

HyPA-RAG: A Hybrid Parameter Adaptive Retrieval-Augmented Generation System for AI Legal and Policy Applications

Rishi Kalra², Zekun Wu^{1,2}, Ayesha Gulley¹, Airlie Hilliard¹,
Xin Guan¹, Adriano Koshiyama¹, Philip Treleaven²

¹Holistic AI, ²University College London

Correspondence: rishi.kalra.20@ucl.ac.uk, zekun.wu@holisticai.com

Abstract

While Large Language Models (LLMs) excel in text generation and question-answering, their effectiveness in AI legal and policy applications is limited by outdated knowledge, hallucinations, and inadequate reasoning in complex contexts. Retrieval-Augmented Generation (RAG) systems improve response accuracy by integrating external knowledge but struggle with retrieval errors, poor context integration, and high costs, particularly in interpreting AI legal texts. This paper introduces a Hybrid Parameter-Adaptive RAG (HyPA-RAG) system tailored for AI legal and policy, exemplified by NYC Local Law 144 (LL144). HyPA-RAG uses a query complexity classifier for adaptive parameter tuning, a hybrid retrieval strategy combining dense, sparse, and knowledge graph methods, and an evaluation framework with specific question types and metrics. By dynamically adjusting parameters, HyPA-RAG significantly improves retrieval accuracy and response fidelity. Testing on LL144 shows enhanced correctness, faithfulness, and contextual precision, addressing the need for adaptable NLP systems in complex, high-stakes AI legal and policy applications.¹

1 Introduction

The development of Large Language Models (LLMs) capable of processing and generating human-like text has made significant strides in recent years, such as OpenAI’s GPT models (Brown et al., 2020; OpenAI, 2023), Google’s Gemini models (Team et al., 2023) and open alternatives such as the LLaMa series (Touvron et al., 2023a,b; Meta, 2024). These models, which store vast amounts of information within their parameters through extensive pre-training, have demonstrated impressive performance in various tasks, including text generation and question-answering across multiple do-

mains (Brown et al., 2020; Singhal et al., 2023; Wu et al., 2023). Despite this, LLMs encounter limitations when applied to specialised fields such as law and policy. These include the rapid obsolescence of their knowledge, which is confined to the data available up to the last pre-training date (Yang et al., 2023), and hallucinations, where the model produces text that seems plausible but is factually incorrect or misleading, driven by internal logic rather than actual context (Ji et al., 2022; Huang et al., 2023). Empirical studies show that many artificial intelligence (AI) tools designed for legal applications overstate their ability to prevent hallucinations (Magesh et al., 2024). Indeed, instances of lawyers being penalised for using hallucinated outputs in court documents (Fortune, 2023; Business Insider, 2023) underscore the need for reliable AI question-answering systems in law and policy

Naturally, Retrieval-Augmented Generation (RAG), which enhances LLMs by incorporating external knowledge, is proposed as a solution. However, this comes with its own challenges. Common failure points (Barnett et al., 2024) include missing content, where relevant documents are not retrieved, leading to unanswered questions; context limitations, where retrieved documents are not effectively incorporated into the response generation process due to limitations in consolidation strategies; and extraction failures, where models fail to extract accurate information from the provided context due to noise or conflicting data. Furthermore, advanced retrieval and generation techniques, such as query rewriters and LLM-based quality checkers, often result in increased token usage and costs.

To address these challenges, this research integrates three key components (see Figure 6 in Appendix A.2 for a flow overview and Figure 1 for the system design):

¹The demo (Appendix A.1), dataset, and code are available at <https://github.com/holistic-ai/HyPA-RAG>.

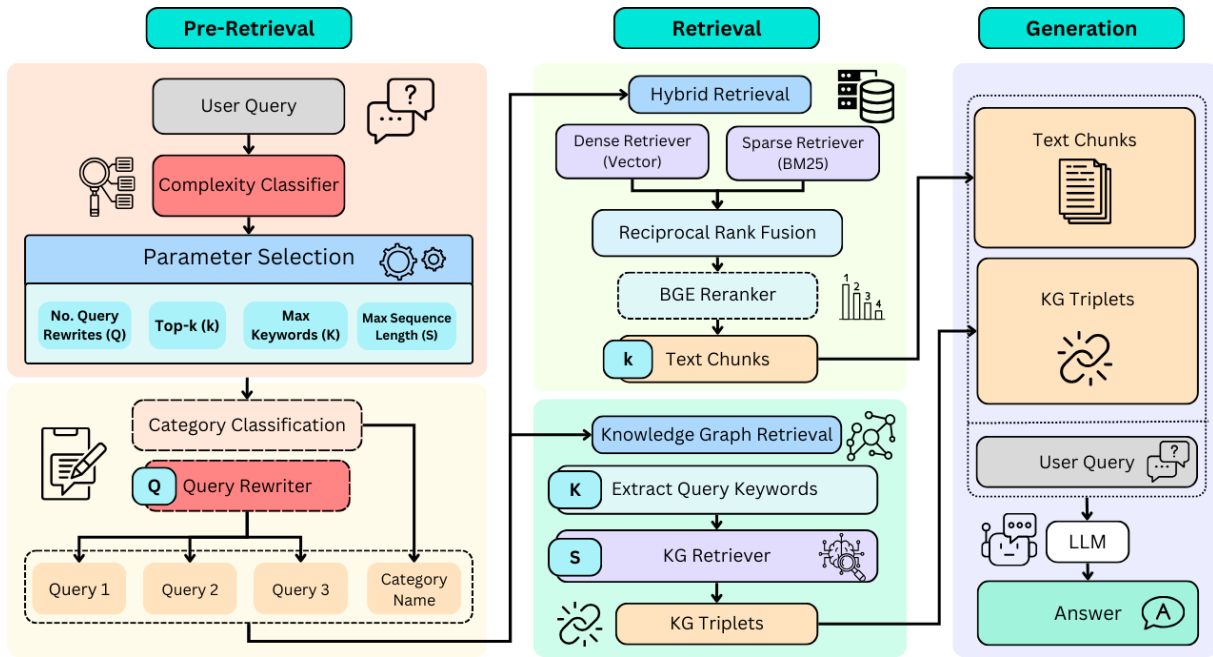


Figure 1: Hybrid Parameter Adaptive RAG (HyPA-RAG) System Diagram

- (1) Adaptive parameter selection using a domain-specific query complexity classifier to minimise unnecessary token usage,
- (2) A hybrid search system combining dense, sparse, and knowledge graph retrieval methods to enhance retrieval accuracy.
- (3) An end-to-end evaluation framework that includes the development of a 'gold standard' dataset, custom question types, and RAG-specific evaluation metrics for robust testing.

These elements are combined to create a hybrid parameter-adaptive RAG system tailored specifically to mitigate the common RAG failure points, for the AI policy domain, using NYC Local Law 144 as the primary corpus. We also provide a streamlit demo for testing purposes.

2 Background and Related Work

Recent LLM advancements have impacted fields like law and policy, where language complexity and large text volumes are prevalent (Blair-Stanek et al., 2023; Choi et al., 2023; Hargreaves, 2023). LLMs have been used for legal judgment prediction, document drafting, and contract analysis, showing their potential to improve efficiency and accuracy (Shui et al., 2023; Sun, 2023; Šavelka and Ashley, 2023). Techniques like fine-tuning, retrieval augmentation, prompt engineering, and agentic methods have adapted these models for specific legal

tasks, enhancing performance in summarisation, drafting, and interpretation (Trautmann et al., 2022; Cui et al., 2023).

Retrieval-Augmented Generation (RAG), as formalized by Lewis et al., enhances pre-trained seq2seq models by integrating external knowledge through indexing, retrieval, and generation stages, improving response specificity and accuracy (Lewis et al., 2020; Gao et al., 2023). RAG systems complement LLMs by combining sparse (e.g., BM25) and dense (e.g., vector) retrieval techniques, using neural embeddings to refine document retrieval and produce grounded, high-quality responses (Jones, 2021; Robertson and Zaragoza, 2009; Devlin et al., 2019; Liu et al., 2019).

To address the limitations of naive RAG, such as insufficient context and retrieval inaccuracies, advanced techniques have been developed, including hybrid retrieval methods, query rewriters, and rerankers to refine relevance (Muennighoff et al., 2022; Ding et al., 2024; Xiao et al., 2023). Hybrid retrieval combines BM25 with semantic embeddings to balance keyword matching and contextual understanding, improving outcomes (Luo et al., 2023; Ram et al., 2022; Arivazhagan et al., 2023). Additionally, knowledge graph retrieval and composed retrievers enhance accuracy and comprehensiveness in document retrieval (Rackauckas, 2024; Sanmartin, 2024; Edge et al., 2024).

