Mind the Gap: Measuring Knowledge Gaps in RAG Pipelines

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) systems aim to improve the reliability of answers by incorporating information from external sources. The value of RAG depends on how well the knowledge base meets users' information needs. However, most existing evaluation methods for RAG pipelines focus on the quality of the generated answers or the precision of the retriever, without assessing whether the knowledge base itself contains the needed information. RAG benchmarks are typically created by generating questions directly from the documents in the knowledge base, which may not reflect the diversity of real user questions. We introduce GapView, a framework for evaluating whether the knowledge base in a RAG pipeline provides sufficient coverage to support expected user questions. GapView uses cosine similarity between embeddings and 2D Multi-Dimensional Scaling (MDS) projections to check whether a question is semantically aligned with any document in the corpus. We evaluated it on six synthetic datasets from clinical and programming domains. Results show that GapView achieves high F1 scores ($\geq$ 0.93) in predicting coverage and reveals domain-specific performance differences. Unlike traditional RAG metrics, GapView identifies knowledge gaps and provides clear visualizations that reveal where information is missing. Our findings highlight the importance of validating knowledge base coverage in RAG pipelines and offer a scalable method for flagging unsupported questions before they go through the RAG pipeline.

1 Introduction

Large language models (LLMs) have advanced the field of Natural Language Processing (NLP), as they demonstrate strong capabilities in understanding and generating human-like text and have achieved remarkable success in numerous domains (Mai et al., 2024). 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) (The). Although LLMs like these have enough knowledge to compete with human performance, they still produce the wrong answer (Perković et al., 2024).

LLMs will produce the wrong answer when they are unable to answer questions about events that have occurred after they were trained. They may also generate incorrect responses when the prompt includes vague phrasing, which can lead the model to make unsupported assumptions. Furthermore, if topics in their training data 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 allows LLMs to access and incorporate information from external sources to improve the accuracy and relevance of LLM responses. It is a way to give LLM "new knowledge" on demand, rather than relying on the LLM's existing training data. Unfortunately, RAG systems can still fail due to poor retrieval or noisy context, which can lead to the generation of inaccurate text (Zhang and Zhang, 2025). To deal with the ongoing problems of wrong answers in RAG systems, different evaluation methods have been proposed to check and improve their reliability.

Recent approaches in the literature have either assessed an entire RAG system using tools such as 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 efforts assess how effectively a RAG system retrieves and incor-

porates documents during answer generation, they often assume that the underlying knowledge base is complete.

To the best of our knowledge, current evaluation methods do not examine whether the underlying knowledge base actually contains sufficient information to support the types of questions users may ask. Yet evaluating whether the knowledge base includes the necessary content is essential for ensuring reliable answers to user questions. This is particularly critical in high-stakes domains such as healthcare, law, and scientific research, where missing information can result in unsafe or misleading outcomes. Therefore, there is a pressing need for tools that can diagnose blind spots in the knowledge base itself to ensure more reliable RAG systems.

In order to address this limitation, we investigate whether it is possible to detect if the documents in the knowledge base of a RAG system contain the information needed to support user questions - even before passing them to the LLM. While prior work often assumes the knowledge base covers the type of user questions a RAG system might receive, we argue that validating this coverage is critical in RAG system development. Gaps in the knowledge base can cause wrong or incomplete answers, even if the system works well. For example, if a RAG system lacks information about a rare drug, the LLM might still generate a confident but incorrect or incomplete answer.

This raises the fundamental question: Does the knowledge base contain the necessary content to support the questions being asked? In order to answer this question, we introduce GapView, a framework that explicitly evaluates the sufficiency of the knowledge base. Unlike prior work that focuses on retrieval precision or generation quality, GapView takes a different approach: it directly evaluates whether the knowledge base contains sufficient information to answer user questions.

GapView operates by projecting document and question embeddings into a shared space using the dimensionality technique of Multidimensional Scaling (MDS) (Saeed et al., 2018) to preserve pairwise distances and help visualize the relationship between documents and questions. Cosine similarity is used to quantify how close each question aligns with the document clusters. Questions that appear far from any document cluster in the embedding space and have low similarity scores are labeled as "not covered", signaling potential knowledge gaps. This approach provides both visual and quantitative evidence of whether the RAG system's knowledge base contains enough information to support user questions before any generation takes place.

