GapView: Measuring Knowledge Base Fitness in RAG Systems

Anonymous ACL submission

Abstract

Retrieval Augmented Generation (RAG) systems extend large language models by grounding them in external documents. However, most evaluations measure retrieval or generation quality rather than asking a more basic question of if the knowledge base itself contain the information needed to answer user questions? This work introduces GapView, a diagnostic framework that measures knowledge-base sufficiency before retrieval occurs. GapView computes cosine similarity between question and document embeddings, analyzes stability across embedding dimensions, and visualizes the resulting relationships using multidimensional scaling(MDS), polar, and one-dimensional ranked plots. Using six small synthetic datasets from programming and medical text, the results show that cosine similarity correlates with human judgments at moderate to strong levels, with correlation values ranging from 0.32 to 0.83. Dimensionality analysis reveals that below 100 dimensions, there is a loss of semantic clarity. Visualization analysis shows that MDS contributes little diagnostic value, as it fails to distinguish which questions relate to which documents. While, simpler polar and one dimensional ranked plots make numerical patterns intuitive and suggest where the knowledge base lacks sufficient information. Together GapView provides an interpretable and pre-retrieval method for detecting missing knowledge and assessing completeness of the knowledge base for RAG systems.

1 Introduction

Large Language Models (LLMs) demonstrate strong abilities in comprehending and producing text that resembles human writing, attaining notable success across various fields. Prominent examples include GPT-4 (OpenAI) (Roumeliotis and Tselikas, 2023), Llama (Meta) (Grattafiori et al., 2024), Gemini (Google) (Islam and Ahmed, 2024), Mistral (Mistral AI) (Jiang et al., 2023), and Claude (Anthropic) (Anthropic, 2024). Despite their success, LLMs like these often achieve human-level performance, but they still produce incorrect answers (Perković et al., 2024).

LLMs can produce incorrect answers when asked about events that occurred after the cutoff date of their training data or they may also generate incorrect responses when prompts are vague or ambiguous. Furthermore, if there are topics in the LLMs training data that are rare or poorly represented, they may struggle to reason about them (Matarazzo and Torlone, 2025). The standard solution to this problem is RAG (Lewis et al., 2020). RAG is a technique that enables LLMs to access and incorporate information from external sources, thereby improving the precision of LLM replies. It is a way to give LLM knowledge on demand, rather than relying on the LLM's existing training data. However, RAG systems can still fail if the knowledge base itself lacks the information required to support a question (Zhang and Zhang, 2025).

In order to address the persistent errors in RAG systems, various evaluation methods have been proposed to assess and enhance their reliability. Most existing approaches focus on retrieval or generation quality, not on whether the knowledge base itself is complete. Examples include RAGAS (Es et al., 2024) and ARES (Saad-Falcon et al., 2024), which evaluate answer quality and factuality, or separately evaluate the retriever's accuracy (Salemi and Zamani, 2024; Alinejad et al., 2024; Zhang et al., 2025; Ampazis, 2024; Li et al., 2024; Shi et al., 2024), and the generator's ability to use retrieved information in its output (Liu et al., 2023; Chen et al., 2024). While these STOA efforts assessed how effectively a RAG system retrieves and incorporates documents for answer generation, they all share the same assumption that the knowledge base of the system already contains the information needed to answer user questions. But, this is not

always the case. For instance, a medical assistant might be asked about a condition or topic that is not represented in its knowledge base at all, which means the system would fail even if retrieval behaves perfectly. This gap motivates the need for a simple way to check whether the knowledge base actually contains the information required to answer a question. Retriever scores depend on the retriever and its settings, while GapView checks whether the knowledge base itself has enough information before any retrieval occurs.

Prior studies show that cosine similarity between embeddings aligns with human judgments of correctness, with correlations of 0.48–0.77 (McGinness et al., 2025; Hua et al., 2025). This suggests that cosine similarity could serve as a simple quantitative way to assess whether a RAG system's knowledge base has the information needed to answer questions a user might ask.

