International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

An Advanced RAG - Based Pipeline for Precise Legal Information Retrieval

Sarath Babu Poovassery Krishnan

Senior Solutions Architect, Amazon Web Services, New Jersey

Abstract: ***Efficient retrieval of relevant information from large volumes of legal documents is both critical and challenging in the legal domain. Large language models (LLMs) offer a transformative opportunity to enhance the efficiency of legal research by enabling natural language - based interfaces for precise information retrieval. Among emerging techniques, Retrieval - Augmented Generation (RAG) has gained prominence as a powerful tool for information retrieval. However, basic RAG methods often fall short of meeting the specific needs of legal professionals in identifying accurate and contextually relevant information. This paper examines the limitations of existing RAG pipelines and proposes an enhanced RAG pipeline tailored for legal information retrieval, designed to improve both accuracy and relevancy. Our approach integrates a multi - layered chunking strategy, enhanced metadata annotations, a hybrid search mechanism that combines sparse and dense vector embeddings within a vector database. To refine query understanding, we employ query expansion techniques and dynamically apply metadata filters. Implementing these approaches can significantly enhance retrieval quality, thereby improving the overall effectiveness of legal information retrieval.***

Keywords: Legal Information Retrieval, Retrieval - Augmented Generation (RAG), Large Language Models (LLM), Legal Research Automation, Hybrid Search, Multi - layered Chunking, Query Expansion, Metadata Annotation, Embedding Fine - tuning, Semantic Search, Re - ranking Algorithms, Answer Faithfulness, Context Relevance, Case Law Retrieval

1. Introduction

Legal research is a critical yet time - intensive aspect of legal practice. Legal research is the process of finding information that is needed to support legal decision - making. In practice, this generally means searching through both statute (as created by the legislature) and case law (as developed by the courts) to find what is relevant for a specific matter at hand [5]. It involves analyzing statutory provisions, judicial precedents, and secondary sources such as legal commentaries to synthesize a clear understanding of the law as it applies to a particular case. Lawyers face significant challenges during legal research, often needing to sift through hundreds of pages of statutes, case opinions, and supporting documents to identify relevant information. This process is further complicated by the complexity of interactions between statutes and case law, and the constant evolution of regulations and judicial interpretations. Despite these hurdles, thorough legal research remains essential for crafting persuasive arguments, ensuring compliance, and providing precise, evidence - based advice to clients, making it a cornerstone of effective legal practice.

In this paper, we will explore the limitations of basic Retrieval - Augmented Generation (RAG) [18] methods, highlighting their shortcomings in the context of legal information retrieval. We will delve into advanced RAG techniques that have the potential to overcome these limitations and provide more accurate and contextually relevant results for legal professionals. To demonstrate the effectiveness of these approaches, we will conduct a comparative analysis using judgments from the Supreme Court of India. This analysis will showcase how the enhanced RAG pipeline outperforms the naive RAG methods in terms of precision and relevance.

2. Research Background or Literature

Since ChatGPT launched in November 2022 [26], Generative AI has made a significant leap forward, with Large Language Models (LLMs) [25] becoming powerful tools for boosting productivity in various fields, including law. Despite their impressive capabilities, these models have a major flaw: they sometimes generate content that looks factual but is completely fabricated, known as hallucination [34]. This issue became evident when a lawyer in the United States filed a legal brief citing fake case laws created by ChatGPT [3]. This incident revealed the dangers of using AI tools without proper guardrails and emphasized the challenges caused by the models' limited expertise in specialized areas like law. To fully harness the potential of Generative AI for legal research, we must address these shortcomings.

To build a Large Language Model (LLM), the process begins with pre - training, where a transformer based model [32] is trained on a massive collection of data. This data is usually general and not tailored to any specific domain or field, and it doesn't update over time. Because of this, LLMs like GPT - 4 [27] perform well on general questions but often struggle with specialized or advanced topics in areas like law. To solve this, we need to inject knowledge into the model. The two main ways to do this are fine - tuning [13] the model with domain - specific data and using in - context learning [7]. One of the most effective methods for in - context learning is Retrieval - Augmented Generation (RAG) [18].