To test this method, we create six synthetic datasets composed of fictional documents and questions where we control whether each question is answerable. We use synthetic data instead of existing benchmarks, which often overlap with LLM training data and do not clearly indicate if a question can be answered from the documents alone (Deng et al., 2024). This setup ensures we can identify failures due to missing information, not memorization or generation. It allows us to evaluate whether GapView can detect when a question is unsupported by the knowledge base.
To assess the effectiveness of GapView, we ask the following three research questions:

- **RQ1:** Can GapView correctly predict whether a document contains enough information to answer a question?
- **RQ2:** Does GapView perform consistently across domains, such as programming and clinical notes?
- **RQ3:** Do the 2D MDS visualizations preserve semantic relationships between documents and questions?

With these research questions, we show that GapView has the potential to detect whether a knowledge base can support user questions before generation. We demonstrate the effectiveness through cross-domain experiments using synthetic datasets and confirm the usefulness of the 2D visualizations for interpreting coverage and detecting potential knowledge gaps.

The remainder of this paper is structured as follows: Section 2 reviews the related work. Section 3 describes the GapView framework. Section 4 outlines the experimental design of using GapView. Section 5 presents the experiment results. Section 6 discusses the results from the experiment. Section 7 concludes and Section 8 describes the study's limitations.

GapView makes the following two contributions:

- **Coverage Prediction:** It uses cosine similarity to determine whether the knowledge base contains sufficient information to answer a question before the RAG pipeline is invoked.

- **Visual Diagnostics:** It applies the same signal to create 2D MDS plots to show how well questions align with documents.

2 Related Work

Prior work has introduced a variety of different frameworks to evaluate RAG systems from different perspectives. RAGAS (Es et al., 2024) and ARES (Saad-Falcon et al., 2024) used data generated from an LLM to evaluate the contextual relevance, answer relevance, and faithfulness. Salemi and Zamani (Salemi and Zamani, 2024) proposed eRAG, which can evaluate retrievers by running the LLM on each retrieved document, scoring its output against the ground truth, and aggregating the results with ranking metrics. Alinejad (Alinejad et al., 2024) introduced LLM-retEval, a framework designed to evaluate the retriever component in a RAG system. LLM-retEval evaluated a retriever by comparing answers generated from retrieved and gold documents using the same LLM, and scoring their similarity with another LLM that gives a binary judgement.

Li et al. (Li et al., 2024) found that RAG models will give different answers depending on which retriever they use, so they proposed using an Ensemble of Retrievers that picks and combines the best retrievers to give the most reliable answer. Building on this direction, Shi et al. (Shi et al., 2024) then proposed a four-module framework to improve the accuracy of RAG systems. Ru et al. then proposed RAGChecker (Ru et al., 2024) which breaks down both the answer and ground truth into claims, then checks if each claim is supported by the retrieved documents to determine if the error is due to the retriever, the generator, or both.

To the best of our knowledge, most prior work related to evaluating RAG systems has focused on improving retrieval precision, output quality, or overall system performance. These methods typically assume that the knowledge base already contains relevant information and evaluate how effective the system retrieves or incorporates it during generation. However, this assumption may not hold in real-world applications, where coverage gaps are common. In contrast, this study focuses on whether it is possible to identify, before generation in a RAG system pipeline, when the knowledge base lacks the information needed to support a user's question.

3 GapView Framework

3.1 Motivation and Purpose

RAG systems rely on the assumption that the underlying documents for their knowledge base contain the information needed to answer user questions. However, if the knowledge base lacks coverage for a given question, even the best retriever cannot produce a useful answer. To address this, we introduce GapView. GapView is a lightweight diagnostic tool that checks whether the knowledge base can support question answering before generation. This helps detect missing knowledge early in the RAG pipeline improving reliability.

3.2 System Overview

The GapView framework operates on a set of pre-embedded documents and questions. This framework can work with any embedding model, such as OpenAI, BERT, or any domain-specific alternatives. It normalizes the embeddings, compares each question to the documents using cosine similarity, and predicts whether the knowledge base contains enough information to answer the question before generation. To visualize these predictions, embeddings are projected into 2D space using MDS, with questions color being coded by coverage.

We define a question as "covered" if the required information is clearly present in the document—either explicitly stated, paraphrased, or framed within the document's fictional or surreal context. A question is labeled "not covered" if it includes any details not found in the document.