Recently, RAG systems have advanced from ba-

sis retrieval to dynamic methods involving multi-source integration and domain adaptation (Gao et al., 2023; Ji et al., 2022). Innovations like Self-RAG and KG-RAG improve response quality and minimize hallucinations through adaptive retrieval and knowledge graphs (Asai et al., 2023; Sanmartin, 2024).

Various frameworks have been developed to evaluate RAG systems, including Ragas, which uses reference-free metrics like faithfulness and relevancy (Shahul et al., 2023b). Giskard (Giskard, 2023) assesses performance using synthetic QA datasets, while ARES utilizes prediction-powered inference (PPI) with specialized LLM judges for accurate evaluation (Giskard, 2023; Saad-Falcon et al., 2023).

3 System Design

The hybrid parameter-adaptive RAG system, depicted in Figure 1, integrates vector-based text chunks and a knowledge graph of entities and their relationships to enhance retrieval accuracy. The system employs a hybrid retrieval process, combining sparse (BM25) and dense (vector) methods to retrieve an initial top- k set of results. These results are refined using reciprocal rank fusion based on predefined parameter mappings.

Simultaneously, a knowledge graph retriever identifies relevant triplets, with retrieval depth and keyword selection dynamically adjusted according to query complexity. Results from both BM25 and vector methods are fused again to produce a final optimised set of k chunks.

Optional components include a query rewriter, which generates reformulated queries to improve retrieval. The rewritten queries fetch additional chunks, which are de-duplicated and fused to maintain uniqueness. An optional reranker can further refine chunk ranking if needed. The final set of selected chunks and knowledge graph triplets are then processed within the LLM’s context window for more accurate, contextually relevant responses.

This framework is implemented in two variations: without knowledge graph retrieval, known as Parameter-Adaptive (PA) RAG, and with knowledge graph retrieval, termed Hybrid Parameter-Adaptive (HyPA) RAG.

4 AI Legal and Policy Corpus

Local Law 144 (LL144) of 2021, enacted by the New York City Department of Consumer and

Worker Protection (DCWP), regulates automated employment decision tools (AEDTs). This paper uses a 15-page version of LL144 that combines the original law text with enforcement rules published by the DCWP. As an early AI-specific law, LL144 is included in the training data of foundational models like GPT-4 and GPT-4o, whose understanding of the law is confirmed manually through targeted prompting and serves as baselines in this research.

LL144 presents significant challenges for AI compliance due to its unique combination of qualitative and quantitative requirements. Unlike most AI legal and policy texts, which are predominantly qualitative, LL144 integrates detailed definitions and procedural guidelines with quantitative compliance metrics. This structure complicates interpretation and retrieval, often exceeding the capabilities of traditional LLMs and RAG systems. Furthermore, AI laws and policies are frequently revised while moving through the legislative process, making them impractical for pre-training and fine-tuning and therefore requiring a robust method for integrating changes.

5 Performance Evaluation

The evaluation process starts by generating custom questions tailored to AI policy and legal question-answering, then introduces and verifies evaluation metrics (see evaluation section of Figure 6 in appendix A.2). **For reproducibility, the LLM temperature is set to zero for consistent responses and all other parameters are set to defaults.**

5.1 Dataset Generation

Creating a "gold standard" evaluation set usually requires extensive human expertise and time, but LLMs like GPT-3.5-Turbo can efficiently handle such tasks, if sufficiently prompted. For this purpose, Giskard (Giskard, 2023) provides a library for synthetic data generation, using LLMs to create various question types from text chunks, such as 'simple', 'complex', and 'situational'. We introduce additional types and question generators: 'comparative', 'complex situational', 'vague', and 'rule-conclusion'. Comparative questions require multi-context retrieval to compare concepts. 'Complex situational' questions involve user-specific contexts and follow-ups. Vague questions obscure parts of the query to test interpretation, while rule-conclusion questions, adapted from LegalBench (Guha et al., 2023), require conclusions based on

legislative content. Table 4 in Appendix A.3 summarises these types with examples.

These question generators produce a set of questions, which are then deduplicated. Inaccurate or incomplete questions are identified through a human expert review process, using the criteria outlined in Table 5 in Appendix A.5.

5.2 Evaluation Metrics

To evaluate our RAG system, we utilise RAGAS metrics (Shahul et al., 2023a) based on the LLM-as-a-judge approach (Zheng et al., 2023), including Faithfulness, Answer Relevancy, Context Precision, Context Recall, and an adapted Correctness metric.

Faithfulness evaluates the factual consistency between the generated answer and the context, defined as Faithfulness Score = $\frac{|C_{\text{inferred}}|}{|C_{\text{total}}|}$, where C_{inferred} is the number of claims inferred from the context, and C_{total} is the total claims in the answer.

Answer Relevancy measures the alignment between the generated answer and the original question, calculated as the mean cosine similarity between the original question and generated questions from the answer: Answer Relevancy = $\frac{1}{N} \sum_{i=1}^N \frac{E_{g_i} \cdot E_o}{\|E_{g_i}\| \|E_o\|}$, where E_{g_i} and E_o are embeddings of the generated and original questions.

Context Recall measures the proportion of ground truth claims covered by the retrieved context, defined as Context Recall = $\frac{|C_{\text{attr}}|}{|C_{\text{GT}}|}$, where C_{attr} is the number of ground truth claims attributed to the context, and C_{GT} is the total number of ground truth claims.

Context Precision evaluates whether relevant items are ranked higher within the context, defined as Context Precision = $\frac{\sum_{k=1}^K (P_k \times v_k)}{|R_k|}$. Here, $P_k = \frac{TP_k}{TP_k + FP_k}$ is the precision at rank k , v_k is the relevance indicator, $|R_k|$ is the total relevant items in the top K , TP_k represents true positives, and FP_k false positives.

5.3 Correctness Evaluation

We assess correctness using a refined metric to address the limitations of Giskard’s binary classification, which fails to account for partially correct answers or minor variations. Our adapted metric, **Absolute Correctness**, based on LlamaIndex (LlamaIndex, 2024), uses a 1 to 5 scale: 1 indicates an incorrect answer, 3 denotes partial correctness, and 5 signifies full correctness. For binary evaluation, we use a high threshold of 4, reflecting our low tolerance for inaccuracies. The

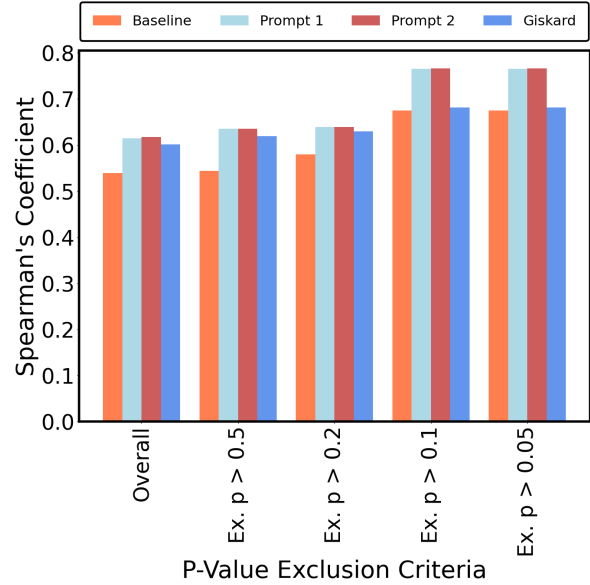


Figure 2: **Spearman Coefficient Comparison**, showing the correlation between model performance and human evaluation.

Correctness Score is computed as the average of these binary outcomes across all responses: Correctness Score = $\frac{1}{N} \sum_{i=1}^N \mathbb{1}(S_i \geq 4)$, where S_i represents the absolute correctness score of the i th response, $\mathbb{1}(S_i \geq 4)$ is an indicator function that is 1 if $S_i \geq 4$ and 0 otherwise, and N is the total number of responses.

The Spearman coefficient (Figure 2) illustrates how our prompt-based LLM-as-a-judge correctness evaluation aligns with human judgment. Prompts 1 and 2 (Appendix A.7) employ different methods: the baseline prompt provides general scoring guidelines, Prompt 1 offers detailed refinements, and Prompt 2 includes one-shot examples and guidance for edge cases.