We introduce GapView. The goal of GapView is to visually assess the fitness of a knowledge base with respect to expected user questions, long before any documents are retrieved from the knowledge base and any answers are elicited from a generative LLM. GapView uses cosine similarity between the documents in the knowledge base and the expected questions as the basis to identify which questions might need additional documents. GapView visualizes these cosine-based relationships through complementary methods of multidimensional scaling (MDS) (Saeed et al., 2018), one-dimensional ranked similarity plots, and polar distance plots. Some of these visualizations offer intuitive representations that help identify where information is dense, sparse, or missing.

We test GapView with text-embedding-3-large from OpenAI as it performed strongly on the MTEB benchmark (Muennighoff et al., 2022). The size of these embeddings is 3,072, but this embedding model follows the Matryoshka principle (Kusupati et al., 2022). As such, we also explore the lower-dimensionality embeddings, which lead to substantial computational savings, especially on large RAG databases.

We evaluate GapView on six synthetic datasets, three of them are clinical and three are programming. We decided to pick the medical and programming domain to determine if this method works for different types of language. The programming documents give clear step by step instructions, while the medical documents entail of more specialized and descriptive terminology. We use synthetic data to test this, as it helps avoid overlap with LLM training corpora (Deng et al., 2024), so the results reflect the new unseen text rather than model memorization. This setup provides a clear way to test whether GapView can identify when the knowledge base lacks the information needed to answer a question.

We assess GapView through the following research questions, and explain the corresponding evaluation methods in the experimental section:

- **RQ1 (Alignment of Cosine Similarity):** How well does the cosine similarity metric align with human judgments?

- **RQ2 (Embedding stability):** How stable is cosine-similarity measure under changes in embedding dimensionality?
  RQ3 (Visualization effectiveness): Do visualizations of cosine similarity help locate problematic gaps in the knowledge bases?

This work makes the following contributions for evaluating knowledge base fitness for RAG systems.

- Shifts the evaluation focus from retrieval and generation to knowledge base sufficiency.

- Demonstrates interpretable visualizations using polar and one-dimensional ranked plots that make cosine similarity patterns easier to interpret and reveal missing or weak areas in the knowledge base.

- Evaluates how embedding dimensionality affects semantic fidelity.

The paper is organized as follows: Section 2 describes the experimental setup; Section 3 presents the related work; Section 4 introduces the experiment; Section 5 details the results; and Section 5 summarizes the findings and concludes this paper. All material for this work is available in our Zenodo repository (Anonymous, 2025).

2 Related Work

The evaluation of RAG systems has been an active area of research, with number frameworks developed to assess either individual components of retrieval, re-ranking, and generation or the overall end-to-end performance of the system. For broader surveys, see Yu et al. (Yu et al., 2024),

Gan et al. (Gan et al., 2025), and Knollmeyer et al. (Knollmeyer et al., 2024).

Several frameworks have been proposed to evaluate (RAG) systems systematically. Ragas (Es et al., 2024) introduced an automated approach that measures performance across three complementary dimensions: context relevance, answer relevance, and faithfulness. While ARES (Saad-Falcon et al., 2024) extended this idea by combining limited human annotations with a fine-tuned lightweight LLM that serves as an evaluator along the same dimensions. Other methods focus more directly on retrieval evaluation. For instance, eRAG (Salemi and Zamani, 2024) runs the language model on each retrieved document and scores the resulting answers against the ground truth to assess retriever performance.

Alternative approaches such as Facts as a Function (FaaF) (Katranidis and Barany, 2024) and RAGElo (Rackauckas et al., 2024) also automate evaluation of RAG systems, but differ in focus. FaaF turns factual statements into callable functions to improve factual recall, and RAGElo employs the model itself as a self-judging evaluator for ranking system outputs. Together, these frameworks represent progress toward more robust and automated RAG evaluation.

