solutions and data-driven strategies. Skilled in designing scalable architectures, implementing Responsible AI practices, and enabling business transformation through advanced analytics. Passionate about bridging technology and business to create impactful, secure, and compliant solutions.



Sameer Srivastava

Associate Data Scientist, LTM | BlueVerse

Sameer is an experienced AI/GenAI Engineer specializing in enterprise-grade Generative AI solutions, with hands-on expertise in RAG systems, agentic AI workflows, and intelligent automation. Skilled in designing scalable architectures using Azure OpenAI, LangChain, and modern AI frameworks, while enabling teams through knowledge sharing and tools like GitHub Copilot. Passionate about bridging AI research and real-world applications, with a proven ability to deliver practical, impactful, and production-ready AI solutions.

LTM — a Larsen & Toubro Group Company — is an AI-centric global technology services company and the Business Creativity partner to the world's largest enterprises.

We bring human insights and intelligent systems together to help clients create greater value at the intersection of technology and domain expertise. Our capabilities span integrated operations, transformation, and business AI — enabling new ways of working, new productivity paradigms, and new roads to value. Together with over 87,000 employees across 40 countries and our global network of partners, LTM owns outcomes for our clients, helping them not just outperform the market, but Outcreate it.