Additional metrics, including macro precision, recall, F1 score, and percentage agreement with human labels, are shown in Figure 8 (Appendix A.8). A detailed breakdown of the Spearman coefficient metrics is provided in Figure 9 (Appendix A.8).

6 Chunking Method

We evaluate three chunking techniques: sentence-level, semantic, and pattern-based chunking.

Sentence-level chunking splits text at sentence boundaries, adhering to token limits and overlap constraints. Semantic chunking uses cosine similarity to set a dissimilarity threshold for splitting and includes a buffer size to define the minimum number of sentences before a split. Pattern-based

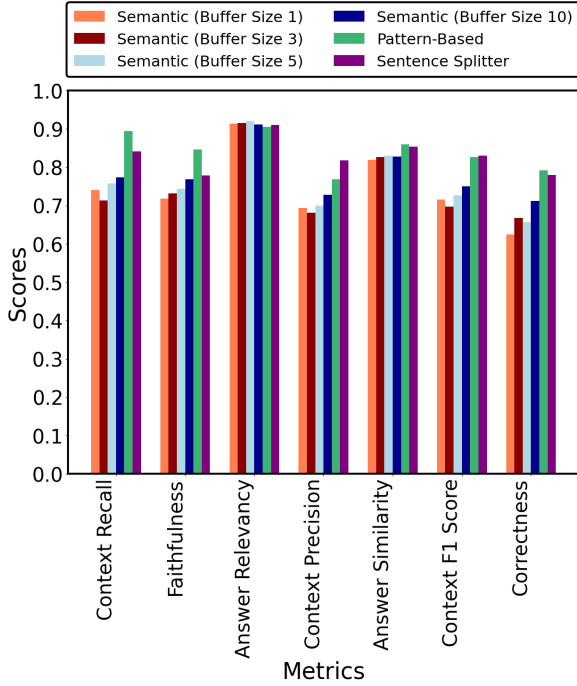


Figure 3: RAG Evaluation Metrics for Sentence-Level, Semantic, and Pattern-Based Chunking Methods

chunking employs a custom delimiter based on text structure; for LL144, this is "\n§".

Figure 3 shows that pattern-based chunking achieves the highest context recall (0.9046), faithfulness (0.8430), answer similarity (0.8621), and correctness (0.7918) scores. Sentence-level chunking, however, yields the highest context precision and F1 scores. Semantic chunking performs reasonably well with increased buffer size but generally underperforms compared to the simpler methods. Further hyperparameter tuning may improve its effectiveness. These findings suggest that a corpus-specific delimiter can enhance performance over standard chunking methods.

For subsequent experiments, we adopt sentence-level chunking with a default chunk size of 512 tokens and an overlap of 200 tokens.

7 Query Complexity Classifier

To enable adaptive parameter selection, we developed a domain-specific query complexity classifier that categorises user queries, each corresponding to specific hyper-parameter mappings. Our analysis of top- k selection indicated different optimal top- k values for various question types, as shown in Figure 7 (Appendix A.4).

Model	Precision	Recall	F1 Score
Random Labels	0.34	0.34	0.34
BART Large ZS	0.31	0.32	0.29
DeBERTa-v3 ZS	0.39	0.39	0.38
LR TF-IDF	0.84	0.84	0.84
SVM TF-IDF	0.86	0.86	0.86
distilBERT Finetuned	0.90	0.90	0.90

Table 1: 3-Class Classification Results

7.1 Training Data

To train a domain-specific query complexity classifier, we generated a dataset using a GPT-4o model on legal documents. Queries were categorised into three classes based on the number of contexts required: one context (0), two contexts (1), and three or more contexts (2). This classification resulted in varying token counts, keywords, and clauses across classes, which could bias models toward associating these features with complexity. To mitigate this, we applied data augmentation techniques to diversify the dataset. To enhance robustness, 67% of the queries were modified. We increased vagueness in 10% of the questions while preserving their informational content, added random noise words or punctuation to another 10%, and applied both word and punctuation noise to a further 10%. Additionally, 5% of questions had phrases reordered, and another 5% contained random spelling errors. For label-specific augmentation, 25% of label 0 queries were made more verbose, and 25% of label 2 queries were shortened, ensuring they retained the necessary informational content. The augmentation prompts are in Appendix A.9.

7.2 Model Training

We employed multiple models as baselines for classification tasks: Random labels, Logistic Regression (LR), Support Vector Machine (SVM), zero-shot classifiers, and a fine-tuned DistilBERT model. The Logistic Regression model used TF-IDF features, with a random state of 5 and 1000 iterations. The SVM model also used TF-IDF features with a linear kernel. Both models were evaluated on binary (2-class) and multi-class (3-class) tasks. Zero-shot classifiers (BART Large ZS and DeBERTa-v3 ZS) were included as additional baselines, mapping "simple question," "complex question," and "overview question" to labels 0, 1, and 2, respectively; for binary classification, only "simple question" (0) and "complex question" (1) were used. The DistilBERT model was fine-tuned with a learn-

Method	Faithfulness	Answer Relevancy	Absolute Correctness (1-5)	Correctness (Threshold=4.0)
LLM Only				
GPT-3.5-Turbo	0.2856	0.4350	2.6952	0.1973
GPT-4o-Mini	0.3463	0.6319	3.3494	0.4572
Fixed k				
$k = 3$	0.7748	0.7859	4.0372	0.7546
$k = 5$	0.8113	0.7836	4.0520	0.7584
$k = 7$	0.8215	0.7851	4.0520	0.7621
$k = 10$	0.8480	0.7917	4.0595	0.7658
Adaptive				
PA: k, Q (2 class)	0.9044	0.7910	<u>4.2491</u>	<u>0.8104</u>
PA: k, Q (3 class)	<u>0.8971</u>	0.7778	4.2528	0.8141
HyPA: k, Q, K, S (2 class)	0.8328	<u>0.7800</u>	4.0558	0.7770
HyPA: k, Q, K, S (3 class)	0.8465	0.7734	4.1338	0.7918

Table 2: Performance metrics for LLM Only, Fixed k , Parameter-Adaptive (PA), and Hybrid Parameter Adaptive (HyPA) RAG implementations for the 2 and 3-class classifier configurations. k is the top- k value, Q the number of query rewrites, S the maximum knowledge graph depth, and K the maximum keywords for knowledge graph retrieval.

ing rate of $2e-5$, batch size of 32, 10 epochs, and a weight decay of 0.01 to optimize performance and generalization to the validation set.

7.3 Classifier Results

Tables 1 and 8 in Appendix A.10 summarise the classification results. We compare performance using macro precision, recall and F1 score. The fine-tuned DistilBERT model achieved the highest F1 scores, 0.90 for the 3-class task and 0.92 for the 2-class task, highlighting the benefits of transfer learning and fine-tuning. The SVM (TF-IDF) and Logistic Regression models also performed well, particularly in binary classification, indicating their effectiveness in handling sparse data. Zero-shot classifiers performed lower, likely due to the lack of task-specific fine-tuning.

8 RAG System Architecture

8.1 Parameter-Adaptive RAG (PA-RAG)

The Parameter-Adaptive RAG system integrates our fine-tuned DistilBERT model to classify query complexity and dynamically adjusts retrieval parameters accordingly, as illustrated in Figure 1, but excluding the knowledge graph component. The PA-RAG system adaptively selects the number of query rewrites (Q) and the top- k value based on the complexity classification, with specific parameter mappings provided in Table 6 in Appendix

A.6.1. In the 2-class model, simpler queries (label 0) use a top- k of 5 and 3 query rewrites, while more complex queries (label 1) use a top- k of 10 and 5 rewrites. The 3-class model uses a top- k of 7 and 7 rewrites for the most complex queries (label 2).

8.2 Hybrid Parameter-Adaptive RAG

Building on the PA-RAG system, the Hybrid Parameter-Adaptive RAG (HyPA-RAG) approach enhances the retrieval stage by addressing issues such as missing content, incomplete answers, and failures of the language model to extract correct answers from retrieved contexts. These challenges often arise from unclear relationships within legal documents, where repeated terms lead to fragmented retrieval results (Barnett et al., 2024). Traditional (e.g. dense) retrieval methods may retrieve only partial context, causing missing critical information. To overcome these limitations, this system incorporates a knowledge graph (KG) representation of LL144. Knowledge graphs, structured with entities, relationships, and semantic descriptions, integrate information from multiple data sources (Hogan et al., 2020; Ji et al., 2020), and recent advancements suggest that combining KGs with LLMs can produce more informed outputs using KG triplets as added context.