3.3 Framework Processing Steps

Each step below describes a component of the preprocessing pipeline that can be reused across different datasets.

3.3.1 Preprocessing and Embedding

We first load the document and question embeddings and normalize them so they can be fairly compared using cosine similarity.

3.3.2 Support Prediction

For each question, we compute the maximum cosine similarity score to any document in the knowledge base, representing how closely the question embedding aligns with its most relevant document. To decide whether a question is covered, we test 100 threshold values evenly spaced between the lowest and highest of these maximum similarity

scores across all questions. For example, if the scores range from 0.50 to 0.91 and 0.80 yields the best F1, then all questions with scores ≥ 0.80 are labeled as covered.

3.3.3 2D Projection with MDS

As part of our framework, we project all document and question embeddings into 2D space to visually inspect semantic alignment. Although we experimented with dimensionality reduction methods such as t-SNE and UMAP, we found that Multidimensional Scaling (MDS) best preserved the relative distances between embeddings based on cosine similarity. Accordingly, MDS is used as the default projection method in GapView.

In the resulting plots, documents represented as chunks from the knowledge base are shown as blue dots. Questions predicted as covered appear as green dots positioned near their closest document. Questions predicted as not covered are shown as red X's. Figure 1 provides an example of this MDS-based visualization. This plot illustrates how GapView predicts semantic alignment between questions and documents. The covered questions cluster near the relevant documents, while not covered questions appear farther away. We revisit this visualization in Section 5 to analyze the trends across all six datasets.

4 Experiment

4.1 Datasets

We generated six synthetic datasets, each consisting of a fictional document paired with approximately 50 questions, totaling 300 questions. Three datasets are in the medical domain and three in programming. Each document was based on a realistic source document of either a clinical note or a programming assignment. We utilized the real-world template alongside the prompt: "Make the following document very weird, strange, and confusing. Make it magical, wine-themed, or anything unusual—just make it weird."

We created fictional synthetic documents to test GapView because existing benchmarks often overlap with LLM training data (Lin et al., 2022) and fail to indicate whether questions are answerable from the provided documents (Jiang et al., 2021). This approach retained structural realism while introducing intentionally surreal content to avoid overlap with LLM training data, as this will help GapView spot missing information using documents that were not memorized by LLMs.

A related benchmark RepliQA (Monteiro et al., 2024), was introduced to ensure models answer questions using the given document, not their training knowledge. However since RepliQA is now part GPT-4o's training data, we adopted a similar strategy through the process described above. All questions and corresponding answers were generated using GPT-4o (OpenAI, 2024) with a temperature of 0. To ensure a mix of answerable and unanswerable questions, we used the following prompt to generate questions for each synthetic document:

> *"Generate a diverse set of factual questions someone might ask about this document and its general topics, including both questions that can be answered using the document and those that cannot. Make sure that the questions that cannot be answered by the document use words in it."*

Answerable questions could be answered solely using the document, while unanswerable ones were designed to use its language but may require external knowledge. 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}`, where `context_blocks` are the retrieved chunks. This minimal prompt allows us to observe whether or not the model answers correctly without being instructed on how to reason. The LLM's response were then manually annotated as either covered or not covered, based on the directions we provided. The annotation procedure is described in detail in Section 4.2.

4.2 Human Annotation

In order to evaluate whether each generated answer was grounded in the content of the fictional documents, two independent annotators manually labeled each response as either covered or not covered. Annotators were given the instructions to follow a strict all-or-nothing rule: if any part of the answer exceeded what was stated in the document, it was marked as not covered. Use of external knowledge—defined as any information not

explicitly present in the document, including real-world facts, domain expertise, or common sense reasoning—was not permitted, even if the answer appeared factually correct. Answers were to be labeled as covered if the answer to the question was clearly stated, paraphrased, or explicitly framed within the surreal context of the document. The answers were labeled as not covered if they included any details not found in the document, such as assumptions, inferences, or information drawn from outside sources.

After completing their annotations independently, the two annotators met to review and resolve any disagreements. Final decisions were then recorded and used as the gold standard for evaluating GapView. We computed 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 1 shows the agreement scores and the number of initial disagreements for each dataset. These results indicate strong agreement between annotators and support the reliability of the evaluation labels.