2.1 Fine - tuning

Fine - tuning [19] [30] adjusts the parameters of a Large Language Model (LLM) to make it better suited for specific tasks or specialized domains. There are two main approaches: supervised fine - tuning, often called instruction fine - tuning, and unsupervised fine - tuning. Instruction fine - tuning trains the model to follow specific instructions more effectively by providing examples of tasks with clear inputs and expected outputs. On the other hand, unsupervised fine - tuning extends the pre - training process but uses domain - specific text instead of general data. Since this method relies on unlabeled

data, the model does not receive explicit instructions or labels. The goal is for the model to absorb and retain the knowledge from the additional training data, enabling it to handle specific domain - related tasks, like answering detailed legal questions. Unsupervised fine - tuning is more scalable because unlabeled data is easier to gather, but it often produces less reliable results compared to supervised methods. Additionally, even with domain - specific data, the model might not consistently retrieve the added knowledge, especially when the domain - specific dataset is small compared to the extensive corpus used during pre - training. Fine - tuning a model can be expensive due to the ongoing training costs required to update it as new data becomes available.

2.2 Naive Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation enables Large Language Models (LLMs) to access new knowledge sources to answer questions. This approach improves pre - trained models by adding extra information to the input query, known as in - context learning. Instead of updating the model's weights like fine - tuning, it simply modifies the query to include relevant knowledge. This method ensures the model can directly reference the provided information, avoiding the risk of it being lost in training data, as might happen with the unsupervised fine - tuning. However, since language models have a limited context window, only a small portion of the total knowledge can be added to a query at once. We also want to reduce the amount of content we pass into model to reduce latency and cost of inference. Retrieval Augmented Generation (RAG) simplifies in - context learning by providing only the most relevant information to the model. In legal research, there are often hundreds of thousands of judgments and statutes, each running hundreds of pages long. Passing an entire document into the model is neither practical nor efficient. When a lawyer searches for case law on a specific scenario, the system must find the right chunk of the document, and include it in the context along with the lawyer's question. This helps the model provide accurate and efficient answers.

As shown in Figure 1 The first step in building a RAG is to process the documents, first split documents into smaller chunks. Then, use an embedding model [21] to generate numerical vector rep - resentations for each chunk. Finally, store the embeddings in a Vectorstore [17].

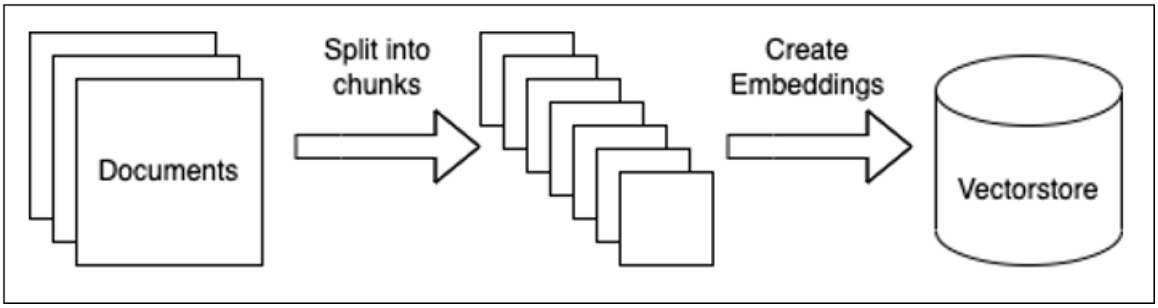

Figure 1: Prepare documents.