The HyPA-RAG system uses the architecture outlined in Figure 1. The knowledge graph is con-

structed by extracting triplets (subject, predicate, object) from raw text using GPT-4o. Parameter mappings specific to this implementation, such as the maximum number of keywords per query (K) and maximum knowledge sequence length (S), are detailed in Table 7, extending those provided in Table 6.

8.3 RAG Results

The adaptive methods generally outperform the fixed k baseline across most metrics (Table 2). PA-RAG with k, Q (2 class) achieves the highest faithfulness score of 0.9044, which is an improvement of 0.0564 over the best fixed $k = 10$ method (0.8480). Similarly, the PA k, Q (3 class) configuration also performs strongly with a faithfulness score of 0.8971, surpassing all fixed k methods.

For answer relevancy, the PA k, Q (2 class) model achieves a score of 0.7910, which is nearly on par with the best fixed $k = 10$ method at 0.7917, showing a slight difference of 0.0007. The PA k, Q (3 class) model has a relevancy score of 0.7778, a drop of 0.0139 compared to the best fixed method.

In terms of absolute correctness, both PA models, k, Q (2 class) and k, Q (3 class), achieve scores of 4.2491 and 4.2528, respectively, which are improvements of approximately 0.1896 and 0.1933 over the best fixed method ($k = 10$) score of 4.0595. This suggests that adaptive parameter settings significantly enhance the model’s ability to provide correct answers.

Correctness scores also favour the adaptive methods. PA k, Q (3 class) model reaches a score of 0.8141, which is 0.0483 higher than the best fixed $k = 10$ score of 0.7658. PA k, Q (2 class) model shows similar strength with a score of 0.8104. HyPA show more varied results. HyPA k, Q, K, S (2 class) achieves a correctness score of 0.7770, a modest increase of 0.0112 over the fixed $k = 7$, suggesting that there is room for further optimisation.

8.4 System Ablation Study

We evaluate the impact of adaptive parameters, a reranker (bge-reranker-large), and a query rewriter on model performance using PA and HyPA RAG methods with 2-class (Table 9 in Appendix A.11) and 3-class classifiers (Table 3).

The highest Answer Relevancy (0.7940) is achieved by varying k alone, suggesting that simpler, focused responses facilitate question reconstruction. The k, Q + reranker configuration

achieves a slightly lower relevancy score (0.7902), indicating that query rewriting and reranking, while enhancing other metrics, introduce complexity that marginally reduces clarity.

The k, Q + reranker configuration also achieves the highest Faithfulness (0.9098), showing that combining adaptive top- k selection with query rewriting and reranking improves factual consistency. This setup provides high Absolute Correctness (4.2342), although slightly lower than k, Q alone (4.2528), indicating that while reranking improves response quality, it may slightly decrease overall accuracy. However, the Correctness Score improves from 0.8141 to 0.8178, highlighting an increase in responses classified as "correct" (scores of 4 or higher).

Adding a knowledge graph in the k, K, S configuration maintains the same Correctness Score (0.8141) as k, Q but reduces Absolute Correctness by 0.1301, suggesting added complexity might lower overall answer quality.

While the k, K, S, Q + reranker configuration does not lead in Faithfulness, Answer Relevancy, or Absolute Correctness, it achieves the highest Correctness Score (0.8402), outperforming k, Q + reranker by 0.0224, demonstrating the effectiveness of adaptive parameters and reranking in meeting the correctness threshold.

9 Overall Results and Discussion

Our analysis shows that adaptive methods generally outperform fixed baselines, particularly in improving faithfulness and answer quality. Incorporating adaptive parameters such as query rewrites and reranking enhances the system’s ability to provide accurate and relevant responses. While adding a reranker improves correctness, it can slightly reduce the overall correctness score, suggesting a trade-off between precision and answer quality.

The introduction of a knowledge graph maintains correctness but can add complexity, potentially lowering overall response quality. However, combining adaptive parameters with a reranker proves effective in maximizing the proportion of correct responses, even if it doesn’t lead to the highest scores in all metrics.

Overall, these findings highlight the importance of adaptivity and careful parameter tuning to balance different performance aspects, enhancing the system’s capability to handle varied and complex queries effectively.

Method	Faithfulness	Answer Relevancy	Absolute Correctness (1-5)	Correctness (Threshold=4.0)
k	0.7723	0.7940	4.0409	0.7621
k, Q	<u>0.8971</u>	0.7778	4.2528	0.8141
k, Q + reranker	0.9098	<u>0.7902</u>	<u>4.2342</u>	<u>0.8178</u>
k, K^*, S^*	0.8733	0.7635	4.1227	0.8141
k, K, S	0.8660	0.7780	4.1822	0.8030
k, K, S + reranker	0.8821	0.7872	4.1858	<u>0.8178</u>
k, K, S, Q	0.8465	0.7734	4.1338	0.7918
k, K, S, Q + reranker	0.8689	0.7853	4.1859	0.8402

Table 3: Ablation study results for different configurations of adaptive k in a 3-class setting. For descriptions of parameters, refer to Table 2. The highest value in each column is highlighted in bold, and the second highest value is underlined. The * indicates parameters held fixed, rather than adaptive.

10 Limitations and Future Work

This study has several limitations that suggest areas for future improvement. Correctness evaluation is limited by reliance on a single evaluator familiar with the policy corpus. Averaging a larger quantity of human evaluations would improve reliability. Additionally, our knowledge graph construction process may be improved. For instance, using LLM-based methods for de-duplication and/or custom Cypher query generation to improve context retrieval and precision. Furthermore, our parameter mappings were not rigorously validated quantitatively. Further evaluation of parameter selections could provide better mappings as well as upper and lower bounds to performance. The classifier was trained using domain-specific synthetically generated data - which, though we inject significant noise, may harbour the LLM’s own unconscious biases in terms of structure - possibly limiting the generalisability of the classifier on unseen user queries. Also, more classification categories e.g. 4 or 5-class, would permit more granular parameter selections and potentially greater efficiency improvements. Another limitation is that while LL144 is included in the GPT models’ training data, subsequent minor revisions may affect the accuracy of these baseline methods.

Integrating human feedback into the evaluation loop (see Figure 4) could better align metrics with user preferences and validate performance metrics in real-world settings. Future work should also consider fine-tuning the LLM using techniques like RLHF (Bai et al., 2022), RLAIIF (Lee et al., 2023), or other preference optimisation methods (Song et al., 2023). Further, refining the query rewriter (Ma et al., 2023; Mao et al., 2024) and exploring

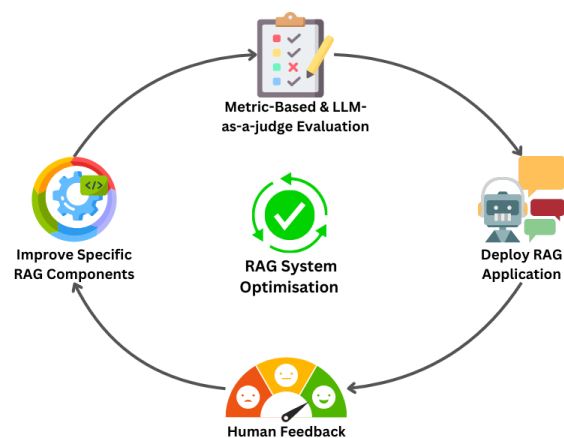


Figure 4: RAG System Optimisation Feedback Loop

iterative answer refinement (Asai et al., 2023) could enhance metrics like relevancy and correctness.

11 Ethical and Societal implications

The deployment of the Hybrid Parameter-Adaptive RAG (HyPA-RAG) system in AI legal and policy contexts raises critical ethical and societal concerns, particularly regarding the accuracy, reliability, and potential misinterpretation of AI-generated responses. The high-stakes nature of legal information means inaccuracies could have significant consequences, highlighting the necessity for careful evaluation. We emphasize transparency and reproducibility, providing detailed documentation of data generation, retrieval methods, and evaluation metrics to facilitate replication and scrutiny. The environmental impact of NLP models is also a concern. Our system employs adaptive retrieval strategies to optimize computational efficiency, reduce energy consumption, and minimize carbon footprint, promoting sustainable AI development.