Despite recent progress, most evaluation frameworks still emphasize retrieval or generation quality rather than asking a more fundamental question: is the knowledge base itself sufficient for the RAG system to answer user queries? Cosine similarity is a standard way to measure semantic relatedness in embedding spaces and has been shown to align well with human judgments of meaning. McGinness et al. (McGinness et al., 2025) found that there were correlations of 0.48–0.77 between embedding similarity and expert judgments. Hua et al. (Hua et al., 2025) observed similar patterns with cosine similarity when comparing the rationales generated by LLMs and humans. They found that models like GPT-4o and Claude 3.5 Sonnet mostly matched human scores and they produced rationales meaning their explanations were semantically similar to human reasoning. Together, these findings show that cosine similarity reliably captures semantic closeness, supporting its use as a diagnostic measure of knowledge-base fitness.

3 Experiment

This section presents our experiment showing how cosine similarity can assess knowledge-base sufficiency and embedding stability across domains.

3.1 Datasets

We generated six synthetic datasets, each consisting of one fictional document paired with about 50 questions, for a total of 300 questions. Three datasets are in the medical domain and three in programming.

Existing QA benchmarks often overlap with LLM training data, making it unclear whether models answer from the provided evidence or from memorized knowledge. To address this limitation, similar to the purpose of RepliQA (Monteiro et al., 2024), we created new document-based datasets. Because RepliQA is now included in GPT-4o training data, we avoided possible contamination by developing our own synthetic documents. The datasets we generate resemble real clinical notes or programming assignments but include inconsistent details that do not appear in any LLM training corpus. This design ensures that results depend only on the information within the constructed datasets In the proceeding sections, we describe how these datasets were built for this study.

Each document was divided into chunks small enough to fit within the model's token limit and embedded using OpenAI's text-embedding-3-large, enabling cosine similarity to be computed between each question and its most similar chunk.

3.1.1 Document Generation

Each document was based on a realistic source, either a clinical note or a programming assignment. We used the real-world template of the clinical note or programming assignment alongside the prompt:

> *"Make the following document very weird, strange, and confusing. Make it magical, wine-themed, or anything unusual—just make it weird."*

3.1.2 Question Generation and Answering

All questions were generated using GPT-4o (OpenAI, 2024) with a temperature of 0. To ensure diversity, we generated two types of questions for each surreal document. Answerable questions had answers that could be found directly in the document. Unanswerable questions reused words from

the document but required external knowledge to answer.

We used the following prompt to generate questions for each synthetic document:

> *"Write 50 questions about the following document. Include both answerable and unanswerable type of questions. Answerable means the answer is stated explicitly in the document. Unanswerable means it reuses words or phrases from the document but the answer is not in the document and would require external knowledge."*

To test answerability of these synthetic questions based on the synthetic documents, we built a temporary RAG pipeline. We computed embeddings for each document and its associated questions using OpenAI's text-embedding-3-large model (OpenAI, 2024).

Each document chunk was embedded and indexed separately using FAISS. For each question, we retrieved the top $k = 4$ most similar chunks from the vector database and provided them to GPT-4o using the prompt:

`Answer briefly. Context: {context_blocks}`

The `context_blocks` are the retrieved document chunks. The answers generated with this pipeline (using text-embedding-3-large for retrieval and GPT-4o for generation) were used only to create gold labels for human annotation. This setup ensured that the knowledge base contained the necessary information to make the correlation analysis in RQ1 valid.

The instruction for the RAG assistant was intentionally minimal to avoid guiding or biasing the LLM's behavior. By simply saying "Answer briefly" and attaching context, we observe how the LLM naturally uses the retrieved content—whether it answers correctly, guesses, or hallucinates—without being told how to reason or what to expect. This design helps isolate the LLM behavior.

3.1.3 Dataset Characteristics

Table 1 provides an overview of the datasets. Each dataset includes one document, the number of chunks into which it was divided, and the average token counts for both the document and its associated questions.