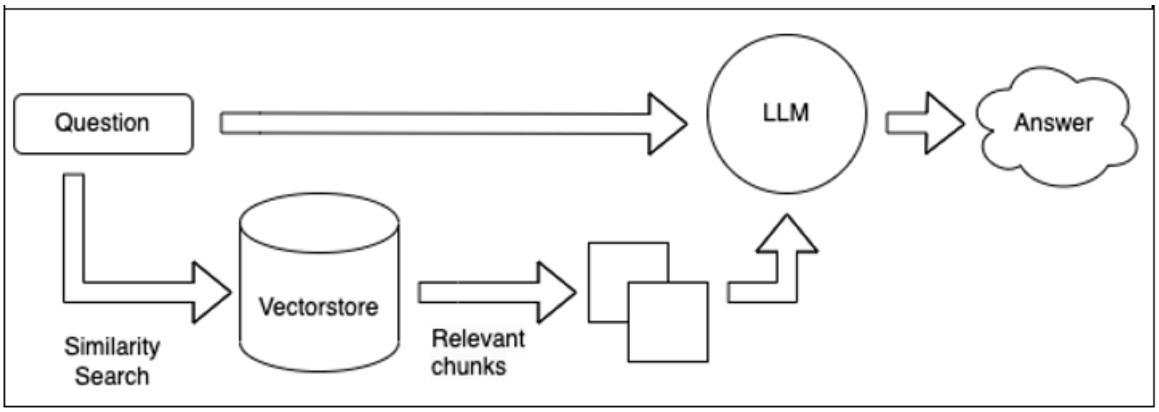

Figure 2: Ask questions.

Once the document index is prepared, we can start asking questions as shown in Figure 2. The system will retrieve relevant documents based on the query. The process begins by creating an embedding for the input question. This embedding is then compared with the embeddings in the Vectorstore to identify the most relevant document chunks. The top N relevant chunks are fetched and added to the context of the prompt. This enriched prompt is then sent to the LLM, which uses the provided context to generate a precise and contextual answer.

2.3 Limitations of the Naive RAG Pipeline for Legal Research

The traditional RAG pipeline has several limitations that reduce its effectiveness in handling knowledge - heavy and specialized natural language processing tasks. Most RAG systems divide documents into equally sized chunks, without considering the document's structure or content. These chunks typically contain the same number of words or tokens, with some overlap to maintain context. The retriever then selects the top k most similar chunks, which are usually of a

uniform size. Legal texts have a distinct hierarchical structure [6] that organizes legal provisions from broad areas to specific articles, clauses, and sub - clauses. In structured normative texts like constitutions, every detail carries significant semantic weight. Unlike everyday language, which often includes redundancy and informality, constitutions are meticulously crafted, with each word, phrase, and clause deliberately chosen to convey precise legal meaning and intent. This level of precision demands a detailed approach to information retrieval. By analyzing and representing legal knowledge at various levels, from individual clauses and sub - clauses to larger articles, chapters, and titles, we can fully capture the depth of meaning contained in these texts.

One of the core components of RAG is semantic search. The semantic search have challenges such as retrieving irrelevant or opposing information, indicating the model's sensitivity to language nuances and the potential for unexpected results. The most similar chunks are not necessarily the most relevant chunks, however many RAG pipelines make this assumption. In legal contexts, this can create issues. For example, newer statutes might override older ones. Users often need a way to filter results to find statutes published after a specific year or to answer questions based only on the latest version of a statute. Unfortunately, basic RAG systems do not provide these filtering features, making it harder to handle such use cases effectively. Additionally, most times, standard embedding algorithms do not have any domain - specific knowledge and therefore, they overlook nuances of certain words or phrases in the legal domain.

3. Methodology

Given the various limitations of current and naive RAG Pipelines, there is clearly a need to develop and implement different strategies in order to address the challenges for the legal research. Throughout this section, we will explore a variety of different techniques that are aimed to optimize RAG performance for the use case.