Our findings enhance the understanding of RAG systems in legal contexts but are intended for research purposes only. HyPA-RAG outputs should not be used for legal advice or decision-making, emphasizing the need for domain expertise and oversight in applying AI to sensitive legal domains.

12 Acknowledgments

We would like to express our sincere gratitude to Holistic AI for their invaluable research and financial support, which made this project possible. Our deepest appreciation also goes to the Center for Artificial Intelligence at University College London for their continuous encouragement and assistance. Without their collaboration and resources, this work would not have been achievable.

References

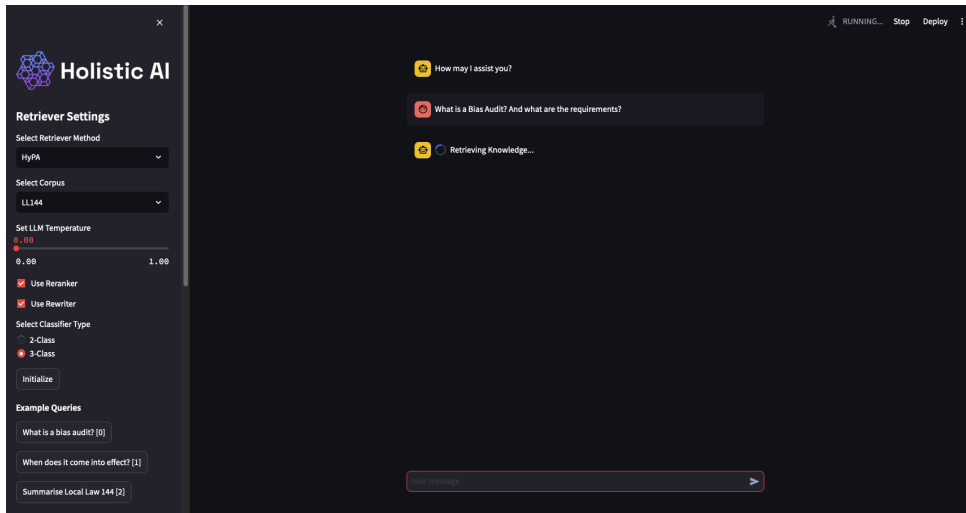
- Manoj Ghuhana Arivazhagan, Lan Liu, Peng Qi, Xinchu Chen, William Yang Wang, and Zhiheng Huang. 2023. [Hybrid hierarchical retrieval for open-domain question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2023*.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). *ArXiv*.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova Dassarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, Benjamin Mann, and Jared Kaplan. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *ArXiv*, abs/2204.05862.
- Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. [Seven failure points when engineering a retrieval augmented generation system](#). *2024 IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pages 194–199.
- Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. [Can gpt-3 perform statutory reasoning?](#) *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *ArXiv*, abs/2005.14165.
- Business Insider. 2023. [Michael cohen used ai chatbot to find bogus legal cases](#). Accessed: 2024-06-10.
- Jonathan H. Choi, Kristin E. Hickman, Amy B. Monahan, and Daniel Benjamin Schwarcz. 2023. [Chatgpt goes to law school](#). *SSRN Electronic Journal*.
- Jiaxi Cui, Zongjia Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. [Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). In *North American Chapter of the Association for Computational Linguistics*.
- Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. [Hybrid llm: Cost-efficient and quality-aware query routing](#). *ArXiv*, abs/2404.14618.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. [From local to global: A graph rag approach to query-focused summarization](#). *ArXiv*, abs/2404.16130.
- Fortune. 2023. [Lawyers fined for filing chatgpt hallucinations in court](#). Accessed: 2024-06-10.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. [Retrieval-augmented generation for large language models: A survey](#). *ArXiv*, abs/2312.10997.
- Giskard. 2023. Giskard: Automated quality manager for llms. <https://www.giskard.ai/>. Accessed: 2024-08-19.
- Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin M. K. Peters, Brandon Waldon, Daniel N. Rockmore, Diego A. Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John J. Nay, Jonathan H. Choi, Kevin Patrick Tobia, Margaret Hagan, Megan Ma, Michael A. Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shangsheng Gao, Spencer Williams, Sunny G. Gandhi, Tomer Zur, Varun J. Iyer, and Zehua Li. 2023. [Legalbench: A collaboratively built benchmark for](#)

- measuring legal reasoning in large language models. *ArXiv*, abs/2308.11462.
- Stuart Hargreaves. 2023. ‘words are flowing out like endless rain into a paper cup’: Chatgpt & law school assessments. *SSRN Electronic Journal*.
- Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, S. Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2020. *Knowledge graphs*. *ACM Computing Surveys (CSUR)*, 54:1 – 37.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. *A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions*. *ArXiv*, abs/2311.05232.
- Shaoxiong Ji, Shirui Pan, E. Cambria, Pekka Marttinen, and Philip S. Yu. 2020. *A survey on knowledge graphs: Representation, acquisition, and applications*. *IEEE Transactions on Neural Networks and Learning Systems*, 33:494–514.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. *Survey of hallucination in natural language generation*. *ACM Computing Surveys*, 55:1 – 38.
- Karen Spärck Jones. 2021. *A statistical interpretation of term specificity and its application in retrieval*. *J. Documentation*, 60:493–502.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbone, and Abhinav Rastogi. 2023. *Rlaif: Scaling reinforcement learning from human feedback with ai feedback*. *ArXiv*, abs/2309.00267.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. *Retrieval-augmented generation for knowledge-intensive nlp tasks*. *ArXiv*, abs/2005.11401.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. *Roberta: A robustly optimized bert pretraining approach*. *ArXiv*, abs/1907.11692.
- LlamaIndex. 2024. *Llamaindex*. Accessed: August 19, 2024.
- Man Luo, Shashank Jain, Anchit Gupta, Arash Einolghozati, Barlas Oguz, Debojeet Chatterjee, Xilun Chen, Chitta Baral, and Peyman Heidari. 2023. *A study on the efficiency and generalization of light hybrid retrievers*. *ArXiv*, abs/2210.01371.
- Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. *Query rewriting for retrieval-augmented large language models*. *ArXiv*, abs/2305.14283.
- Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2024. *Hallucination-free? assessing the reliability of leading ai legal research tools*.
- Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. *Rafe: Ranking feedback improves query rewriting for rag*. *ArXiv*, abs/2405.14431.
- Meta. 2024. *The llama 3 herd of models*.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2022. *Mteb: Massive text embedding benchmark*. *arXiv preprint arXiv:2210.07316*.
- OpenAI. 2023. *Gpt-4 technical report*.
- Zackary Rackauckas. 2024. *Rag-fusion: a new take on retrieval-augmented generation*. *ArXiv*, abs/2402.03367.
- Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2022. *Learning to retrieve passages without supervision*. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Stephen E. Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: Bm25 and beyond*. *Found. Trends Inf. Retr.*, 3:333–389.
- Jon Saad-Falcon, O. Khatib, Christopher Potts, and Matei Zaharia. 2023. *Ares: An automated evaluation framework for retrieval-augmented generation systems*. *ArXiv*, abs/2311.09476.
- Diego Sanmartin. 2024. *Kg-rag: Bridging the gap between knowledge and creativity*. *ArXiv*, abs/2405.12035.
- Jaromír Šavelka and Kevin D. Ashley. 2023. *The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts*. *Frontiers in Artificial Intelligence*, 6.
- ES Shahul, Jithin James, Luis Espinosa Anke, and S. Schockaert. 2023a. *Ragas: Automated evaluation of retrieval augmented generation*. *ArXiv*.
- ES Shahul, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2023b. *Ragas: Automated evaluation of retrieval augmented generation*. In *Conference of the European Chapter of the Association for Computational Linguistics*.
- Ruihao Shui, Yixin Cao, Xiang Wang, and Tat-Seng Chua. 2023. *A comprehensive evaluation of large language models on legal judgment prediction*. *ArXiv*, abs/2310.11761:7337–7348.