Table 1: Dataset characteristics.

Data	Chks	Doc Tok.	Q Tok.
Crawler	5	145.4	11.4
Search Engine	11	127.7	9.6
Programming Styles	7	135.0	11.0
Medical Note 1	7	112.4	10.5
Medical Note 2	8	121.2	10.3
Medical Note 3	6	124.7	10.5

3.2 Human Annotations

Six datasets were annotated to build a gold standard for GapView. The goal was to test if each generated answer was fully supported by the retrieved documents. Annotators followed detailed instructions. The annotator guidelines and annotated dataset are available in our Zenodo repository (Anonymous, 2025).

The annotated answers were generated using a pipeline that combined OpenAI's text-embedding-3-large model for retrieval with GPT-4o as the generator. Two annotators labeled each answer as covered if it was fully supported by the document which we considered to be covered. A response was labeled not covered if any part went beyond the document by adding details, assumptions, or inferences. In these cases, the question was considered unanswerable. After annotating independently, the two annotators met to review and resolve any disagreements. The final decisions were then recorded and used as the gold standard for evaluating GapView.

We computed the inter-annotator agreement using Cohen's κ to assess the consistency and reliability of the annotation process beyond chance agreement, and the scores ranged from 0.67 to 0.81 across the six datasets between the two annotators. Table 2 shows substantial to strong inter-annotator agreement across all datasets where κ is between 0.67 and 0.81, with few initial disagreements prior to the consensus round (D/A). Across all datasets, there were consistently more Covered (Cov.) than Not Covered (Not C.) labels, indicating that most questions were judged to be supported by the information in the knowledge base.

Table 2: Annotator Agreement

Data	κ	D/A	Cov.	N.C.
Crawler	0.81	4	36	14
Search Engine	0.67	6	36	14
Prog. Styles	0.80	4	37	13
Med. Note 1	0.70	3	44	6
Med. Note 2	0.67	4	41	9
Med. Note 3	0.70	5	37	13

3.3 Methods

RQ1: Alignment In order to answer the question about alignment of cosine similarity, we analyze box plots and compute the point biserial correlation between human judgments (a binary variable) and the maximum cosine similarity for each question and the chunks (a continuous variable).

RQ2:Embedding stability In order to test whether GapView's cosine-similarity measure remains stable across different embedding sizes and domains, and whether the alignment observed in RQ1 still holds under these changes.

We truncate the Matryoshka embeddings to 10, 100, and 1000 dimensions, recompute each question's maximum cosine similarity to its nearest document chunk, and correlate the results with human judgments in both programming and medical datasets. We truncate the Matryoshka embeddings because this allows us to test whether the observed correlations remain stable as the dimensionality scale changes. This helps determine if the alignment across domains holds, when embeddings are compressed, which is often done in RAG systems.

RQ3: Visualization effectiveness We use visualization to turn the complex numerical results into patterns that can be intuitively understood. Instead of scanning thousands of numerical values, researchers can see clusters, outliers, and missing regions directly. This makes the diagnostic process more interpretable by revealing where the knowledge base supports questions and where clear gaps or inconsistencies emerge, which are insights that are difficult to capture through numbers alone.

We generate three different visualizations. We use MDS, a circular polar plot, and a 1-D cosine rank plot. MDS was selected rather than UMAP, t-SNE or PCA because it showed the highest Spearman correlation with the original cosine similarities, preserving the data's structure better. The three different plots give us a different angle on the same data, helping see patterns one plot might miss. The MDS plot shows which questions are close or far form the documents. The polar plot shows the magnitude of how strong each question's match is. The 1-D plot shows how the similarity values compare across the domains.

4 Results

4.1 RQ1: Alignment of Cosine Similarity

Figure 1 and Table 3 show the relationship between the maximum cosine similarity of each question–document pair and the human judgments of whether the question was covered or not covered. The box plot in Figure 1 illustrates that covered questions generally have higher cosine similarity values, while not covered questions show lower cosine similarity scores. A few outliers appear in both the covered and uncovered groups, represented as individual points beyond the whiskers of the box plots. Although the box plot indicates that covered questions generally achieve higher cosine similarity values, the strength of the correlation between human judgments and cosine similarity varies across datasets.