3.1 Multi - layered Chunking and Embedding

Chunking involves breaking down text into smaller, manageable units such as sentences, paragraphs, or specific legal provisions. By segmenting complex documents into these smaller parts, it becomes easier to closely analyze each segment's content and understand its role within the overall context of the document. This approach is especially useful for processing and interpreting detailed legal texts. Semantic chunking, as implemented in systems like semchunk [4], takes this a step further by dividing text into semantically meaningful sections. It uses a recursive process that starts by splitting the text based on logical separators, such as punctuation or paragraph breaks. This process continues until each segment reaches a defined size, ensuring that each chunk represents a distinct theme or concept. However, this method has limitations when applied to legal documents. It fails to account for the inherent hierarchical structure of legislative texts, where the organization of provisions is carefully defined by the author. Traditional semantic chunking treats all segments as being on the same level, ignoring the multiple layers of structure that determine the legal importance and interconnections of different parts. Consequently, while it creates coherent and manageable chunks, it does not fully capture the layered and systematic organization essential to understanding legal documents.

One of the better approaches for chunking and embedding the legal documents is to organize the legal documents into multiple hierarchical layers [6], each capturing different levels of detail. At the highest level, a document - level embedding represents the overall theme and purpose of the legal text for classification purposes. The document component level focuses on specific parts like main texts, justifications, or schedules, assigning each an embedding to reflect their unique contributions. The basic unit hierarchy level captures broader structures such as books, chapters, and sections, while the basic unit level creates embeddings for individual articles to represent specific legal issues. Further granularity is added at the basic unit component level by embedding article components like headings or paragraphs and at the enumeration level for detailed elements within these components. This multi - layered approach enables RAG models to provide detailed and contextually relevant responses to user queries.

3.2 Query Expansion and Identifying the Filters

Query Expansion [15], also known as query transformation, modifies the user's original question to improve the retrieval of relevant information in a RAG pipeline. Often, user questions do not provide enough detail to guide the algorithm effectively, particularly when using cosine similarity for matching. This can result in the algorithm retrieving irrelevant chunks by focusing on the wrong sections of the document. For more complex queries and documents, additional reasoning steps are necessary, which basic RAG pipelines are not equipped to handle.

An advanced method to address this is Hypothetical Document Embeddings (HyDE) [9]. HyDE enhances retrieval by using a large language model (LLM) to generate a hypothetical response to the user's query [16]. The system then performs similarity searches using both the original question and the generated hypothetical document. This technique has been shown to outperform traditional methods and eliminates the need for custom embeddings. However, it is not without limitations, as the quality of results depends on the accuracy of the LLM - generated hypothetical document, which can occasionally lead to errors. Query expansion techniques can also be used to automatically identify and apply filters while searching the Vectorstore.

3.3 Metadata Annotation

Metadata annotation plays a vital role in enhancing the document processing phase of Retrieval - Augmented Generation (RAG) systems. This process involves enriching documents with structured information, such as titles, authors, publication dates, and categories, to improve both retrieval accuracy and contextual understanding. For example, when processing case laws, metadata may include details such as the names of judges, the year of judgment, the type of case, and the popular name of the case. Each document or its segmented parts is tagged with metadata that captures its content, origin, and relevance. This added layer of

information allows the RAG pipeline to conduct more targeted searches and prioritize results based on contextual factors like dates or legal jurisdictions. By integrating metadata, RAG systems can better align retrieved chunks with user queries, resulting in more accurate and meaningful responses. Additionally, metadata annotation enables advanced query features, such as filtering and sorting results, making it an essential component of effective document processing.

3.4 Re - ranking To Improve Retrieval Relevancy

Standard RAG pipelines allow users to specify how many documents or chunks the algorithm should retrieve and use as context for a query. Typically, the top 1 or 2 chunks determined by cosine similarity or k - nearest neighbors search are included as context. However, these methods prioritize similarity, which does not always align with relevance. Re - ranking algorithms [10] [12] address this limitation by reorganizing the retrieved chunks based on their relevance rather than similarity. For example, after ranking the top 10 chunks by cosine similarity, a re - ranking algorithm evaluates their relevance and selects the most pertinent chunks to include as context.