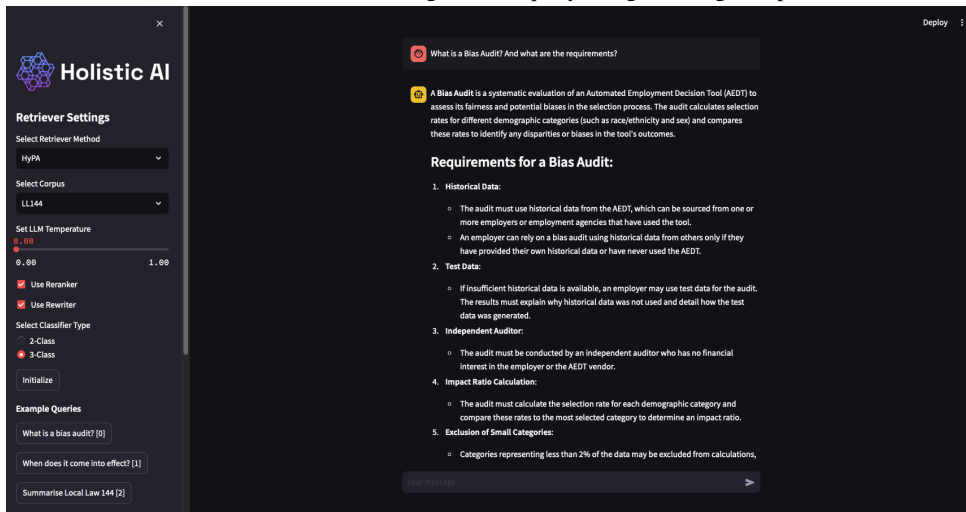
- K. Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen R. Pfohl, Heather J. Cole-Lewis, Darlene Neal, Mike Schaeckermann, Amy Wang, Mohamed Amin, S. Lachgar, P. A. Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Agüera y Arcas, Nenad Tomašev, Yun Liu, Renee C. Wong, Christopher Semturs, Seyedeh Sara Mahdavi, Joëlle K. Barral, Dale R. Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. [Towards expert-level medical question answering with large language models](#). *ArXiv*, abs/2305.09617.
- Feifan Song, Yu Bowen, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. [Preference ranking optimization for human alignment](#). *ArXiv*, abs/2306.17492.
- ZhongXiang Sun. 2023. [A short survey of viewing large language models in legal aspect](#). *ArXiv*, abs/2303.09136.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *ArXiv*, abs/2302.13971.
- Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *ArXiv*, abs/2307.09288.
- Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. [Legal prompt engineering for multilingual legal judgement prediction](#). *ArXiv*, abs/2212.02199.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. [Bloomberggpt: A large language model for finance](#). *ArXiv*, abs/2303.17564.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. [C-pack: Packaged resources to advance general chinese embedding](#). *Preprint*, arXiv:2309.07597.
- Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. [Harnessing the power of llms in practice: A survey on chatgpt and beyond](#). *ACM Transactions on Knowledge Discovery from Data*, 18:1 – 32.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *ArXiv*, abs/2306.05685.

A Appendix

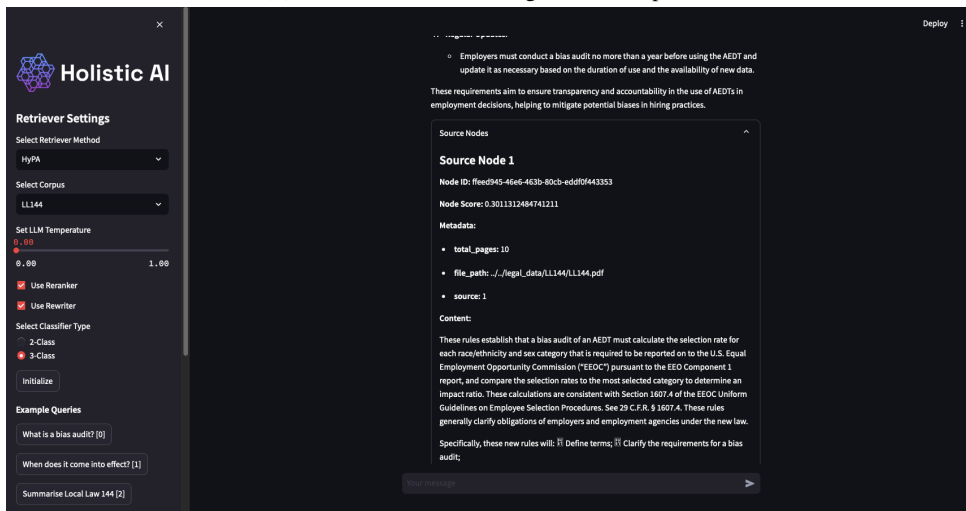
A.1 RAG Demonstration User Interface



(a) Demo Screenshot: Entering the user query and generating a response.



(b) Demo Screenshot: The generated response.



(c) Demo Screenshot: Information on retrieved node metadata and content.

Figure 5: Demo screenshots showing each key stage of the user experience.

