

# Symphony: Towards Trustworthy Question Answering and Verification using RAG over Multimodal Data Lakes

Nan Tang, Chenyu Yang, Zhengxuan Zhang, Yuyu Luo  
HKUST(GZ), China

Ju Fan  
Renmin University, China

Lei Cao  
University of Arizona, USA

Sam Madden  
MIT, USA

Alon Halevy  
Google, USA

## Abstract

Large Language Models (LLMs) have revolutionized access to multimodal data lakes, enabling users to query and analyze complex information across diverse data modalities using natural language. However, their generative nature unavoidably leads to hallucinations, resulting in inaccuracies and misinformation in models like GPT, Llama, and Gemini. To address this, we introduce **Symphony**, a system designed for trustworthy question answering and verification using multimodal data lakes. **Symphony** supports two core functions: reasoning and verification. In reasoning, **Symphony** retrieves relevant information from multimodal data sources, breaks down complex queries into manageable sub-questions, and uses specialized tools (e.g., LLMs or DBMS) to generate grounded answers. For verification, it cross-checks (LLM) generated answers against trusted sources, such as private or enterprise data lakes, to enhance accuracy and reliability. By integrating these processes, **Symphony** mitigates factual inaccuracies, aligns outputs with trusted data, and adapts to a wide range of applications.

## A Introduction

The rise of **Large Language Models (LLMs)** has unlocked transformative **opportunities** across diverse domains, including natural language processing, data analysis, and creative design, among many others. By enabling interaction with complex systems through intuitive natural language queries, LLMs democratize access to knowledge and data, allowing users to obtain valuable information quickly and cost-effectively without requiring specialized expertise or manual data processing. This accessibility has driven widespread adoption across industries such as healthcare, finance, and customer service, empowering both professionals and non-experts to extract insights and make informed decisions with ease. By reshaping traditional workflows, LLMs have the potential to significantly enhance productivity and decision-making across a wide range of applications.

However, alongside these opportunities, significant **risks of LLMs** (or more generally, generative AI) have emerged. The phenomenon of hallucinations, i.e., the generation of inaccurate or misleading

---

Ju Fan is the corresponding author

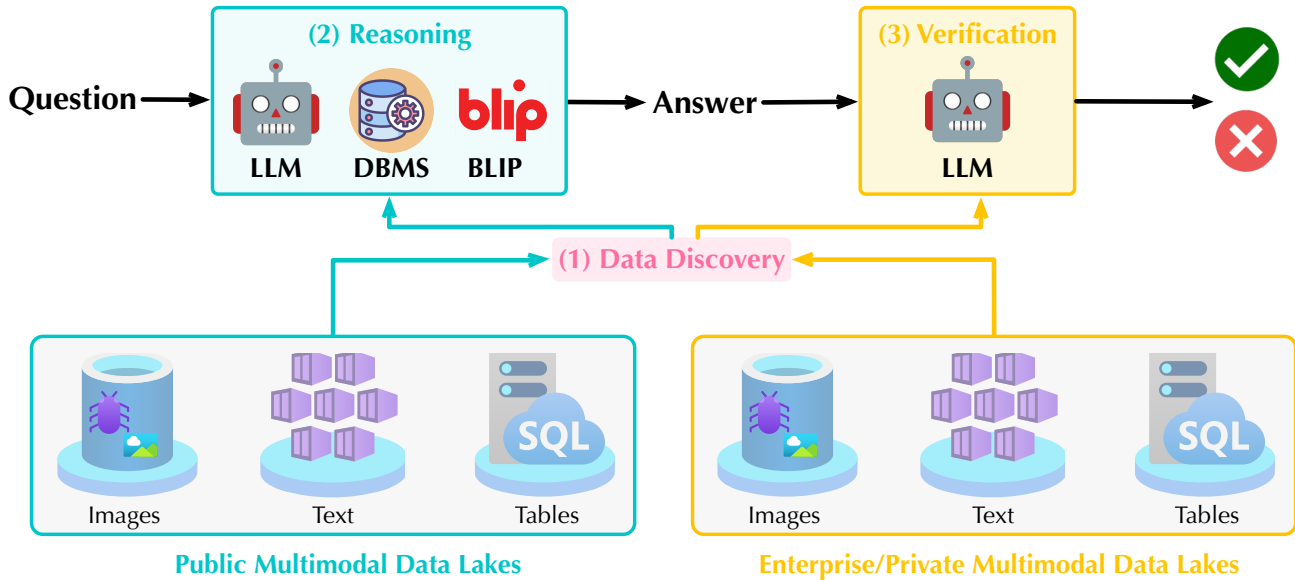


Figure 42: An Overview of **Symphony**.

information, poses a serious challenge to trust in LLM outputs. By 2023, analysts estimated that chatbots hallucinate as much as 27% of the time<sup>1</sup>, and factual errors were present in 46% of generated texts [5], underscoring the prevalence of this issue. These inaccuracies can negatively impact various aspects, including decision-making, the spread of misinformation, privacy violations, and potential legal liabilities.