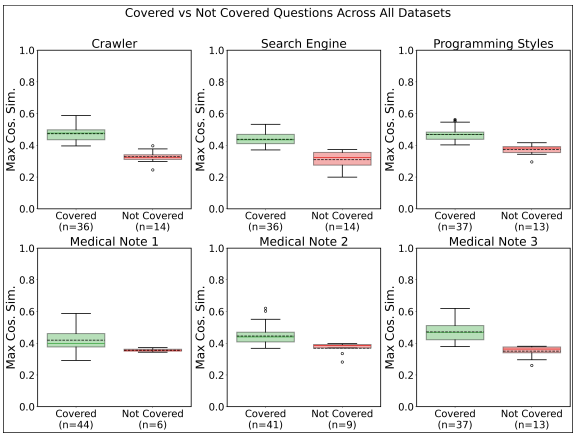

Figure 1: Cosine similarity values between the the questions and their most similar documents.

The correlation values are reported as r in Table 3[1] show that the correlations range from 0.32 (moderate correlation) to 0.83 (very strong correlation), all statistically significant (p-value $<<$ 0.05). The three datasets about programming (i.e. the Crawler, Search Engine, and Programming Styles assignments) show higher correlations, between 0.63 and 0.83, while the correlations are weaker in the three medical notes, especially Medical Note 1 (0.32) and Medical Note 2 (0.44); Medical Note 3 shows a strong correlation (0.61) but still lower than the programming texts. Across all six datasets, the p-values confirm statistical significance.

Although we do not know why the correlation values for the medical notes are lower than for the programming assignments, we speculate that it is related to the fact that the underlying embedding space has been trained on a much larger corpus of programming texts than medical material. But this is an open question for future research and

[1]We report the biserial correlation values between the continuous variable cosine similarity and the categorical variable covered/not covered.

experimenting with specialized embedding models for medicine. Our results show, however, that the correlations between cosine similarity and human judgments are at least moderate and often strong, a result that has been observed before (McGinness et al., 2025; Hua et al., 2025) but that has not yet been used for assessing the coverage of knowledge bases.

Table 3: Correlation Analysis Across Datasets

Data	**r**
Crawler	0.83
Search Engine	0.77
Programming Styles	0.63
Medical Note 1	0.32
Medical Note 2	0.44
Medical Note 3	0.61

4.2 RQ2: Embedding Stability

Table 4 shows the effects of embedding dimensionality as we start removing layers. It has a substantial effect on the stability of GapView's cosine-based signal, as we start removing semantic structure.

As we begin with the full 3072 dimensional embeddings, the correlations between cosine similarity and judgments are moderate to strong across all datasets. When the dimensionality is reduced to 1000, these correlations remain, nearly identical. Then at 100 dimensions, correlations begin to drop particular in the medical datasets. This might suggest that some detail might be lost. Once the embeddings are reduced to 10 dimensions, all correlations become very weak. In short, as we progressively remove the dimensions, the embedding space losses meaningful structure, suggesting the cosine similarity signal might remain reliable only about 100 dimensions.

Table 4: Correlation for Various Embedding Dimensionalities

Data	**3072**	**1000**	**100**	**10**
Crawler	0.83	0.80	0.73	*0.14*
Search Engine	0.77	0.75	0.31	*0.29*
Programming Styles	0.63	0.57	*0.21*	*0.14*
Medical Note 1	0.32	0.34	0.52	*0.06*
Medical Note 2	0.44	0.45	0.37	*-0.19*
Medical Note 3	0.61	0.57	0.49	*0.07*

4.3 RQ3: Visualization Effectiveness

Figures 2, 3, and 4 show three visualizations of the data with the goal of facilitating the visual discovery of knowledge gaps. In all plots, green points represent covered questions, and red points represent uncovered questions. If blue points appear in the plots, they correspond to document chunks.