A.2 Overall Workflow Diagram

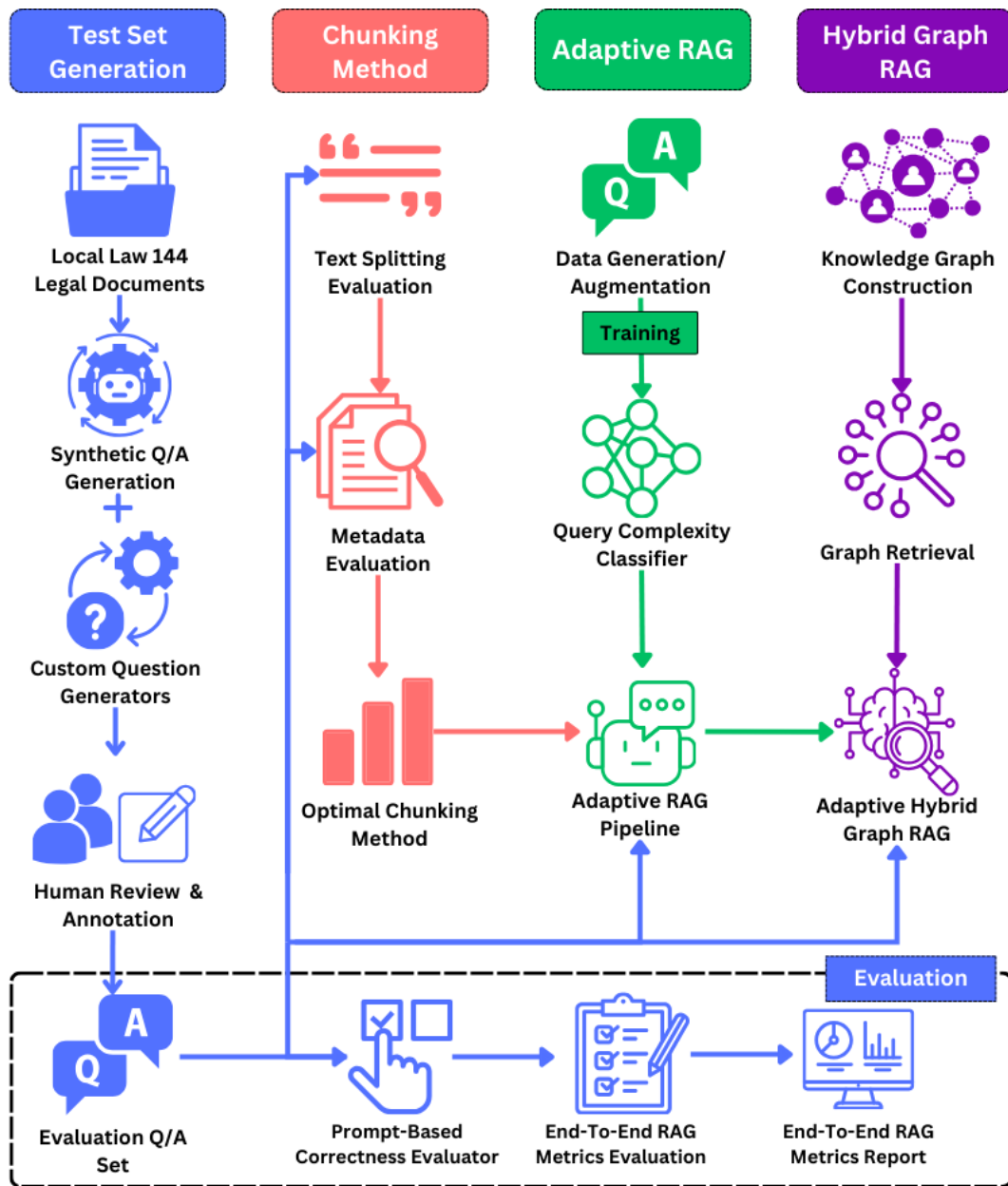


Figure 6: Overall RAG Development Workflow Diagram

A.3 Question Types

Question Type	Description	Example Question	Target RAG Components
Simple	Requires retrieval of one concept from the context	What is a bias audit?	Generator, Retriever, Router
Complex	More detailed and requires more specific retrieval	What is the purpose of a bias audit for automated employment decision tools?	Generator, Retriever
Distracting	Includes an irrelevant distracting element	Italy is beautiful but what is a bias audit?	Generator, Retriever, Rewriter
Situational	Includes user context to produce relevant answers	As an employer, what information do I need to provide before using an AEDT?	Generator
Double	Two distinct parts to evaluate query rewriter	What are the requirements for a bias audit of an AEDT and what changes were made in the second version of the proposed rules?	Generator, Rewriter
Conversational	Part of a conversation with context provided in a previous message	(1) I would like to know about bias audits. (2) What is it?	Rewriter
Complex situational	Introduces further context and one or more follow-up questions within the same message	In case I need to recover a civil penalty, what are the specific agencies within the office of administrative trials and hearings where the proceeding can be returned to? Also, are there other courts where such a proceeding can be initiated?	Generator
Out of scope	Non-answerable question that should be rejected	Who developed the AEDT software?	Generator, Prompt
Vague	A vague question that lacks complete information to answer fully	What calculations are required?	Generator, Rewriter
Comparative	Encourages comparison and identifying relationships	What are the differences and similarities between 'selection rate' and 'scoring rate', and how do they relate to each other?	Generator, Rewriter
Rule conclusion	Provides a scenario, requiring a legal conclusion	An employer uses an AEDT to screen candidates for a job opening. Is the selection rate calculated based on the number of candidates who applied for the position or the number of candidates who were screened by the AEDT?	Generator, Rewriter

Table 4: Question types and their descriptions with targeted RAG components.

A.4 Evaluation Results for Varied Top- k

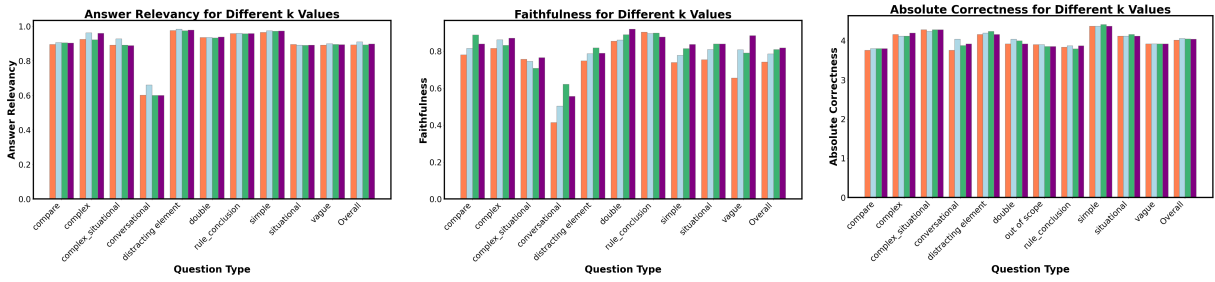


Figure 7: RAG Evaluation Metrics for Varied Top- k

A.5 Human Annotation Criteria

No.	Criterion	Description
1	Faithfulness	Are all claims in the answer inferred from the context?
2	Answer Relevancy	Is the answer relevant to the question?
3	Context Relevancy	Is the context relevant to the question?
4	Correctness	Is the answer correct, given the context?
5	Clarity	Is the answer clear and free of extensive jargon?
6	Completeness	Does the answer fully address all parts and sub-questions?

Table 5: Criteria for evaluating the quality of QA pairs.

A.6 Parameter Mappings

A.6.1 Top- k (k) and Number of Query Rewrites (Q)

Parameter	Symbol	Description	2-Class Mappings	3-Class Mappings
Number of Query Rewrites	Q	Number of sub-queries generated for the original query	0: $Q = 3$	0: $Q = 3$
			1: $Q = 5$	1: $Q = 5$
				2: $Q = 7$
Top- k Value	k	Number of top documents or contexts retrieved for processing	0: $k = 5$	0: $k = 3$
			1: $k = 10$	1: $k = 5$
				2: $k = 7$

Table 6: Parameter Symbols, Descriptions, and Mappings

A.6.2 Maximum Keywords (K) and Maximum Sequence Length (S)

Parameter	Symbol	Description	2-Class Mappings	3-Class Mappings
Max Keywords per Query	K	Maximum number of keywords used per query for KG retrieval	0: $K = 4$	0: $K = 3$
			1: $K = 5$	1: $K = 4$ 2: $K = 5$
Max Knowledge Sequence	S	Maximum sequence length for knowledge graph paths	0: $S = 2$	0: $S = 1$
			1: $S = 3$	1: $S = 2$ 2: $S = 3$

Table 7: Parameter Symbols, Descriptions, and Mappings (Part 2)

A.7 Correctness Evaluator Prompts

A.7.1 Method 1: LLamaIndex CorrectnessEvaluator

You are an expert evaluation system for a question answering

chatbot. You are given the following information:

- a user query,
- a reference answer, and
- a generated answer.

Your job is to judge the relevance and correctness of the generated answer. Output a single score that represents a holistic evaluation. You must return your response in a line with only the score. Do not return answers in any other format. On a separate line, provide your reasoning for the score as well.

Follow these guidelines for scoring:

- Your score has to be between 1 and 5, where 1 is the worst and 5 is the best.
- If the generated answer is not relevant to the user query, give a score of 1.
- If the generated answer is relevant but contains mistakes, give a score between 2 and 3.
- If the generated answer is relevant and fully correct, give a score between 4 and 5.

A.7.2 Method 2: Custom Prompt 1

You are an expert evaluation system for a question answering

chatbot. You are given the following information:

- a user query,
- a reference answer, and
- a generated answer.

Your job is to judge the correctness of the generated answer. Output a single score that represents a holistic evaluation. You must return your response in a line with only the score. Do not return answers in any other format. On a separate line, provide your reasoning for the score as well.

Follow these guidelines for scoring:

- Your score has to be between 1 and 5, where 1 is the worst and 5 is the best.
 - Use the following criteria for scoring correctness:
1. Score of 1:
 - The generated answer is completely incorrect.

- Contains major factual errors or misconceptions.
- Does not address any components of the user query correctly.

2. Score of 2:

- The generated answer has significant mistakes.
- Addresses at least one component of the user query correctly but has major errors in other parts.

3. Score of 3:

- The generated answer is partially correct.
- Addresses multiple components of the user query correctly but includes some incorrect information.
- Minor factual errors are present.

4. Score of 4:

- The generated answer is mostly correct.
- Correctly addresses all components of the user query with minimal errors.
- Errors do not substantially affect the overall correctness.

5. Score of 5:

- The generated answer is completely correct.
- Addresses all components of the user query correctly without any errors.
- The answer is factually accurate and aligns perfectly with the reference answer.

A.7.3 Method 3: Custom Prompt 2

You are an expert evaluation system for a question answering

chatbot. You are given the following information:

- a user query,
- a reference answer, and
- a generated answer.

Your job is to judge the correctness of the generated answer. Output a single score that represents a holistic evaluation. You must return your response in a line with only the score. Do not return answers in any other format. On a separate line, provide your reasoning for the score as well. The reasoning must not exceed one sentence.

Follow these guidelines for scoring:

- Your score has to be between 1 and 5, where 1 is the worst and 5 is the best.
- Use the following criteria for scoring correctness:

1. Score of 1:

- The generated answer is completely incorrect.
- Contains major factual errors or misconceptions.
- Does not address any components of the user query correctly.

- Example:
Query: "What is the capital of France?"
Generated Answer: "The capital of France is Berlin."

- If the answer provides more information than necessary, it should not be penalized as long as all information is correct.

2. Score of 2:

- Significant mistakes are present.
- Addresses at least one component of the user query correctly but has major errors in other parts.
- Example:
Query: "What is the capital of France and its population?"
Generated Answer: "The capital of France is Paris, and its population is 100 million."

3. Score of 3:

- Partially correct with some incorrect information.
- Addresses multiple components of the user query correctly.
- Minor factual errors are present.
- Example:
Query: "What is the capital of France and its population?"
Generated Answer: "The capital of France is Paris, and its population is around 3 million."

4. Score of 4:

- Mostly correct with minimal errors.
- Correctly addresses all components of the user query.
- Errors do not substantially affect the overall correctness.
- Example:
Query: "What is the capital of France and its population?"
Generated Answer: "The capital of France is Paris, and its population is approximately 2.1 million."

