

Structuring Content for Retrieval-Augmented Generation Chatbots: An Analysis of Current Best Practices

Amari Conner

University of Minnesota - Twin Cities

Master of Science in Scientific and Technical Communication

Dr. Dan Card | May 2025

Abstract

Background. As generative Artificial Intelligence (AI) tools become more popular, information-seeking behaviors are shaping how businesses produce content and how users access information. However, mass-market chatbots like ChatGPT are prone to producing inaccurate information due to their broad training data. **Objective.** My research aimed to identify broadly recognized best practices for structuring content to enhance the effectiveness of Retrieval-augmented Generation (RAG) chatbots.

Methods. To find practical information on RAG chatbots, I analyzed source content from practitioner spaces, including Software as a Service (SaaS) blogs and professional conference materials, focusing on broadly recognized recommendations for structuring content for RAG models. **Results.** I analyzed 16 unique practitioner sources and coded recurring themes into seven best practice heuristics for structuring content to improve RAG chatbot performance. **Conclusion.** While no universal standards exist yet for structuring content for RAG chatbots, my research identified overlapping best practices that can guide the implementation of RAG chatbots with an organization's documentation source. To be successful, organizations must test and adapt these strategies to their specific content and remain committed to ongoing performance monitoring.

Introduction

Given the popularization of generative Artificial Intelligence (AI) tools like ChatGPT, the way people look for information is changing. Instead of typing short keyword searches into Google or other search interfaces, users are now asking full questions or having conversations with chatbots to get personalized responses. For example, a simple keyword search might be "broken phone help," whereas with generative AI, people ask conversational questions like "My phone won't turn on. Can you tell me how to figure out what might be causing it?" Generative AI tools are trained on large datasets, giving

them natural language processing capabilities and allowing them to carry out human-like conversations, answer questions, provide detailed instructions, and more.

Generative AI in Modern Information-Seeking

This shift in information-seeking behaviors is reshaping how businesses produce content, how employees find internal knowledge, and how consumers seek information on specific products and services. Generative AI chatbots are becoming a popular tool in customer service contexts, allowing users to search for information on demand via a chatbot interface hosted on an organization's website, freeing up time for support teams.

Research Focus

The problem with mass-market conversational chatbots like ChatGPT is that they are trained on vast amounts of data from across the internet, making them prone to producing inaccurate or generalized information. For this reason, these types of chatbots are not a reliable solution for businesses trying to enhance self-service with their documentation. Instead, my research focused on AI chatbots trained on an organization-specific data source to increase response accuracy. These chatbots are commonly called Retrieval-augmented Generation (RAG) chatbots, combining conversational AI capabilities with a relevant data source to deliver accurate and contextually relevant responses to technical questions.

In RAG models, your “documents,” such as a knowledge base, content management system, or domain-specific web content, are fed into the AI model and indexed. Indexing is the process of segmenting the data into smaller chunks, organizing them according to semantic similarity, and storing them in a database for quick retrieval. When the user asks a question or provides a prompt to the chatbot interface, the model **retrieves** the most relevant chunk of information from the database based on its understanding of the user's intent. The **generation** mechanism involves the chatbot's large language model (LLM), transforming the retrieved content into a conversational, contextually grounded response. Figure 1 shows a visual representation of this process.

Case Examples of RAG Chatbot Performance

Retrieval-augmented generation chatbots have shown impressive accuracy when trained on a specific data source. One case study of an RAG chatbot trained with external data sources reported 94% accuracy (Feldman et al., 2024). With successful cases of implementation, these types of chatbots are becoming a popular choice for organizations seeking to enhance their content strategy with AI.

To give an example of a real-world RAG model, Appendix A shows a sample interaction transcript from [kapa.ai's](#) RAG chatbot featured on their website. To determine which

portions of the response were generative and which portions were retrieved from kapa’s documentation sources, I cross-referenced kapa’s documentation. It was fairly easy to determine, given kapa’s RAG model often provides links to the documentation that the response was grounded in.

Appendix B shows ChatGPT’s response to the same question posed to kapa’s RAG model in Appendix A. While not inherently a “hallucination,” ChatGPT’s response lacks the depth, nuance, and conservative tone that kapa’s response provided, most notably missing kapa’s emphasis on testing your content and what “content quality” means. Unlike kapa, ChatGPT also does not ground its response with direct links to kapa’s documentation.

Appendix C shows a sample interaction transcript from kapa.ai’s RAG chatbot where the bot does not have enough context from the documentation. The bot makes it clear that it does not “have specific information” related to my question. It proceeds to provide a summary of the type of information covered in kapa’s documentation and encourages me to reach out directly to the kapa team for an answer.