Figure 2 shows an MDS plot where dissimilarity matrices were initialized to the complement of the pair-wise cosine similarities (i.e. $1 - CosSim(a, b)$) of questions and documents. We find that this visualization offers limited structural clarity and heavy overlap between the not covered and covered questions. The plots are confusing, as there is no clear indication of which document is closet to which question.

Figure 3 omits the documents altogether, and shows only the questions in polar charts. The distance of each dot to the center is the cosine similarity between the corresponding question and its closest document: the closer a question is to the center, the closer it is from a document, and, conversely, the farther away, the farther it is from any document. With these polar plots it is much easier to see which questions might need additional documents, simply by eye-balling the distances from the center.

Figure 4 shows simple scatterplots of the cosine similarity between each question and its closest document, in decreasing order. This simple visualization is also effective for quickly identifying which questions – the ones on the right tail – might need additional documents.

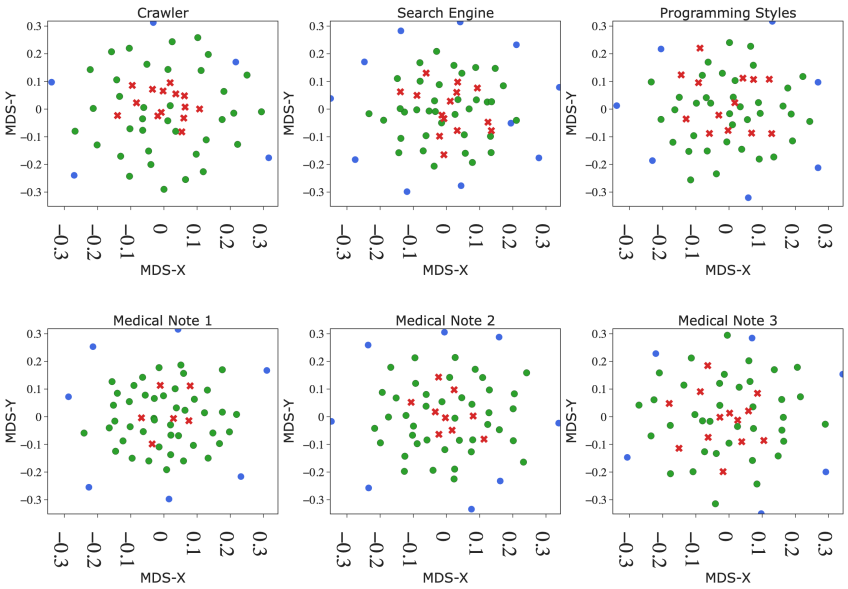

Figure 2: MDS Visualization

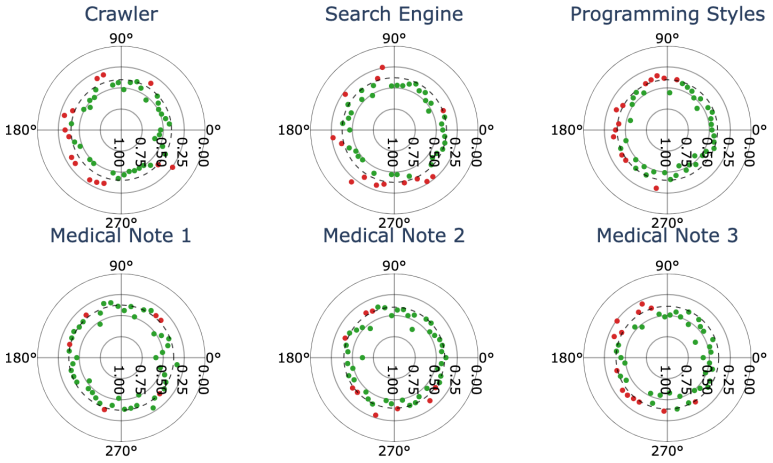

Figure 3: Polar Visualization

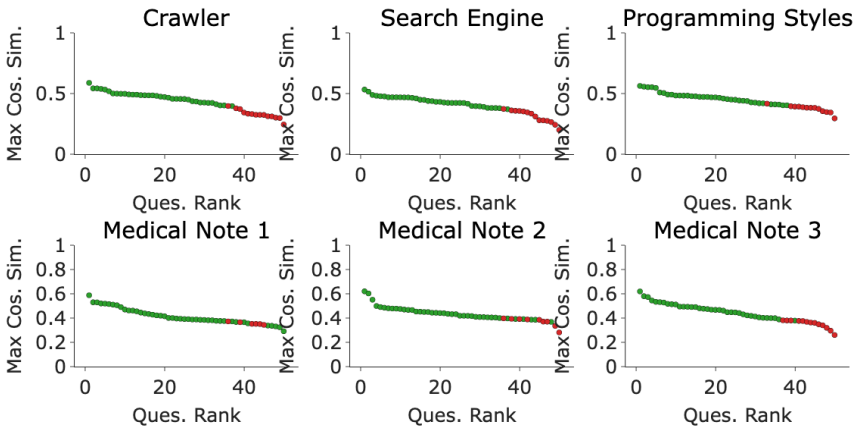

Figure 4: 1-D Cosine Visualization

5 Conclusion

This study explored a simple way to measure how well a knowledge base supports questions before retrieval or generation. By comparing question and document embeddings through cosine similarity, examining how this relationship changes with dimensionality, and visualizing it from several perspectives, the analysis shows that embedding structure meaningfully reflects the presence or absence of information.

Dimensionality affected how well the embeddings preserved relationships between questions and the documents. When the embeddings were truncated below 100 dimensions, the correlation between cosine similarity weakened, as much of the underlining semantic structure was lost from the embeddings. At 100 dimensions, the relationship improved and became more stable. For example, covered questions appeared closer to their related document chunks, while not covered questions remained farther away. This level of representation starts to maintain enough structure in the embedding space for similar questions and documents to remain close together and unrelated ones to stay apart, preserving the spatial patterns where knowledge base is strong or is lacking in certain areas.