Table 1: Annotator Agreement Summary

Dataset	Cohen's κ	Disagreements
Crawler	0.810	4
Search Engine	0.674	6
Programming Styles	0.803	4
Medical Note 1	0.696	3
Medical Note 2	0.672	4
Medical Note 3	0.703	5

4.3 Research Questions

We evaluate GapView by answering the following research questions:

- **RQ1:** Can GapView correctly predict whether a document contains enough information to answer a question?
  In order to answer whether GapView can predict coverage, we compare the cosine similarity–based coverage predictions to the human annotations across all six datasets. We report the macro-averaged F1, precision, and recall to assess how well GapView is able to predict if the knowledge base contains sufficient information before generation.

- **RQ2:** Does GapView perform consistently across the domains of programming and medicine?
  In order to determine whether GapView performs consistently across domains, we grouped the six datasets into two categories: clinical and programming (three each). We then computed the macro-averaged F1 score, precision, and recall for each dataset. We then used the computed metrics to compare performance between the two domains. For example, if the F1 scores for the clinical datasets were 0.71, 0.61, and 0.54, and for the programming datasets 0.43, 0.32, and 0.73, we applied Welch's t-test (Zimmerman, 2004) to compare the mean F1 scores across the two categories. This test was chosen as it is appropriate for small sample sizes with unequal variances and assesses whether the observed performance differences between domains are statistically significant.

- **RQ3:** Do the 2D MDS visualizations preserve semantic relationships between documents and questions?
  To determine whether the MDS visualizations maintain semantic relationships—indicating that questions are located close to semantically related documents in a 2D space—and can assist in uncovering unsupported questions, we calculate Spearman's rank-order correlation between cosine similarity in the original embedding space and the pairwise distances in the 2D MDS projection for all six datasets. The correlation strength is interpreted using the following standard thresholds: weak ($\rho < 0.30$), moderate ($0.30 \leq \rho < 0.70$), strong ($0.70 \leq \rho < 0.90$), and very strong ($\rho > 0.90$) (Hinkle et al., 2003).

5 Results

We present the results for GapView across six synthetic datasets, organized around three research questions: Prediction Accuracy (RQ1), Domain-Specific Performance (RQ2), and Visualization Utility (RQ3).

5.1 RQ1: Prediction Accuracy

GapView achieved high precision and recall across all six datasets (Table 2). The programming datasets reached near-perfect performance, with F1 scores above 0.986. In comparison, the medical datasets showed lower recall. For example, Medical Note 1 had a perfect recall of 1.000 but a

lower F1 score of 0.936, while Medical Note 3 had a lower recall but a higher F1 score of 0.986. This indicates that Medical Note 1 had lower precision than Medical Note 3.

To support these results, we show the 2D MDS projections of both the predicted and ground truth coverage (Figures 1 and 2). In domains like programming, covered questions (green dots) appear close to documents (blue dots), and predictions match ground truth well. In clinical datasets, especially Medical Note 3, questions are more spread out. Some covered questions appear far from any document and are incorrectly marked as not covered. These visualizations help explain where and why prediction errors occur, and highlight areas where alignment is more difficult. Even with these challenges, the MDS plots show that GapView distinguishes covered from not covered questions based on semantic similarity, aligning with the ground truth.

Table 2: GapView Prediction Metrics

Dataset	Prec.	Recall	F1
Crawler - Fiction	0.973	1.000	0.986
Search Engine - Fiction	0.973	1.000	0.986
Programming Styles	0.974	1.000	0.987
Medical Note 1	0.880	1.000	0.936
Medical Note 2	0.889	0.976	0.930
Medical Note 3	1.000	0.973	0.986

5.2 RQ2:Domain-Specific Performance

Table 3 shows Welch's t-test results comparing GapView's performance across the programming and medical domains. Recall and F1 score differences were statistically significant ($p < 0.05$), indicating that GapView performs differently across domains for these metrics. Precision also differed numerically—0.973 for programming vs. 0.923 for medical (Table 4)—but the difference was not statistically significant ($p = 0.0941$).

To better interpret these results, Table 4 reports the average precision, recall, and F1 scores by domain. Programming datasets achieved perfect recall (1.000), higher precision (0.973), and stronger F1 scores (0.986). In contrast, medical datasets showed slightly lower recall (0.983), precision (0.923), and F1 scores (0.951).