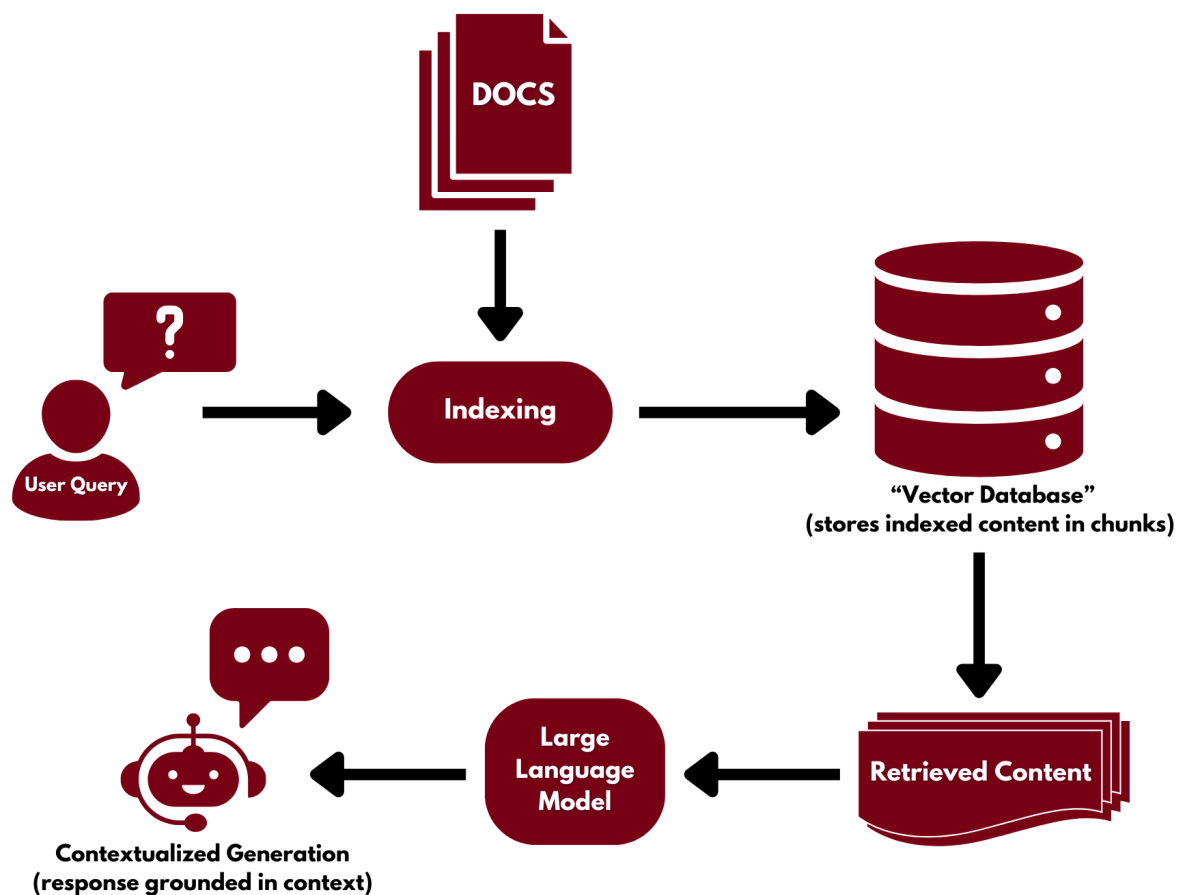


Figure 1. Basic Retrieval-Augmented Generation Architecture.

Motivation for the Study

AI and chatbots are new and complex subjects in the field of technical communication, making it hard to address specific considerations in structuring content for AI chatbots. Many businesses with product troubleshooting and service needs want to explore use cases of AI chatbots with their documentation to reduce calls or tickets about common questions. However, they may lack readiness in terms of clean, consistently formatted, and complete documentation necessary for a seamless user experience with a chatbot.

In order to be accurately interpreted and relayed by an AI chatbot, existing documentation must be reviewed to identify gaps, redundancies, or outdated information. Technical communicators are important resources in implementing these tools, assisting in the restructuring and rewriting of existing content, along with the development of standardized content schemas and information ecosystems. They also work with IT teams on training, testing, and refining RAG models.

Another challenge businesses face is understanding the specific ways to structure and write content that increases a model's accuracy in responding to user questions. Despite the hype of machine-learning capabilities, you cannot just "plug in" documentation and expect AI to do the rest; these models require continuous monitoring and fine-tuning. Therefore, my research zooms in on structuring content for AI chatbots, which can be used in the broader context of an organization's content strategy and AI "readiness."

The following questions guided my research:

- What are the best practices in structuring content to enhance the usefulness of a RAG chatbot?
- What are the dependencies and nuances of these practices?

To answer these questions, my research had the following objectives:

- Perform a content analysis of several practitioner sources.
- Synthesize recurring themes and recommendations.
- Develop best practice heuristics.
- Identify nuances in the recommendations.

Defining chatbot success

In order to develop best practices, it is important to first understand what success means in the context of RAG chatbots as opposed to other chatbot models. In her AI the Docs conference presentation (2024), Amara Graham defines success with three characteristics:

- The chatbot is a low effort to implement and maintain by stakeholders.
- The chatbot accurately responds to user queries, indicating that the documentation is complete and accurate.
- The chatbot allows immediate user feedback to promote continuous improvement.

These characteristics of successful implementation should guide implementation along with the generalized best practices I identify.

Methods

Because my research topic has developed rapidly over the last few years, it is not fully represented in academic literature yet. Instead, valuable knowledge currently exists in practitioner spaces, such as software as a service (SaaS) blogs and professional conference materials.

To find practical knowledge on this topic, I performed a mixed-source content analysis. Mixed-source content analysis is a qualitative research method that involves analyzing content from a variety of sources to identify patterns, themes, and recommendations on a specific topic.

Some sources describe the capabilities of specific tools used in RAG, but my goal was to primarily explore recommendations that are broadly recognized, regardless of the tool used. I aimed to identify recommendations that may not work the same in all contexts or should be tested to determine how they work with specific content sources.

Inclusion criteria

- The source must describe recommendations in the context of RAG (or similarly defined) chatbot models.
- The source must describe overarching recommendations in structuring content for RAG (or similarly defined) chatbots.
- The source must be less than five years old.

Based on these criteria, I excluded sources that exclusively described strategies for a particular chatbot tool.

Results

I analyzed 16 unique sources according to my inclusion criteria. Many sources provided recommendations in more than one category. For authors of more than one source, I only counted them once per category. I consulted a combination of online sources, including presentations from the [AI The Docs 2024](#) online conference, professional

blogs, and YouTube videos. I coded the information into seven categories representing broad best practices. Figure 2 shows an overview of the total supporting sources in each category. Appendix D shows a complete list of the supporting sources in each category.

Best Practices in Structuring Content for RAG Chatbots

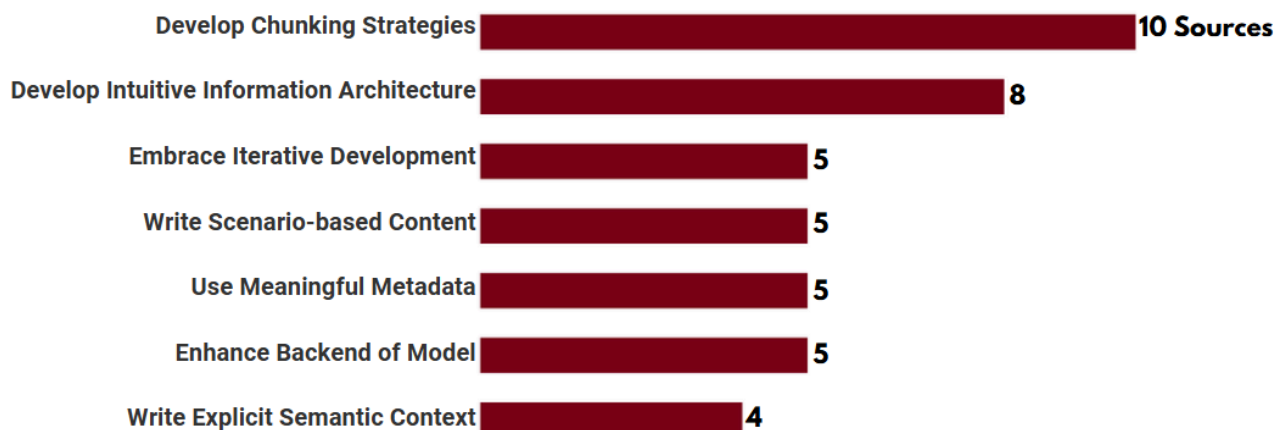


Figure 2. Total number of supporting sources for each best practice heuristic.