Across domains, the programming datasets had a tighter relationship with the question and the documents. Whereas the medical datasets, showed greater variability. This could be due to the linguistic diversity and inconsistent terminology of medical text. These patterns suggest that cosine similarity can serve as a signal for detecting where a knowledge base provides strong support and where there are gaps.

Visualization added an interpretable view of these numeric patterns. MDS offered little diagnostic value, since overlapping points made it hard to see which questions were related to which documents. The polar and one-dimensional ranked plots displayed the same information more clearly, showing where information is sufficient and where clear gaps remain before invocation of the RAG pipeline.

Overall, results show that cosine similarity and low-dimensional visualization together could offer a way to assess the completeness of a knowledge base before retrieval begins. A natural next step for future work is to extend this work to new synthetic corpora designed for multi-hop reasoning, where questions depend on combining information from multiple documents. Studying these multi-hop relationships would show whether the same embedding-based approach can capture deeper connections across sources and provide a broader view of knowledge completeness in more complex retrieval settings.

6 Limitations

We were unable to extend the analysis to additional embedding models, as the GapView evaluation workflow requires human annotators to review question–document pairs and determine whether each question is supported by evidence in the knowledge base. Due to time constraints, annotators did not have sufficient bandwidth to complete additional assessments on more data.

References

Ashkan Alinejad, Krtin Kumar, and Ali Vahdat. 2024. Evaluating the retrieval component in llm-based question answering systems. *arXiv preprint arXiv:2406.06458*.

Nicholas Ampazis. 2024. Improving rag quality for large language models with topic-enhanced reranking. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 74–87. Springer.