Table 3: Welch's t-test Results by Metric

Metric	t-statistic	p-value
Precision	-2.062	0.0941
Recall	-3.146	0.0255
F1	-3.176	0.0246

Table 4: Domain-Level Averages for GapView Performance

Domain	Precision	Recall	F1
Medical	0.923	0.983	0.951
Programming	0.973	1.000	0.986

5.3 RQ3:Visualization Utility

Table 5 reports Spearman correlations between cosine similarity in the original embedding space and pairwise distances in the 2D MDS projections. All six datasets show positive and statistically significant correlations ($p < 0.05$), meaning the MDS layout preserves the semantic distances between questions and documents from the original embedding space.

Based on the correlation strength criteria (Hinkle et al., 2003), Medical Note 3 shows a *strong* correlation ($\rho = 0.722$), while the remaining five datasets—Crawler, Search Engine, Programming Styles, and Medical Notes 1–2—fall within the *moderate* range ($0.30 \leq \rho < 0.70$). No dataset shows a *weak* correlation ($\rho < 0.30$), suggesting that semantic distances between questions and documents are consistently preserved across the two domains.

These results support the use of 2D MDS visualizations for identifying alignment and coverage gaps in the knowledge base. As shown in Figures 1 and 2, covered and not-covered questions are clearly separated based on their proximity to the most relevant documents, confirming that MDS preserves the semantic distances between questions and documents.

Table 5: Embedding Space Alignment

Dataset	Spearman	p-value
Crawler	0.6559	0
Search Engine	0.3275	0.0202
Programming Styles	0.3224	0.0224
Medical Note 1	0.5757	0
Medical Note 2	0.4288	0.0019
Medical Note 3	0.722	0