More than half of the sources I analyzed describe the importance of considering both humans and AI as audience members of your content (Benedetti, 2024; Chinchilla, 2024; Crowhurst & Poulstrup, 2024; Murugesan, 2023; Gomez, 2024; Graham, 2024; Gurtam, 2024; Perlmutter, 2024; Sorensen, 2024b). Among these, several described the importance of basic technical writing principles while also considering additional ways to make content digestible by AI chatbots.

Fabrizio Benedetti (2024), a UX professional, summarized the intersection of writing for AI and humans in his [technical writing predictions for 2025](#):

“Providing a better channel for AI consumption is not the same as writing for AIs. You must keep writing for humans, because humans are the ones who’ll buy your goods or hire you; the difference is that this might now be mediated by LLMs, which act as a channel. Much in the same way you present content differently for mobile or print, you need to think how to present content to AI agents.”

In this excerpt, Benedetti describes AI agents as a “channel,” urging content professionals to consider them as another consumer of your content.

On the other hand, Murugesan (2023), a data science professional at a leading knowledge management SaaS firm, acknowledges considering both human and AI readers but emphasizes the differences between writing for humans vs. AI. In his blog,

Murugesan (2023) writes, “Technical writers need to shift their thinking in producing any knowledge base content that is AI bot-friendly and makes it easy for both AI bots and humans to comprehend. In the same paragraph, Murugesan (2023) writes, “The characteristics of writing for human consumption are extremely different from how to produce content for AI bots” (Murugesan, 2023). Nonetheless, the emphasis in these sentiments is that AI bots alter the channel through which users are accessing your content. This change does not mean writing for AI is at the expense of human readers. Instead, it requires content to be optimized for AI to effectively meet end-user needs through a chatbot interface.

Similarly, other sources highlight the importance of viewing both AI and humans as “users” or an “audience” of your documentation (Crowhurst & Poulstrup, 2024; Gomez, 2024), reinforcing the need to write for both in mind to ensure accurate chatbot responses.

Amara Graham, a leading voice in AI and docs, describes the importance of content quality in the context of AI chatbots simply as “garbage in, garbage out” (2024). When considering AI chatbot use cases, you should already have effective content management strategies designed to maximize findability and efficient navigation of content repositories. If your documentation is not already serving the needs of your users, investing in an AI solution will not solve your problem.

Others describe current technical writing best practices as reliable practices in the context of AI:

- “Many things that work for humans will work for AI (but not all)” (Gurtam, 2024).
- In describing tips for writing good content for machines, Chinchilla (2024) states, “As good technical writers, we’re probably doing all of these, but check the tools you use are.”
- “If it’s not parsable by a machine, with some exceptions...it’s probably not going to be parsable by a human” (Graham, 2024).

These sentiments are reassuring that writing for AI is not entirely reinventing the wheel, but **does** require considerations for making content more easily digestible by machines. The following sections describe the best practices introduced in Figure 2.

Embrace Iterative Development

Many sources describe the importance of iterative development and continuous improvement when implementing a chatbot for your documentation; doing so is necessary to achieve the best possible user experience.

Sources mainly described two different strategies: testing your content with a chatbot model and leveraging usage analytics to drive continuous improvement of your content. To test the model, you should use mock questions representing real use cases to determine the accuracy of responses (Graham, 2024; Crowhurst & Poulstrup, 2024).

Gomez (2024) describes the concept of “Fail-Case Driven Development” because of the nature of LLMs. Large language models are not “deterministic” (Gomez, 2024), so it is best to improve the system when you find failures.

Sources also describe the importance of monitoring system performance in production by analyzing metrics, such as user questions (Gomez, 2024; Graham, 2024) and implementing user feedback mechanisms (Graham, 2024; Singh, 2024). These strategies can help evaluate whether users are getting their questions answered and whether there are gaps in documentation.

Crowhurst & Poulstru (2024) make the distinction of measuring “retrieval performance separate from generation performance” because not all failures are due to the documentation itself; they can also be caused by how the machine parses the information.

Develop Intuitive Information Architecture

Information architecture (IA) focuses on the structure, organization, and labeling of information to increase findability. Several sources recommend strategies related to the structure of content to support more accurate indexing and retrieval.

Consistent And User-Centered Terms

For LLMs to accurately match user queries to content, the terms in your documentation must be consistent and associated with the user’s intent. This is not a new recommendation with the advent of AI, but it has become an important part of improving chatbot response accuracy. Whereas humans may be able to quickly associate a word with familiar synonyms, it may not always be explicit to LLMs. AI User Group (2023) recommends including not only synonyms but also antonyms, hyponyms, and booleans within your documentation to provide more context to your model and mitigate bias. Other sources recommend using a business glossary to keep terms consistent across your documentation (Murugesan, 2024c; Gurtam, 2024).

Logical and Structured Document Headings

Several sources mention heading structure and document hierarchy. Many emphasize the importance of document structure being “scrape-able” (Perlmutter, 2024), as in being a machine-readable format, such as HTML or Markdown, so machines can recognize the hierarchy of your document.

Some sources describe the overall importance of a clear hierarchy of headings and subheadings (Gomez, 2024; Sorensen, 2024b). Gurtam (2024) notes that headings are pointers for humans and AI. These recommendations are not new, as they are standard technical writing principles that already benefit human readers.

Other sources build on the importance of ensuring heading text not only describes the section accurately but is also marked up properly in the source code, such as HTML, with the use of heading tags (AI User Group, 2023; Chinchilla, 2024; Gurtam, 2024; Murugesan, 2024). Humans primarily rely on visual hierarchy, such as heading size, color, and font, relative to body text. On the other hand, the source code structure gives context to AI models for what that chunk is about (Gomez, 2024). This recommendation means there should be an extra step in ensuring headings are suitable for human readers and AI. Chinchilla (2024) specifically recommends that headings be in an active gerund “-ing” style because users often ask questions, such as “How do I do this?” and bots will look for answers in a “similar form.”