5. Score of 5:

- Completely correct.
- Addresses all components of the user query correctly without any errors.
- Providing more information than necessary should not be penalized as long as all provided information is correct.
- Example:
Query: "What is the capital of France and its population?"
Generated Answer: "The capital of France is Paris, and its population is approximately 2.1 million. Paris is known for its rich history and iconic landmarks such as the Eiffel Tower and Notre-Dame Cathedral."

Checklist for Evaluation:

- Component Coverage: Does the answer cover all parts of the query?
- Factual Accuracy: Are the facts presented in the answer correct?
- Error Severity: How severe are any errors present in the answer?
- Comparison to Reference: How closely does the answer align with the reference answer?

Edge Cases:

- If the answer includes both correct and completely irrelevant information, focus only on the relevant portions for scoring.
- If the answer is correct but incomplete, score based on the completeness criteria within the relevant score range.

A.8 Correctness Evaluator Results

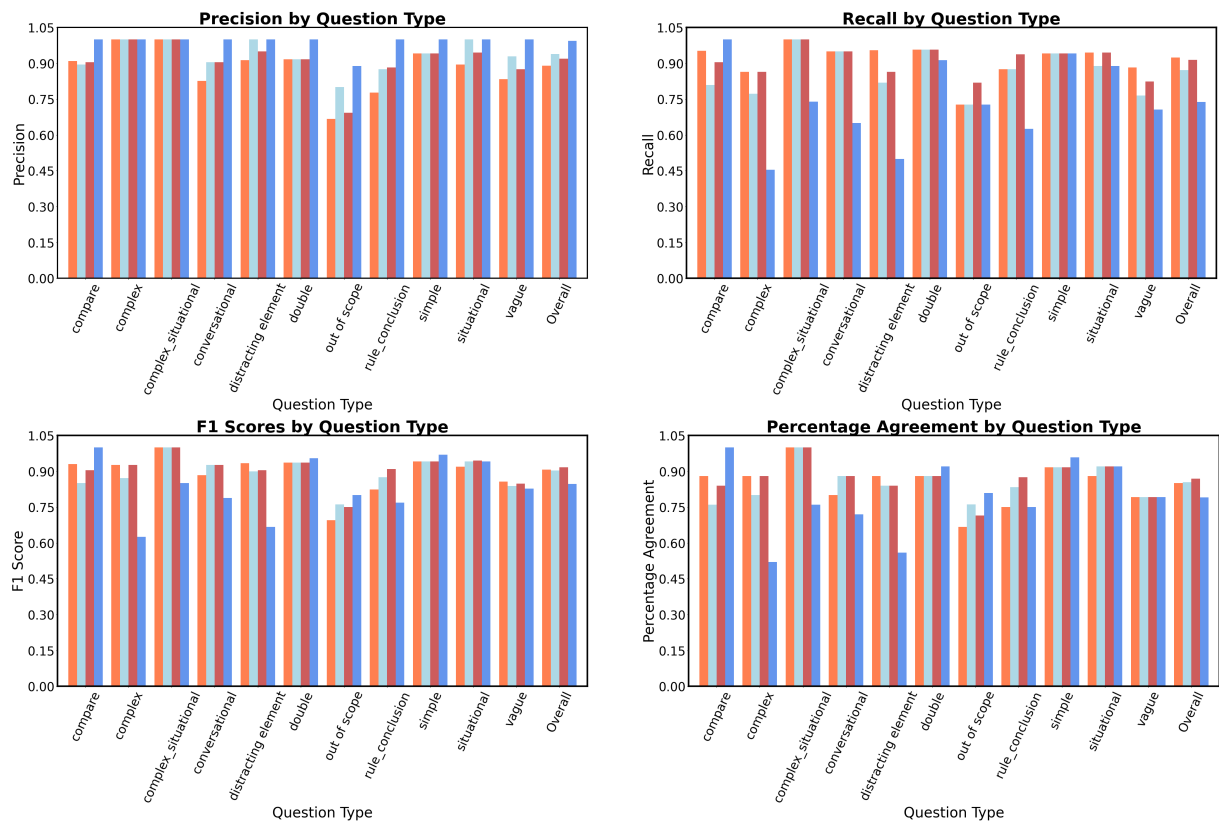


Figure 8: Precision, recall, F1 score, and percentage agreement of the prompt-based (1-5 scale) LLM-as-a-judge correctness evaluation compared to human judgments.

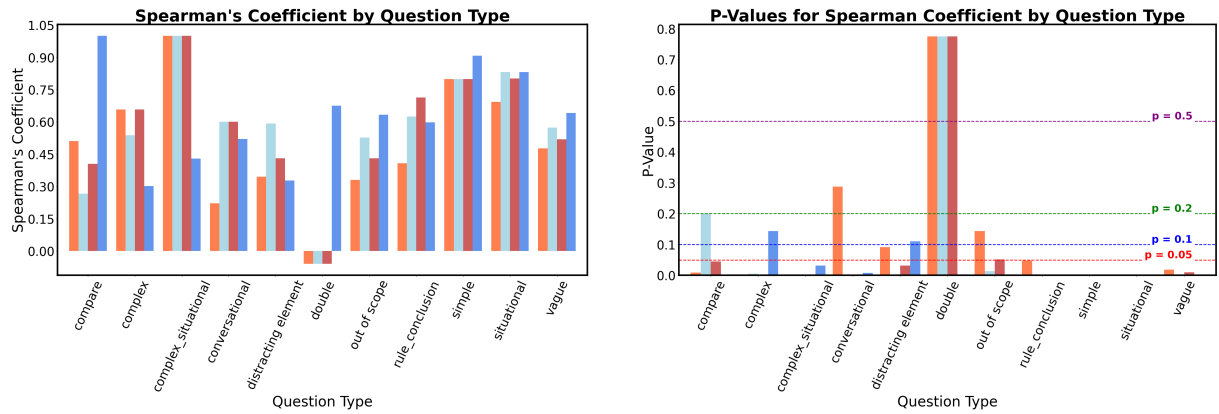


Figure 9: Spearman Coefficient comparing our custom LLM-as-a-judge (1-5 scale) prompts with Giskard's binary correctness evaluator for each question type. The second plot displays the p-values.

A.9 Classifier Data Augmentation Prompts

A.9.1 Vague Prompt

Rewrite the following question to be more vague, but it must still require the same number of pieces of information to answer. For example, a definition is one piece of information. A definition and an explanation of the concept are two separate pieces of information. Do not add or remove any pieces of information, and do not alter the fundamental meaning of the question. Output only the rewritten question, absolutely nothing else: {question}

A.9.2 Verbose Prompt

Rewrite the following question to be more verbose, but it must still require the same number of pieces of information to answer. For example, a definition is one piece of information. A definition and an explanation of the concept are two separate pieces of information. Do not add or remove any pieces of information, and do not alter the fundamental meaning of the question. Output only the rewritten question, absolutely nothing else: {question}

A.9.3 Concise Prompt

Rewrite the following question to be more concise, but it must still require the same number of pieces of information to answer. For example, a definition is one piece of information. A definition and an explanation of the concept are two separate pieces of information. Do not add or remove any pieces of information, and do not alter the fundamental meaning of the question. Output only the rewritten question, absolutely nothing else: {question}

A.10 2-Class Classifier Results

Model	Precision	Recall	F1 Score
Random Labels	0.49	0.49	0.49
facebook/bart-large-mnli	0.55	0.55	0.53
DeBERTa-v3-base-mnli-fever-anli	0.59	0.57	0.56
Logistic Regression (TF-IDF)	0.88	0.88	0.88
SVM (TF-IDF)	0.92	0.92	0.92
distilbert-base-uncased finetuned	0.92	0.92	0.92

Table 8: 2-Class Classification Results

A.11 2-Class Ablation Results

Method	Faithfulness	Answer Relevancy	Absolute Correctness (1-5)	Correctness (Threshold=4.0)
k	0.8111	0.7835	4.0372	0.7546
k, K^*, S^*	0.8725	<u>0.7830</u>	4.1115	<u>0.8216</u>
k, K, S	0.8551	0.7810	4.1487	0.7955
k, K, S + reranker	0.8792	0.7878	4.1710	0.8141
k, K, S + adaptive Q	0.8328	0.7800	4.0558	0.7770
k, K, S + Q + reranker	<u>0.8765</u>	0.7803	<u>4.1636</u>	0.8253

Table 9: Ablation study results for different configurations starting from adaptive k . The highest value in each column is highlighted in bold, and the second highest value is underlined.