3.5 Fine - tuning the Embedding Model

Embedding algorithms are what convert text into numerical representations, and play a crucial role in RAG pipelines. It is possible to fine - tune [13] embedding models based on domain - specific knowledge to enhance retrieval [31] [2] [11] in that specific domain. Embeddings can also be dynamic where they adapt when words have slightly different meanings based on the context. Fine - tuning embedding models with legal documents could significantly improve models ability to differentiate meaning for different words in the legal context and improve overall accuracy.

3.6 Hybrid Search for Retrieval

A hybrid approach combining dense and sparse vector embeddings is a powerful method for improving accuracy and relevance in Retrieval - Augmented Generation (RAG) systems [22]. Dense embeddings, generated by transformer based models capture semantic meaning and are effective for finding contextually similar content [14], even when the exact words differ. Sparse embeddings, on the other hand, rely on traditional term - based methods like TF - IDF or BM25 [23] and excel at capturing ex - act matches and keyword - based relevance [24]. By integrating both methods, the hybrid approach leverages the strengths of each: dense embeddings ensure semantic richness, while sparse embeddings provide precision for keyword alignment. This combination enhances the retrieval pipeline by covering a broader spectrum of contextual and literal relevance, reducing the chances of missing critical information. Additionally, re - ranking mechanisms can further refine results by weighting dense and sparse contributions based on the query type, ensuring that the most pertinent chunks are selected as context for RAG responses. This dual - layered strategy leads to more accurate, relevant, and robust retrieval performance

4. Evaluation

When evaluating RAG systems in the legal domain, two primary aspects must be assessed: the system's ability to retrieve relevant legal context and its capacity to provide accurate answers based on that context. The Legal dataset we have include ground truth context and answers curated by legal experts, such as case summaries or interpretations of statutes. Structured evaluation involves comparing the model's responses to these ground truth answers, offering a robust measure of the model's accuracy. On the other hand, unstructured evaluation assesses the quality of retrieved chunks and the generated answers without relying on predefined ground truth data. While structured evaluation provides a clearer picture of the model's precision, unstructured evaluation is valuable in scenarios where ground truth data is unavailable, a common situation in the legal domain due to the complexity and variability of legal texts.

4.1 Retrieval Quality

The primary method to evaluate retrieval quality in the legal domain using structured evaluation is through document - level and section - level accuracy. Since the dataset includes entire legal documents and the specific sections referenced by legal professionals, we compare these referenced sections to the chunks returned by the system. If the retrieved chunks correspond to the same document or section cited by the expert, the retrieval accuracy is considered high. A similar approach is applied to evaluate section - level accuracy.

For unstructured evaluation, the quality of retrieved chunks can be assessed using Context Relevance, as outlined in the RAGAS Framework [8]. This metric involves prompting a large language model (LLM) to identify the number of sentences within the retrieved context that are directly relevant to answering a legal query. The context relevance score is calculated as the ratio of relevant sentences to the total number of sentences in the retrieved chunk. This score penalizes redundant or irrelevant information and rewards chunks that predominantly contain sentences providing substantive answers to the legal question, ensuring the retrieval is both precise and meaningful.

4.2 Answer Accuracy

For structured evaluation of question - answering in the legal domain, we assess the accuracy of the model's answers against ground truth answers provided in the dataset. Standard metrics like BLEU [29], Rouge - L [35] and cosine similarity [1] are used for comparison. However, these metrics often fall short in capturing the semantic nuances between two legal answers and may provide misleading results. To address this limitation, we also employ LLM - based evaluation [20], using tailored prompts with models like GPT to assess the accuracy of a model's response compared to the ground truth answer.

Additionally, we evaluate the quality of the generated answers in an unstructured manner by analyzing their alignment with the retrieved legal context. Metrics like Answer Faithfulness, introduced by the RAGAS [8] framework, are particularly valuable. Faithfulness measures

how well the model's answers are grounded in the provided context by calculating the proportion of statements in the answer that are supported by the context. This score also serves as an indicator of whether the model is hallucinating information without relying on ground truth answers. A high faithfulness score indicates that the generated answer is well - supported by the retrieved legal documents and does not introduce information beyond the context, ensuring reliability and accuracy in legal applications.