Other sources specify ensuring the *correct* heading tag structure in documents. You should only have one H1 (title) tag for each document and subsequent subheadings (AI User Group, 2023; Chinchilla, 2024).

Intuitive Information Taxonomy

Information taxonomy is how information is organized and categorized within a knowledge base or content library. Taxonomies “clarify the context, relationships, and semantics among entities with an overarching framework” (AI User Group, 2023). AI tools use this information to understand the relationships among your document topics. The relationships between your topics should be clearly defined in the structure of your document library. Most authoring tools used to publish digital content or manage content in a searchable knowledge base have features to organize content into categories and subcategories.

Other sources describe this heuristic as an intuitive “breadcrumb” trail (Gomez, 2024; Gurtam, 2024), otherwise known as the navigation path (Gomez, 2024) to get to a chunk of content.

It is important to avoid “overlapping and duplicate categories” and strive for “mutually exclusive and collectively exhaustive” categories for your content (Gurtam, 2024; Solis, n.d.). Duplicate categories and sources of information lead to less precise response generation.

Avoiding Content Stored in PDFs

In a separate, but related blog post to his conference presentation, Sorensen (2024a) recommends avoiding “storing docs in files” because LLMs “have a harder time parsing

these.” Although recommended in just one source without much explanation, it likely stems from the tendency of PDF content to be unstructured, contrasting with how LLMs scrape the structural HTML tags in web content.

Optimizing Performance with Tables

Tables are often a great way to make complex data more digestible to humans. However, LLMs are designed to process primarily textual, verbose data, so their reasoning model can be challenged when confronted by the tabular structure of tables. Gurtam (2024) states that tables are “still painful” for AI models.

On the other hand, some sources describe specific strategies to make tables more digestible by AI models. For example, Murugesan (2024b) offers [Guidelines for Structuring tables in technical writing for GenAI-based agents](#), including:

- “Do not use symbols inside the table content as they are removed during pre-processing steps
- Do not have null values / empty spaces inside your table content as GenAI-based agents might hallucinate while trying to use those data!
- Ensure that tables have header information along with proper rows
- If you wish to have some binary information part of the table content, use Yes/No, True/ False, or any other option. Ensure that this information is covered in the system message of your RAG (Retrieval Augmented Generation) tool
- Use <abbr> tag to define abbreviations of terms inside the table content
- Use <abbr> tag to describe tick mark and cross mark so that LLMs can understand the meaning of symbols inside the table content
- ...[It] is recommended to have one type of data present inside those table cells”

Chinchilla (2024) and Gurtam (2024) recommend providing context in the text surrounding the table to help models understand what information the table contains.

These recommendations should be taken into account when including tables in your documentation, and performance should be tested to ensure that the model is able to parse tables accurately.

Develop Chunking Strategies

Chunking strategies are techniques for breaking larger pieces of information into smaller chunks and determining how your model should parse your documents to maximize efficiency and accuracy during retrieval. In general, larger sections of content take more processing power for AI tools, so you should consider how to chunk your content

effectively so the model can process your information efficiently without losing essential context.

Gomez (2024) recommends content be granular and structured as opposed to “monolithic pages.” Humans also tend to prefer easily scannable, bite-sized chunks of content with clear headings, so organizations should already be trying to do this with their content.

Singh (2024) recommends experimentation to determine the right chunk size, stating, “The ideal chunk size varies depending on the use case and can be small, medium, or large. The only way to determine the right size is through experimentation and validation.”

Some chunking strategies involve configuring your model to parse your documents in a certain way. If your RAG model is open source, allowing customization by your development team, you may experiment with some common types of chunking strategies:

- **Sliding window chunking** (Sharma, 2024), where the model digests content in fixed-size, overlapping chunks.
- **Document structure-based chunking** (Chawla, 2024; Raghunaathan, 2024; Sharma, 2024), where the model digests the inherent structure of your content using headings, sections, and paragraphs.
- **Semantic chunking** (Chawla, 2024; Raghunaathan, 2024; Sharma, 2024), where the model segments your content by semantic similarity among sentences, paragraphs, or sections.

Understanding how your model parses your documents can help guide how you structure your documents to meet system needs while keeping information coherent and logically segmented. These strategies stress the importance of balancing chunk size to meet machine performance while capturing the essential context in each chunk.

Self-Contained Chunks

Several sources also discussed the idea of self-contained chunks of information, meaning all chunks of content contain enough information to make sense on their own (Crowhurst & Poulstrup, 2024; Gomez, 2024; Gurtam, 2024; Solis, n.d.; Sorensen, 2024b). If your organization uses a component content management system (CCMS) like DITA or XML, you are already following this best practice. These systems encourage self-contained topics by design.

When considering how to chunk content effectively, Maghunaathan (2024) cautions, “It’s not merely about reducing size; it’s also crucial to ensure that each chunk contains sufficient information and context to be meaningful independently.”

Gomez (2024) describes creating “end-to-end” content because it is harder for LLMs to pull content from several different sources and synthesize it accurately.

Gomez (2024) also cautions against the use of “client-side loaded content,” also known as “tabs,” where the content does not appear until clicked, but asserts that the primary issues arise when the information is not visible in the raw HTML so it can be indexed properly by AI tools. Perlmutter (2024) recommends using this type of content as a supplement to the main content. These recommendations can be considered on a case-by-case basis to determine how your information is best segmented while providing enough context to machine readers.

Crowhurst & Poulstrup (2024) describe that an AI tool “does not read from cover to cover; it looks at different sections and everything related to that section needs to be included.” This recommendation falls in line with avoiding “monolithic” content and instead breaking up content into manageable chunks with descriptive headings.

Use Meaningful Metadata

Several sources describe the importance of using meaningful metadata to help refine and improve a RAG model’s contextual understanding of your content. Again, if your organization uses a CCMS, you likely already have a rich metadata schema for your content.

Pertinent metadata will depend on the type of content you produce and the affordances of your authoring tools, but some sources describe considering “baseline,” or minimum metadata. For example, Chinchilla (2024) and Gurtam (2024) recommend using alt-text for media or non-text elements. Alt-text is not a new recommendation with AI; it is commonly discussed in the context of accessibility and screen readers. However, AI can use this information to understand what a multimedia element conveys. Another common example of metadata used by AI is the last updated or published date because it allows models to filter by recency (Singh, 2024).