Anonymous. 2025. Gapview experiment.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719, Mexico City, Mexico. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Aoran Gan, Hao Yu, Kai Zhang, Qi Liu, Wenyu Yan, Zhenya Huang, Shiwei Tong, and Guoping Hu. 2025.

Retrieval augmented generation evaluation in the era of large language models: A comprehensive survey. *arXiv preprint arXiv:2504.14891*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Haowei Hua, Hong Jiao, and Dan Song. 2025. Comparison of scoring rationales between large language models and human raters. *arXiv preprint arXiv:2509.23412*.

Raisa Islam and Imtiaz Ahmed. 2024. Gemini-the most powerful llm: Myth or truth. In *2024 5th Information Communication Technologies Conference (ICTC)*, pages 303–308. IEEE.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Vasileios Katranidis and Gabor Barany. 2024. Faaf: Facts as a function for the evaluation of generated text.

Simon Knollmeyer, Oğuz Caymazer, Leonid Koval, Muhammad Uzair Akmal, Saara Asif, Selvine George Mathias, and Daniel Großmann. 2024. Benchmarking of retrieval augmented generation: A comprehensive systematic literature review on evaluation dimensions, evaluation metrics and datasets. In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024)-Volume 3*, pages 137–148. SciTePress.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and 1 others. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Mingda Li, Xinyu Li, Yifan Chen, Wenfeng Xuan, and Weinan Zhang. 2024. Unraveling and mitigating retriever inconsistencies in retrieval-augmented large language models. *arXiv preprint arXiv:2405.20680*.

Yi Liu, Lianzhe Huang, Shicheng Li, Sishuo Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Recall: A benchmark for llms robustness against external counterfactual knowledge.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*.

Lachlan McGinness, Peter Baumgartner, Esther Onyango, and Zelalem Lema. 2025. Highlighting case studies in llm literature review of interdisciplinary system science. In *Australasian Joint Conference on Artificial Intelligence*, pages 29–43. Springer.

Joao Monteiro, Pierre-Andre Noel, Etienne Marcotte, Sai Rajeswar Mudumba, Valentina Zantedeschi, David Vazquez, Nicolas Chapados, Chris Pal, and Perouz Taslakian. 2024. Repliqa: A question-answering dataset for benchmarking llms on unseen reference content. *Advances in Neural Information Processing Systems*, 37:24242–24276.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

OpenAI. 2024. New embedding models and api updates. https://openai.com/index/new-embedding-models-and-api-updates. Accessed: 2025-07-29.

OpenAI. 2024. Say hello to GPT-4o. https://openai.com/index/hello-gpt-4o/. Accessed 2025-07-29.

Gabrijela Perković, Antun Drobnjak, and Ivica Botički. 2024. Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pages 2084–2088.

Zackary Rackauckas, Arthur Câmara, and Jakub Zavrel. 2024. Evaluating rag-fusion with ragelo: an automated elo-based framework. *arXiv preprint*.

Konstantinos I Roumeliotis and Nikolaos D Tselikas. 2023. Chatgpt and open-ai models: A preliminary review. *Future Internet*, 15(6):192.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. Ares: An automated evaluation framework for retrieval-augmented generation systems.

Nasir Saeed, Haewoon Nam, Mian Imtiaz Ul Haq, and Dost Bhatti Muhammad Saqib. 2018. A survey on multidimensional scaling. *ACM Computing Surveys (CSUR)*, 51(3):1–25.

Alireza Salemi and Hamed Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400.

Yunxiao Shi, Xing Zi, Zijing Shi, Haimin Zhang, Qiang Wu, and Min Xu. 2024. Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems. *arXiv preprint arXiv:2407.10670*.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pages 102–120. Springer.

Jintao Zhang, Guoliang Li, and Jinyang Su. 2025. Sage: A framework of precise retrieval for rag.

Wan Zhang and Jing Zhang. 2025. Hallucination mitigation for retrieval-augmented large language models: A review. *Mathematics*, 13(5):856.