5. Results

To evaluate the performance of the advanced RAG system, we developed a benchmark dataset based on judgments from the Supreme Court of India. This dataset consists of 300 questions about case laws covering a wide range of legal topics. Each entry includes a question (e. g., "Which is the most important case law discussing LGBTQ rights in India?"), an answer (e. g., "Navtej Singh Johar vs. Union of India, Ministry of Law and Justice, September 6, 2018"), an evidence string (containing the necessary information to verify the answer), and the page number from the relevant document. This dataset enables a comprehensive assessment of the RAG model's ability to retrieve relevant context and generate accurate answers to legal questions, showcasing the potential of large language models (LLMs) in advancing legal research. We compared the performance of a naive RAG setup against the advanced RAG. As illustrated in table 3, the advanced RAG approach outperformed the naive RAG setup across all evaluated metrics.

RAG	Cosine_Similarity	Bert_Score	Lim_Eval
Naïve RAG	0.1	0.6	0.2
Advanced RAG	0.28	0.8	0.6

Figure 3: Results

6. Conclusion

Enhancing retrieval performance significantly improves the overall quality of systems designed for document - based question - answering in the legal field. Identifying the correct sections of text not only ensures more accurate legal references but also results in clearer and more precise answers to legal in - quiries. This underscores the critical role of effective retrieval mechanisms, as even the most advanced generation models rely on accurate context to produce reliable responses. We utilized various evaluation methods to highlight how advanced RAG techniques address known retrieval and generation challenges.

Despite the significant progress made in enhancing Retrieval - Augmented Generation (RAG) systems, there are additional opportunities to improve the advanced RAG architecture. One such avenue is the integration of Corrective RAG approaches [33], which focus on dynamically refining the retrieval process and correcting errors during runtime. This method enables the system to iteratively improve its retrieval accuracy by analyzing feedback from earlier stages, ensuring more relevant context is provided for answering questions. Additionally, Agentic RAG [28] approaches offer a promising di - rection by introducing autonomous decision - making capabilities within the RAG framework. These approaches involve leveraging agents that can reason, plan retrieval strategies, and adapt to complex, multi - step queries often encountered in the legal domain. By incorporating these advanced method - ologies, the performance of RAG systems could be significantly enhanced, addressing limitations in retrieval accuracy and contextual understanding while pushing the boundaries of their application in specialized fields like legal research and beyond.

References

[1] Mohammad Alodadi and Vandana P. Janeja. Similarity in patient support forums using tf - idf and cosine similarity metrics. In *2015 International Conference on Healthcare Informatics*, pages 521–522, 2015.
[2] Amazon Web Services. Improve rag accuracy with fine - tuned embedding models on amazon sagemaker, n. d. Accessed: 2024 - 11 - 29.
[3] Molly Bohannon. Lawyer used chatgpt in court and cited fake cases—a judge is considering sanctions. *Forbes*, 2023.
[4] Umar Butler. Sem chunk. https: //github. com/umarbutler/semchunk/.
[5] ROBERT DALE. Law and word order: Nlp in legal tech. *Natural Language Engineering*, 25 (1): 211–217, 2019.
[6] Jo˜ao Alberto de Oliveira Lima. Unlocking legal knowledge with multi - layered embedding - based retrieval, 2024.
[7] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in - context learning, 2022.
[8] Shahul Es, Jithin James, Luis Espinosa - Anke, and Steven Schockaert. Ragas: Automated evalu - ation of retrieval augmented generation, 2023.
[9] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero - shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd - Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguis - tics.
[10] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Peng - shan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate, 2022.
[11] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen - Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low - rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021.
[12] Taeho Hwang, Soyeong Jeong, Sukmin Cho, SeungYoon Han, and Jong C. Park. Dslr: Document refinement with sentence - level re - ranking and reconstruction to enhance retrieval - augmented gen - eration, 2024.
[13] Cheonsu Jeong. Domain - specialized llm: Financial fine - tuning and utilization method using mis - tral 7b. *Journal of Intelligence and Information Systems*, 30 (1): 93–120, March 2024.
[14] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen - tau Yih. Dense passage retrieval for open -

domain question answering. In *Pro - ceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020.