Murugesan (2023) recommends adding metadata to both textual and multimedia elements to help “augment the textual content.” As with the other sources, this recommendation points to the usefulness of metadata in providing more context to machine readers about how your content is used.

Singh (2024) describes “grouping related documents” to determine what kind of metadata to use, or using AI-generated article summaries as metadata to provide

models with more context about the information contained in each article. These recommendations relate to making it easier for AI models to identify patterns in your content.

Write Scenario-Based Content

Several sources discussed making content scenario-based because this format makes it easier for AI models to match content to user queries more precisely.

Some sources recommend including a few troubleshooting FAQs for each article (Murugesan, 2024c; Sorensen, 2024b) because “they mirror how users ask questions” (Sorensen, 2024b). Similarly, Graham (2024) explains that “LLMs are best at ‘how do I’ type questions,” which can help reduce support team load.

Gomez (2024) recommends being “scenario-focused even in reference content,” further supporting the idea that scenario-based formats help machine readers match the content to the user’s intent. Benedetti (2025) cautioned against using FAQ-style content to exclusively benefit LLMs, stating, “A better way of providing FAQ-like content to LLMs is to turn questions and answers into knowledge base articles, each peppered with metadata and living within a semantically sound structure.” In this sense, user questions should guide content development rather than dominate it.

Write Explicit Semantic Context

Explicit semantic context ensures your AI model has enough context in order to synthesize information accurately. Although sources vary in the specificity of their recommendations, they generally advise clearly stating prerequisites for understanding within the content itself.

Gurtam (2024) and Murugesan (2024a) caution against the use of pronouns such as “it” or “they” because they can make it harder for LLMs to determine the correct referent. This recommendation is especially pertinent between different paragraphs or sections discussing the same concept (Gurtam, 2024). This recommendation coincides with other sources’ recommendations of making sure each chunk of content can be contextually clear on its own.

Gurtam (2024) also encourages the use of complete sentences to provide full context to the AI model. Similarly, Murugesan (2024a) recommends writing “elaborate content,” and Crowhurst & Poulstrup (2024) recommend “More information is generally better.” Murugesan (2024a) asserts that elaborate content “helps GenAI-based agents get a holistic perspective of the topic covered in your article.” These recommendations may go against common assumptions of writing concise content for humans.

Chinchilla (2024) and Crowhurst & Poulstrup (2024) also recommend providing descriptions for multimedia elements to help machine readers understand their relevance.

Explicit semantic context also applies to hyperlinks (Chinchilla, 2024). Although not a new recommendation with AI, links must describe the destination or purpose so that machine readers can accurately interpret them.

AI User Group (2023) describes the importance of supplementary context, such as including an article summary in each article to tell the bot what the article is about. AI User Group (2023) also describes the importance of supplementary context in eliminating bias in your content, such as including related topics, synonyms, and antonyms.

Enhance Backend of Model

This heuristic describes strategies implemented to the “backend” of the model rather than the content itself in order to enhance the retrieval accuracy.

Adding Primers to Your Model - LImS.txt File

The LImS.txt file was proposed by Jeremy Howard in September 2024 and popularized by Mintlify as a web standard to make web content more ingestible by AI tools (Chen, 2024). It is a type of markdown file you can add to your RAG pipeline. This tool is intended to address the challenges AI tools face in effectively parsing web content. The effectiveness of the LImS.txt file may evolve with advancements in RAG models, and it may not add value to all models. Nonetheless, complementary tools designed to enhance retrieval mechanisms may become more prevalent with RAG models.

Vector Embedding Size

Vector embeddings are numerical values representing semantic similarity and relationships among your content. Vector size is the number of contextual dimensions, represented numerically, that the AI model can consume at a time. Sharma (2024) explains that longer embeddings can lead to more contextual and accurate responses, while shorter embeddings support computational requirements. Knowing this, it is likely best to test and evaluate the optimal vector size with your particular content and RAG model.

Knowledge Graphs

A knowledge graph is a structured representation of your documentation source(s), showing the relationships among your document topics and increasing the AI model’s contextual understanding of your documents. Figure 3 shows a basic visualization of a knowledge graph. The concept of knowledge graphs is not new; Google has been using knowledge graphs to enhance structured search results for over a decade. However,

the implementation of knowledge graphs with RAG models, known as GraphRAG, is a way to utilize this concept within RAG frameworks. GraphRAG is an enhanced RAG model where the retrieval path includes a Knowledge Graph (AI Engineer, 2024).

Knowledge Graphs are an explicit, human-readable representation of your documents organized as interrelated entities (nodes) and their relationships. They make your data more understandable to business stakeholders, moving beyond the reliance on mathematical vector embedding values in traditional RAG models (AI Engineer, 2024). Knowledge Graphs provide RAG with additional context of your documents and their relationships in order to generate more accurate responses and handle more complex questions (AI User Group, 2023; Sharma, 2024). Figure 4 shows the basic RAG architecture with the addition of a knowledge graph. A key dependency of Knowledge Graphs is that they typically require a paid tool, such as Neo4j, to manage them and integrate them with your RAG model, and the type of graph that works best for your document ecosystem may vary.

Reranking Content

Reranking is another strategy that sources discussed to improve your model's accuracy (Gomez, 2024; Sharma, 2024; Singh, 2024). Reranking involves fine-tuning the model to score the relevance of particular documents higher with their respective user queries. The purpose of reranking is to address instances where the most similar content to the user's query is not the most relevant information (Sharma, 2024; Singh, 2024). Sharma (2024) recommends leveraging user feedback or interaction data to fine-tune the ranking algorithm.