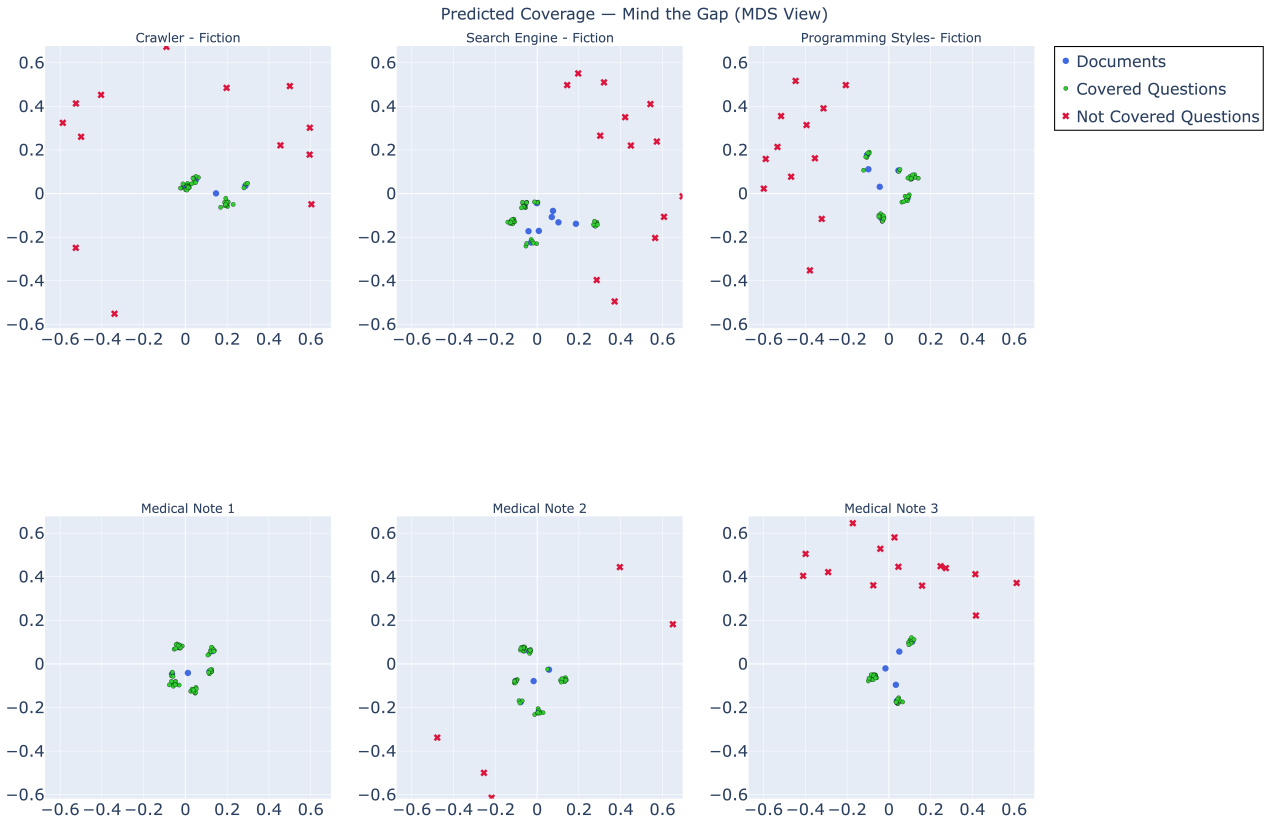

Figure 1: GapView Predicted Alignment

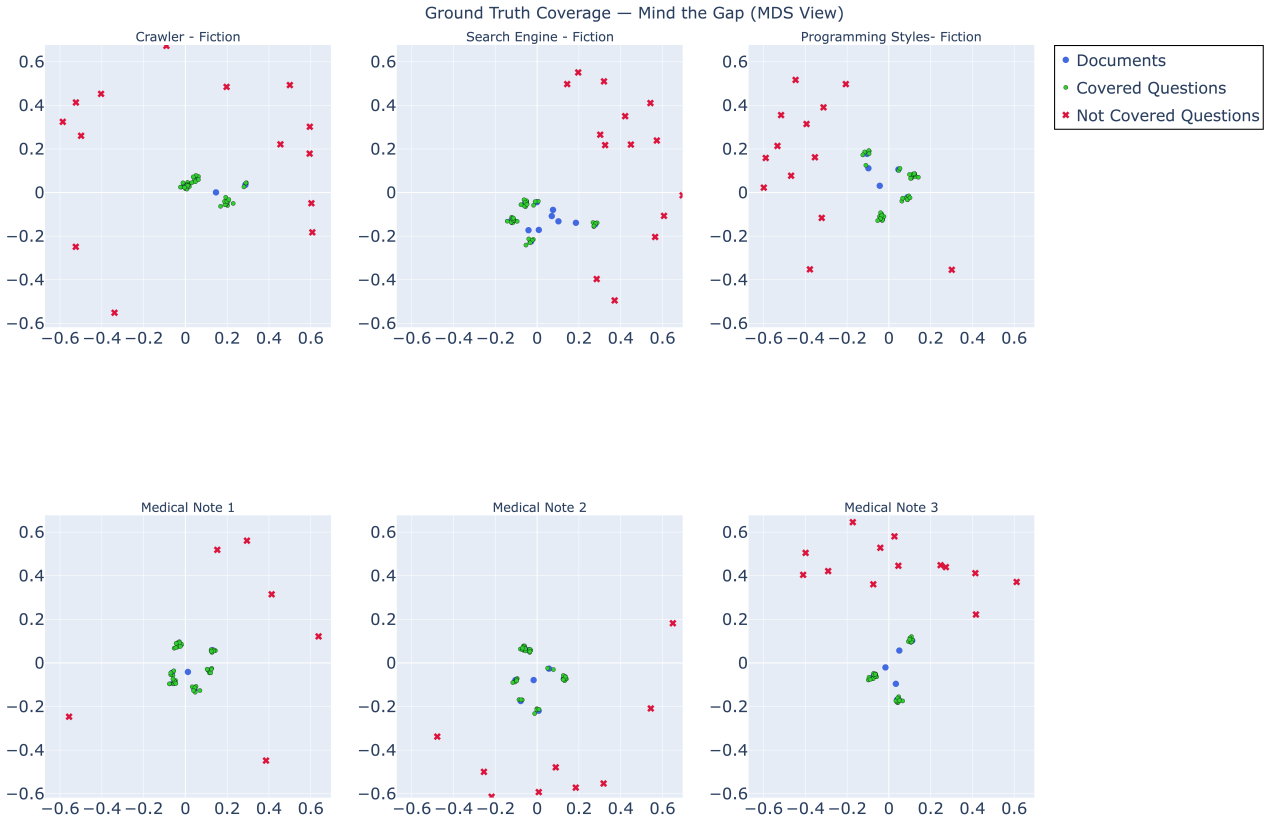

Figure 2: GapView Ground Truth Alignment

6 Discussion

GapView shifts the focus of RAG evaluation toward the knowledge base by determining whether it contains enough information to support a given question. It does so by computing cosine similarity between question and document embeddings, using this signal for both coverage prediction and visualization through 2D MDS. For each question, it finds the document with the highest similarity score and labels the question as covered if that score exceeds a tuned threshold. Across all six synthetic datasets, GapView achieved F1 scores above 0.93, with high precision and recall. These results confirm that cosine similarity is a strong signal for determining whether a knowledge base can support a user question.