[15] Hamin Koo, Minseon Kim, and Sung Ju Hwang. Optimizing query generation for enhanced document retrieval in rag, 2024.

[16] Hamin Koo, Minseon Kim, and Sung Ju Hwang. Optimizing query generation for enhanced document retrieval in rag, 2024.

[17] Sanjay Kukreja, Tarun Kumar, Vishal Bharate, Amit Purohit, Abhijit Dasgupta, and Debashis Guha. Vector databases and vector embeddings - review. In *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)*, pages 231–236, 2023.

[18] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Ku¨ttler, Mike Lewis, Wen - tau Yih, Tim Rockt¨aschel, Sebastian Riedel, and Douwe Kiela. Retrieval - augmented generation for knowledge - intensive nlp tasks. In H. Larochelle,

[19] M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Pro - cessing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[20] Chun - Hsien Lin and Pu - Jen Cheng. Legal documents drafting with fine - tuned pre - trained large language model. In *Software Engineering amp; Trends*, SE, page 201–214. Academy Industry Research Collaboration Center, April 2024.

[21] Yen - Ting Lin and Yun - Nung Chen. Llm - eval: Unified multi - dimensional automatic evaluation for open - domain conversations with large language models, 2023.

[22] Qi Liu, Matt J. Kusner, and Phil Blunsom. A survey on contextual embeddings, 2020.

[23] Priyanka Mandikal and Raymond Mooney. Sparse meets dense: A hybrid approach to enhance scientific document retrieval, 2024.

[24] Divyanshu Marwah and Joeran Beel. Term - recency for TF - IDF, BM25 and USE term weight - ing. In Petr Knoth, Christopher Stahl, Bikash Gyawali, David Pride, Suchetha N. Kunnath, and Drahomira Herrmannova, editors, *Proceedings of the 8th International Workshop on Mining Sci - entific Publications*, pages 36–41, Wuhan, China, 05 August 2020. Association for Computational Linguistics.

[25] I. C. Mogotsi. Christopher d. manning, prabhakar raghavan, and hinrich schu¨tze: Introduction to information retrieval. *Information Retrieval*, 13 (2): 192–195, 2010.

[26] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2023.

[27] OpenAI. Chatgpt. https: //chatgpt. com/.

[28] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren - cia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Moham - mad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett - Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna - Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brit - tany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim´on Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo - Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jo - moto, Billie Jonn, Heewoo Jun, Tomer Kaftan, L - ukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, L - ukasz Kondraciuk, An - drew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David M´ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ash - ley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Hen - rique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Pow - ell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian

Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Pet - roski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer´on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Weli - hinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt - 4 technical report, 2024.

[29] Chidaksh Ravuru, Sagar Srinivas Sakhinana, and Venkataramana Runkana. Agentic retrieval - augmented generation for time series analysis, 2024.

[30] Ehud Reiter. A structured review of the validity of bleu. *Computational Linguistics*, 44 (3): 393– 401, 09 2018.

[31] Nolan Satterfield, Parker Holbrook, and Thomas Wilcox. Fine - tuning llama with case law data to improve legal domain performance, 05 2024.

[32] Yixuan Tang and Yi Yang. Do we need domain - specific embedding models? an empirical inves - tigation, 2024.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,

[34] L - ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg,

[35] S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[36] Shi - Qi Yan, Jia - Chen Gu, Yun Zhu, and Zhen - Hua Ling. Corrective retrieval augmented genera - tion, 2024.

[37] Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. Cognitive mirage: A review of hallucinations in large language models, 2023.

[38] Ming Zhang, Chengzhang Li, Meilin Wan, Xuejun Zhang, and Qingwei Zhao. Rougesem: Better evaluation of summarization using rouge combined with semantics. *Expert Systems with Applications*, 237: 121364, 2024.