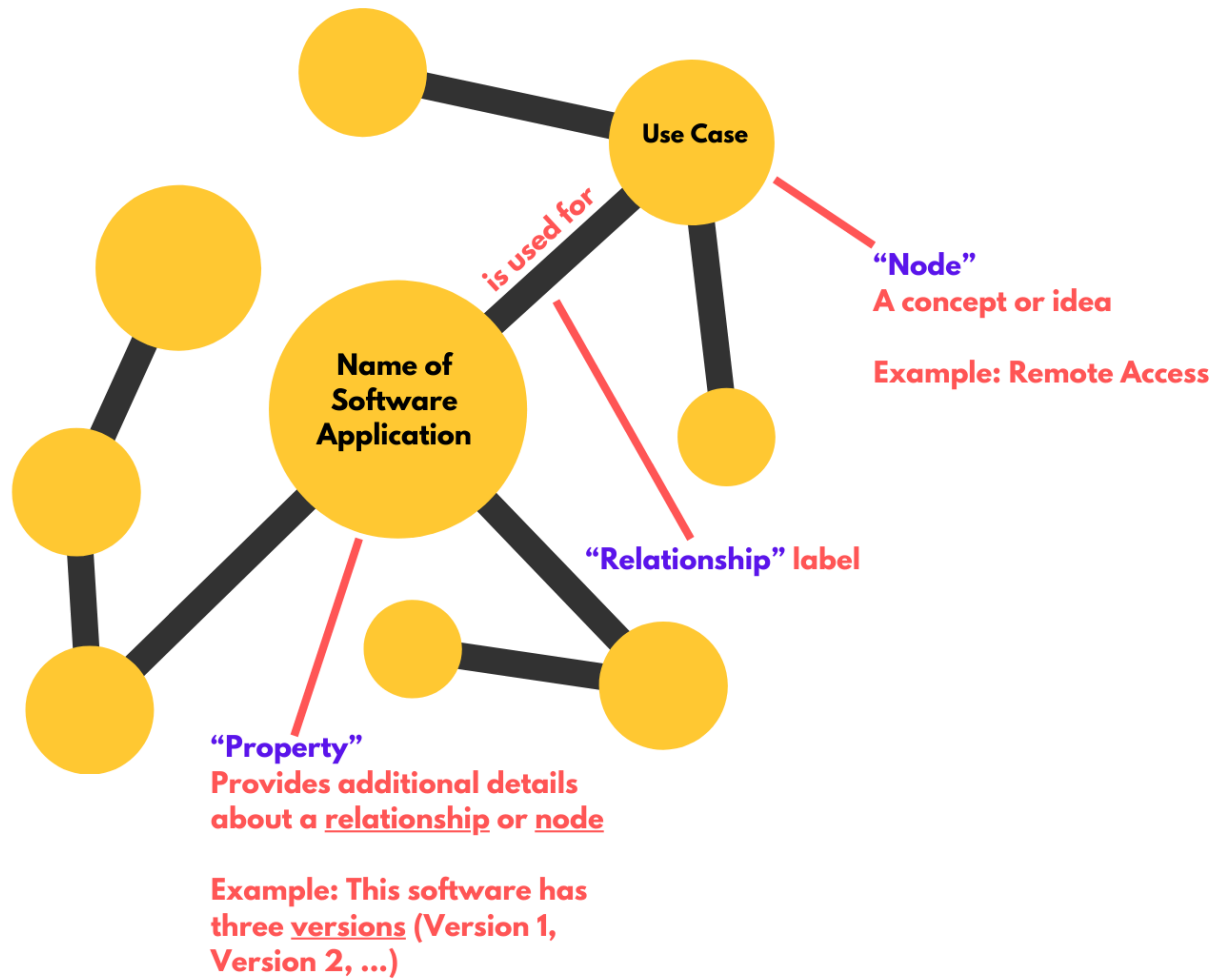


Figure 3. Simple knowledge graph architecture.

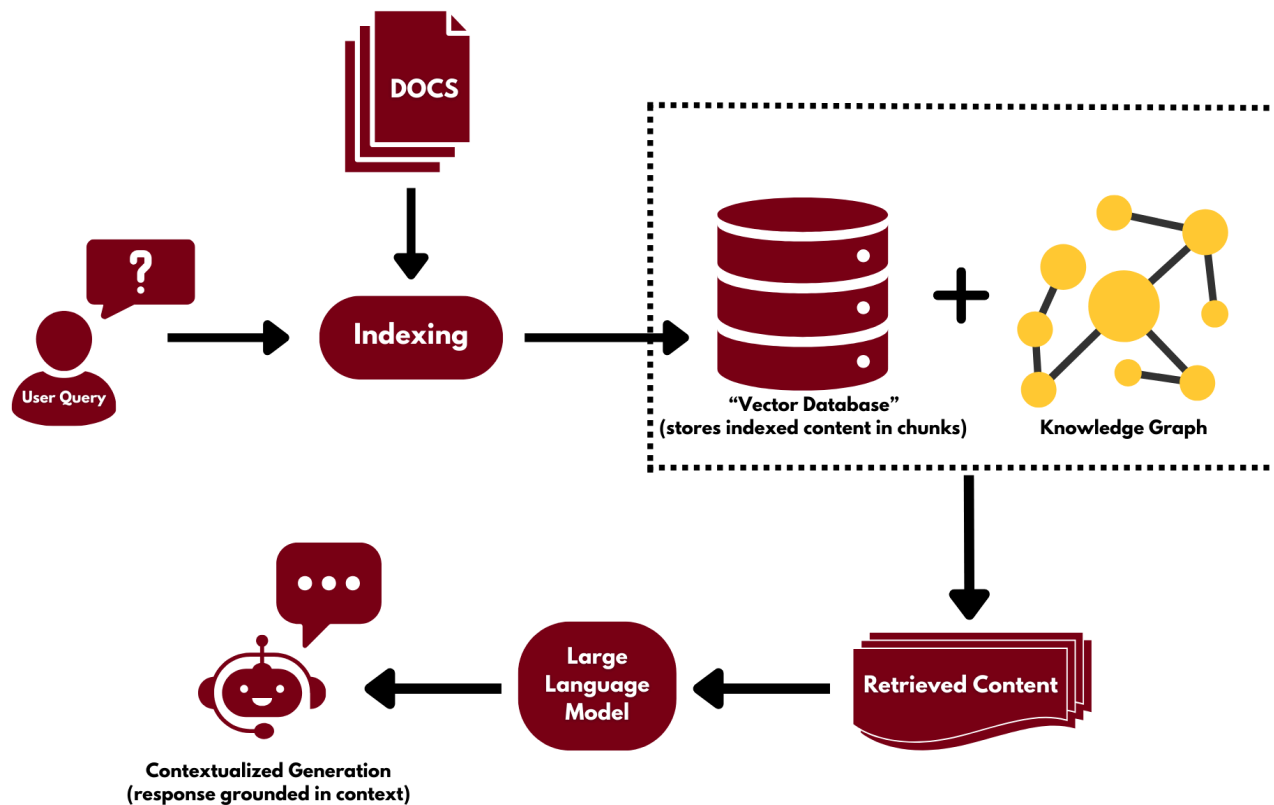


Figure 4. RAG architecture with a knowledge graph, highlighting its role in adding more context to your document ecosystem and enhancing the retrieval process.