Programming datasets showed higher recall and F1 scores than clinical datasets, with statistically significant differences (Table 3). Precision was also higher in programming, but the difference was not statistically significant. These differences reflect the structured, explicit nature of programming text versus the more variable language used in clinical notes. This suggests that domain-specific tuning or embeddings may improve performance.

Spearman correlations between cosine similarity and 2D distances were moderate to strong across all datasets, indicating that the MDS plots preserve the semantic distance relationships between questions and documents. The visual separation between covered and not-covered questions supports GapView's second contribution: enabling intuitive, interpretable diagnostics through 2D visualization.

In the MDS plots for each dataset (Figures 1 and 2), most covered questions in the programming datasets appeared to be closer to the document embeddings, and GapView's predictions matched the ground truth well. In contrast, clinical datasets like Medical Note 3 had some covered questions that appeared farther away and were incorrectly marked as not covered. These errors occurred even when the questions could be answered by the document. For example, the question "What pharmacogravitational agent was used to stabilize the omniliver?" was answerable from Medical Note 3 but still misclassified because it appeared far from the relevant chunks. These cases show that MDS plots can reveal not only missing information, but also prediction errors such as when answerable questions fail to align with the relevant document embeddings.

These insights are useful in real-world settings by flagging unsupported questions before system deployment. For instance, in a clinical QA assistant, GapView could detect that a question about side effects is not covered by any retrieved drug information, prompting corpus expansion before generating an answer with a RAG system. This makes GapView a practical tool for improving document coverage in RAG pipelines for high-stakes domains.

7 Conclusion

This paper introduces GapView, a framework designed to assess whether the knowledge base within a RAG system contains sufficient information to respond to user questions. GapView integrates cosine similarity with MDS visualizations to provide both accurate coverage predictions and interpretable insights. Evaluations conducted on the six synthetic datasets utilizing OpenAI embeddings demonstrate that GapView consistently identifies unsupported questions, achieving F1 scores exceeding 0.93 and uncovering performance variations specific to different domains. In contrast to conventional RAG metrics that emphasize retrieval or generation quality, GapView directly evaluates the adequacy of the knowledge base which can help improve reliability in important areas like healthcare. Its visual and numeric results have the potential to help evaluate RAG systems by showing where information is missing. As RAG systems become more common, tools like GapView will be essential for ensuring answers are grounded in sufficient knowledge. The data and code used in this study will be made publicly available upon publication to support reproducibility.

8 Limitations

This study has two limitations. The first limitation is that only two human annotators were used for labeling the synthetic datasets, which may limit the generalizability of the annotations. Although disagreements were resolved through discussion, a larger annotation pool would strengthen the reliability of the ground truth labels. The second limitation is that GapView depends on the quality of the embeddings. We used OpenAI's text-embedding-3-large, but domain-specific models like BioBERT or CodeBERT may work better for medical or programming texts, so future work could test whether any of these type of models help GapView improve alignment by better handling domain context.

References

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

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.

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.

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*.

Dennis E Hinkle, William Wiersma, Stephen G Jurs, and 1 others. 2003. *Applied statistics for the behavioral sciences*, volume 663. Houghton Mifflin Boston.

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.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

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*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

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.

Huu Tan Mai, Cuong Xuan Chu, and Heiko Paulheim. 2024. Do llms really adapt to domains? an ontology learning perspective. In *International Semantic Web Conference*, pages 126–143. Springer.

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*.

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.

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.

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

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, and 1 others. 2024. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation. *Advances in Neural Information Processing Systems*, 37:21999–22027.

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*.

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.

Donald W Zimmerman. 2004. A note on preliminary tests of equality of variances. *British Journal of Mathematical and Statistical Psychology*, 57(1):173–181.