Discussion

Some sources seem to have conflicting recommendations. For example, some sources recommend explicit semantic context, while others recommend using metadata for additional context. It may be best to consider where metadata can be used when the tag does not add value or additional information to the textual content. Explicit context seems especially important for multimedia content, such as images, videos, and tables, as AI models cannot digest these in the same way humans do (yet).

Another recommendation lacking concrete information is PDFs, including whether to use them at all in your documentation or to make sure they are structured properly. This is another opportunity to test performance with attachments and PDFs.

Two recommendations that seem to contradict each other are using descriptive links and creating self-contained information. Links are frequently used in documentation to provide users with related information or articles that are not directly related to an article's topic. You may want to consider how links can be used strategically in self-contained information so it is clear how the link might be used in conjunction with a different chunk of information.

An important consideration is that Retrieval-augmented Generation (RAG) models are not immune to hallucinations. Despite being designed to produce responses from your data source(s), they may still confidently provide incorrect information, especially when handling complex user queries or inconsistent data. For this reason, ensuring content is up-to-date, accurate, and complete may not always be enough. It is still important to have a team committed to monitoring performance and usage data to improve your documentation and the model's performance.

If your organization is considering implementing a RAG chatbot, you should consider key tool requirements for maintaining accuracy, such as source-linking capabilities in responses (Graham, 2024; Gomez, 2024) and the ability to say, "I do not know" when context is insufficient (Graham, 2024).

Most importantly, implementing a RAG chatbot with your documentation should be incorporated into your existing content strategy, enhancing the overall user experience with your documentation.

My research can be used to create broad guidelines to answer the question, **Is my documentation ready for Gen-AI?** The following criteria can be used:

1. Your content should already be meeting your users' needs.
2. Follow current best practices described in my research (test performance for specific recommendations).
3. Have a team committed to monitoring performance in production.
4. Test and fine-tune the model according to the specific performance needs of your documentation.

The following are some final considerations from my sources in implementing a RAG chatbot:

- "Check what's going on in the industry - it changes quickly" (Gurtam, 2024).
- "Technical writers need to be aware of the preprocessing steps involved with the knowledge base content while deploying any GenAI-based agents" (Murugesan, 2024c).

These sentiments urge implementation teams to understand how RAG models process content, enabling them to make informed decisions that best support end users' needs.

Conclusion

Although no universally accepted standards currently exist for structuring documentation for AI consumption, the sources I analyzed suggest many overlapping recommendations to enhance the ease with which RAG chatbot models process content. Several recommendations leave room for testing and validation with specific content types, but they can still be used to guide implementation. Further, implementation teams need to stay up-to-date on evolving strategies for integrating their docs with AI and remain committed to monitoring chatbot performance in production.

References

- AI Engineer. (2024, August 28). *GraphRAG: The marriage of knowledge graphs and RAG: Emil Eifrem* [YouTube Video].
<https://www.youtube.com/watch?v=knDDGYHnnSI>
- AI User Group. (2023, October 20). *Optimizing Chatbot UX Using Knowledge Bases*. [YouTube Video]. <https://www.youtube.com/watch?v=hHZM6vwUelc&t=99s>
- Benedetti, F. (2024, December 27). *My technical writing predictions for 2025*.
<https://passo.uno/tech-writing-predictions-2025/>
- Benedetti, F. (2025, January 24) *Should you write documentation differently for LLMs?*
<https://passo.uno/writing-for-llms-ai-chatbots/>
- Chawla, A. (2024, October 18). 5 chunking strategies for RAG. *Daily Dose of Data Science*. <https://blog.dailydoseofds.com/p/5-chunking-strategies-for-rag>
- Chen, T. (2024, November 20). Simplifying docs for AI with //lms.txt. *Mintlify*.
<https://mintlify.com/blog/simplifying-docs-with-llms-txt>
- Chinchilla, C. (2024, April 4). *Writing for robots*. [Conference presentation]. AI The Docs online conference. <https://pronovix.com/events/ai-the-docs-2024/chris-chinchilla>
- Crowhurst, S. & Poulstrup, K. (2024, April 4). *Supercharging customer integrations with Gen AI*. [Conference presentation]. AI The Docs online conference.
<https://pronovix.com/event/ai-the-docs-2024/stella-crowhurst-kristian-poulstrup>
- Feldman, P., Foulds, J. R., & Pan, S. (2024). RAGged edges: The double-edged sword of retrieval-augmented chatbots. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2403.01193>
- Gomez, N. (2024, April 4). *Using AI to drive content roadmap and documentation style*. [Conference presentation]. AI The Docs online conference.
<https://pronovix.com/event/ai-the-docs-2024/nick-gomez>
- Graham, A. (2024, April 4). *The good, bad, and ugly of AI + docs*. [Conference presentation]. AI The Docs online conference.
<https://pronovix.com/event/ai-the-docs-2024/amara-graham>
- Gurtam. (2024, September 19). *Preparing internal knowledge for AI: Making content fit for LLMs and RAG, yet human-readable*. [YouTube Video].
<https://www.youtube.com/watch?v=oXaV9rU3NM4>
- Murugesan, Selvaraaju. (2023, April 29). The future of knowledge base in the age of generative AI. *Document360*.

<https://document360.com/blog/future-of-knowledge-base-in-the-age-of-generative-ai/>

Murugesan, Selvaraaju. (2024a, January 11). Technical writing guidelines to create AI friendly content. *Document360*.

<https://document360.com/blog/technical-writing-ai-guidelines/>

Murugesan, Selvaraaju. (2024b, April 5). Guidelines for structuring tables in technical writing for GenAI-based agents. *Document360*.

<https://document360.com/blog/structuring-tables-for-gen-ai-based-agents/>

Murugesan, Selvaraaju. (2024c, May). *Is your content ready for GenAI-based agents?* TCWorld.

<https://www.tcworld.info/e-magazine/intelligent-information/is-your-content-ready-for-genai-based-agents-1316>

Raghunaathan. (2024, May 14). *Indexing and routing strategies in retrieval-augmented generation (RAG) chatbots*. Medium.

<https://blog.gopenai.com/indexing-and-routing-strategies-in-retrieval-augmented-generation-rag-chatbots-06271908e9f6>

Sharma, H. (2024, May 22). *Techniques to Enhance Retrieval Augmented Generation (RAG)*. AWS.

<https://community.aws/content/2gp2m3BJcl9mSMWT6njCIQNiz0e/techniques-to-enhance-retrieval-augmented-generation-rag>

Singh, K. (2024, April 11). *Taking RAG chatbots from POCs to production*. Medium.

<https://medium.com/@kamalmeet/taking-rag-chatbots-from-pocs-to-production-e3af4dd7d4f4>

Solis, J. (n.d.). Knowledge base best practices for generative AI. *ADA*.

<https://www.ada.cx/blog/knowledge-base-best-practices-for-generative-ai>

Sorensen, E. (2024a, March 4). Optimizing technical docs for LLMs. *Kapa.ai*.

<https://www.kapa.ai/blog/optimizing-technical-documentation-for-llms>

Sorensen, E. (2024b, April 4). *Optimizing Technical Documentation for LLMs*. [Conference presentation]. AI The Docs online conference.

<https://pronovix.com/event/ai-the-docs-2024/emil-soerensen>

Appendix A

Sample RAG chatbot interaction transcript from kapa.ai

Legend

Generative

Retrieved Content from Documentation Source

User Query: How do I know if my content is ready for a kapa solution?

Bot Response: Based on the knowledge sources, there are a few key indicators to determine if your content is ready for a kapa.ai solution:

Content Quality and Structure

The principle of "garbage in, garbage out" applies to kapa.ai. For optimal performance, your content should be:

- Clearly structured with descriptive titles and headers
- Up-to-date (avoid ingesting outdated content)
- Logically organized by topic or department
- Intentionally curated, even for internal use

As mentioned in the [Internal Assistant documentation](#), "the better quality and structure of the sources, the better Kapa's answers are." Kapa is designed to answer truthfully based on the sources it's connected to, so adding content that isn't useful or is incorrect can degrade response quality.

Content Sources

Kapa can connect to various knowledge sources, including:

- Internal wikis and documentation
- Process guidelines
- Company policies
- Public documentation relevant to your team

The accuracy of kapa depends on your content quality. As stated in the [Product documentation](#), "incorrect or incomplete answers are due to issues with existing content or missing information" in nearly all cases.

Testing Your Content

The best way to determine if your content is ready is to try kapa on your own content. You can [request a demo](#) to test kapa with your specific content sources.

This hands-on approach will give you the clearest indication of readiness.

One of kapa's strengths is its ability to help you identify gaps in your content, allowing you to continuously improve your documentation and consequently improve kapa's accuracy.

If you're concerned about content readiness, kapa provides analytics and insights to help you understand where your content can be enhanced for better accuracy.

Appendix B

ChatGPT's response to the same question from Appendix A

User Query: How do I know if my content is ready for a kapa solution?

ChatGPT's Response:

Great question — figuring out if your content is ready for a Kapa solution (or any Retrieval-Augmented Generation system) boils down to a few key points:

 **Your content is ready if:**

1. You have well-organized documentation

- Think FAQs, API docs, guides, changelogs, READMEs, or forum threads.
- Format doesn't need to be perfect, but it should be readable and coherent.

2. It's publicly accessible or can be uploaded

- Kapa can pull from public URLs (like GitHub, Notion, websites).
- Or, you can upload PDFs, markdown files, etc., directly.

3. The content is information-rich and not overly sparse

- Kapa works best when it has enough context to retrieve accurate answers.
- One-liners or super high-level marketing fluff won't help much.

4. You can update it over time

- Docs that evolve (and stay maintained) make your Kapa assistant smarter long-term.

5. The questions your users ask are answerable from your existing content

- If your support or community team keeps answering the same questions — and the answers live in your docs — you're golden.

Appendix C

Sample RAG chatbot interaction transcript from kapa.ai where the bot does not have enough context from the documentation

User Query: What kind of chunking strategy does kapa's RAG pipeline use?

Bot Response: Based on the knowledge sources provided, I don't have specific information about the chunking strategy used in kapa.ai's RAG pipeline. While the documents explain that kapa.ai uses retrieval augmented generation (RAG) and has specialized data pipelines for ingesting and transforming content, they don't detail the specific chunking methods or strategies employed in the retrieval process.

The sources mention that kapa.ai:

- Uses RAG to search an index of content to find relevant pieces of information
- Transforms content into LLM-friendly formats
- Uploads data to a vector database (Weaviate) hosted on Google Cloud
- Is optimized for answering technical product questions

However, the specific chunking approach (such as fixed-size chunks, semantic chunking, or other strategies) is not described in the provided knowledge sources.

If you need detailed information about kapa.ai's chunking strategy, I'd recommend reaching out to the kapa.ai team directly at support@kapa.ai.

Appendix D

Best Practice Heuristics and Supporting Sources

Best Practice	Supporting Sources	Total Sources
Develop Chunking Strategies	(Chawla, 2024) (Crowhurst & Poulstrup, 2024) (Gomez, 2024) (Gurtam, 2024) (Murugesan, 2024c) (Raghunaathan, 2024) (Sharma, 2024) (Singh, 2024) (Solis, n.d.)* (Sorensen, 2024b)	10
Develop Intuitive Information Architecture	(AI User Group, 2023) (Chinchilla, 2024) (Gomez, 2024) (Gurtam, 2024) (Murugesan, 2024b) (Perlmutter, 2024) (Solis, n.d.)* (Sorensen, 2024b)	8
Embrace Iterative Development	(Crowhurst & Poulstrup, 2024) (Gomez, 2024) (Graham, 2024) (Singh, 2024) (Solis, n.d.)*	5
Write Scenario-Based Content	(Bendetti, 2025) (Gomez, 2024) (Graham, 2024) (Murugesan, 2024c) (Sorensen, 2024b)	5
Use Meaningful Metadata	(Chinchilla, 2024) (Gurtam, 2024)	5

	(Murugesan, 2023) (Sharma, 2024) (Singh, 2024)	
Enhance Backend of Model	(AI Engineer, 2024) (AI User Group, 2023) (Chen, 2024) (Sharma, 2024) (Singh, 2024)	5
Write Explicit Semantic Context	(Chinchilla, 2024) (Crowhurst & Poulstrup, 2024) (Gurtam, 2024) (Murugesan, 2024a)	4

*According to Solis' LinkedIn, he worked at Ada between 2021 and 2023, which was likely the timeframe this post was published.