Developing **trustworthy solutions for question answering** is a priority for both academia and industry, with efforts focused on improving accuracy, reducing biases, and aligning models with human values. Companies like OpenAI and Google enhance model reliability through advanced training and Responsible AI principles<sup>2</sup>. Initiatives like the Partnership on AI and collaborations by the AI Ethics Lab promote ethical guidelines and accountability in AI development<sup>3</sup>. However, challenges remain due to the probabilistic and generative nature of LLMs, which leads to unpredictability in outputs. While current efforts are valuable, they are insufficient to fully address issues such as inaccuracies, biases, and lack of explainability. More robust systems are needed, especially in high-stakes applications like data analysis over multimodal data, where trustworthiness is critical.

In this paper, we present **Symphony** [4, 17], a system designed for trustworthy question answering and data analysis over multimodal data lakes. Given a multimodal data lake  $L$  and a natural language question  $Q$  requiring factual or objective answers, the task of reasoning involves generating an answer to  $Q$  by retrieving relevant data from  $L$  and applying various reasoning tools such as LLMs, RDBMSs, or graph databases. Additionally, if an answer  $A$  is provided—whether from humans or LLMs (possibly enhanced with RAG)—the task of verification is to assess the correctness of  $A$  for  $Q$ , using the data lake  $L$  (either a public data lake or a private/enterprise data lake) to ensure factual accuracy and reliability.

As illustrated in Figure 42, **Symphony** comprises three core modules: (1) **Discovery** operates over multimodal data lakes and serves as the retrieval module; (2) **Reasoning** formulates answers to natural language questions using retrieved information and various tools; and (3) **Verification** assesses the correctness of provided answers utilizing LLMs and multimodal data lakes.

**Roadmap.** Section B describes the **Discovery** module. Section C discusses the **Reasoning** module.

<sup>1</sup><https://www.nytimes.com/2023/11/06/technology/chatbots-hallucination-rates.html>

<sup>2</sup><https://openai.com/safety/>

<sup>3</sup><https://aiethicslab.com/>

Section D presents the **Verification** module. Section E describes empirical findings. Section F identifies open problems. Section G discusses related work. Finally, we close this paper by concluding remarks in Section H.

## B Symphony: Data Discovery over Multimodal Data Lakes

Data discovery is the process of identifying relevant data files from multimodal data lakes efficiently. As shown in Figure 43, **Symphony** provides two main categories of data discovery methods: (a) word-level similarity search and (b) holistic embedding-based similarity search. These methods are chosen based on a balance between efficiency and effectiveness, enabling retrieval of relevant data from data lakes, thus supporting both reasoning and verification.

**Word-level similarity search** breaks down queries and data items in the data lakes into words (or terms), retrieves and ranks data items based on combined word-level similarity, as shown in Figure 43 (a). Methods like BM25, TF-IDF, and Jaccard similarity are used. These approaches are simple, interpretable, and computationally efficient, making them ideal for scenarios with high word overlap between the query and data items.

**Embedding-based similarity search** encodes multimodal data items and queries into high-dimensional vectors in a shared embedding space, enabling fast and precise similarity calculations. As shown in Figure 43, in this process, multimodal data (e.g., text, tables or images) are encoded into dense vector embeddings. These embeddings are then stored in a vector database (e.g., Meta Faiss), and indexed for fast similarity-based retrieval, typically using efficient distance metrics like cosine similarity or dot product. To support various data representations, **Symphony** investigates the following two encoding strategies:

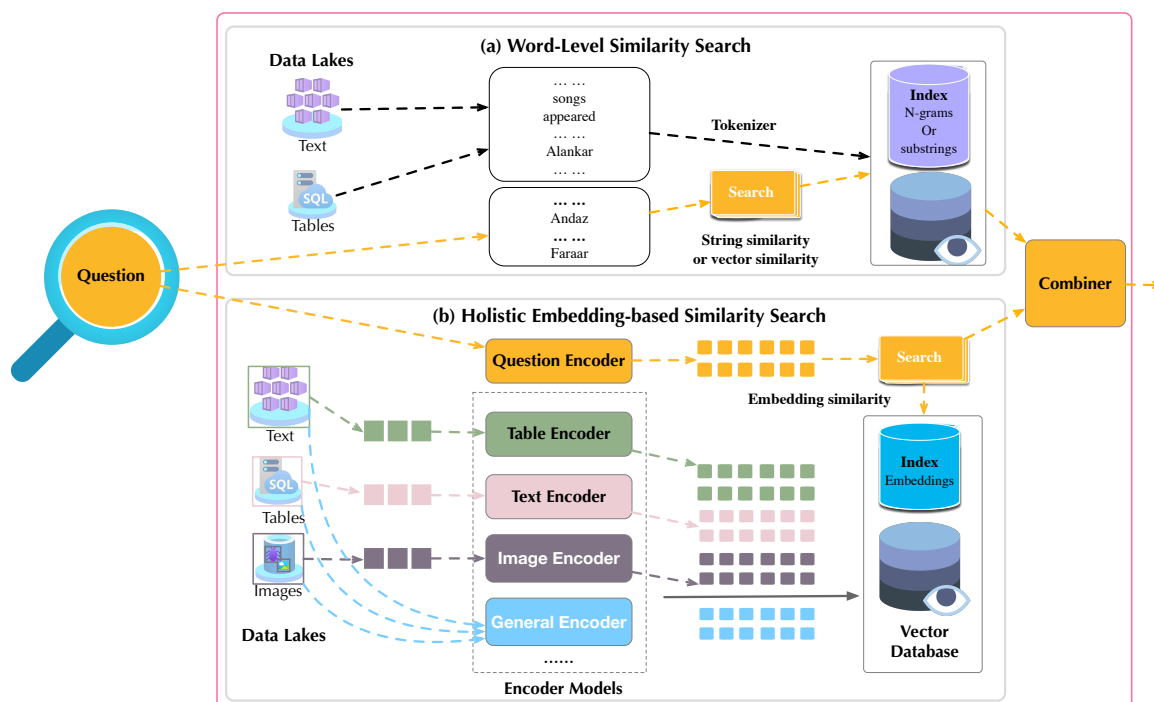


Figure 43: The **Discovery** Module.

1. **Modal-specific Representation Learning.** This strategy uses models tailored to each data type (e.g., text encoder [11], table encoder [16] or image encoder CLIP [14]), capturing unique features like intricate table structures or text nuances, making it ideal for precise retrieval within individual modalities.
2. **Modal-agnostic (Cross-Modal) Representation Learning.** This strategy learns a shared embedding space, using a General Encoder, through cross-modal representation learning (see [4] for more details), enabling similarity comparisons across modalities. This approach supports queries that can retrieve relevant items from different modalities, enhancing interoperability where cross-modal relationships are crucial.

## C Symphony: Reasoning over Multimodal Data Lakes

In this section, we present the reasoning process of **Symphony**, which integrates Large Language Models (LLMs) with Retrieval-Augmented Generation (RAG) to reason over information from multimodal data lakes. Given a natural language (NL) question, **Symphony** first retrieves top- $k$  relevant data items from the data lakes, as discussed in the previous section. **Symphony** then conducts a question decomposition strategy to address complex queries effectively, where a question can be decomposed into sub-questions, and each sub-question can be answered by different tools, such as LLMs, DBMSs, and so on.

### C.1 Question Decomposition

We propose a Question Decomposition strategy to address complex questions requiring information from multiple sources. Here, a *data source* is defined as a collection of data items originating from the same origin, such as an isolated table, a database, or a text passage. When multiple data items, like a table and passage, come from the same Wikipedia page or structured document, they are also treated as a single data source. Similarly, if two tables  $d_i$  and  $d_j$  have a predefined primary-foreign key (PK-FK) relationship, as with tables from the same database, they are merged into a unified data source  $d'_k$ . Initially, we retrieve a set of data items  $D = \{d_1, d_2, \dots, d_n\}$  from multimodal data lakes and process them using heuristic methods to form  $D' = \{d'_1, d'_2, \dots, d'_m\}$ , where  $m \leq n$  and each  $d_i \in D'$  represents a distinct data source.

The **objective** of our question decomposition strategy is to break down a complex query into simpler sub-questions to facilitate retrieval of relevant information. Ideally, each sub-question should focus on a primary data source. However, we allow flexibility for sub-questions that still require multiple sources after decomposition, accommodating scenarios where information from different sources complements or corroborates each other. This adaptable approach enhances reasoning effectiveness by balancing simplicity, which reduces each sub-question to its essential elements, with the integration of supportive data, enabling relevant details from multiple sources to strengthen the answer’s accuracy. Together, these elements minimize fusion errors and streamline the reasoning flow, creating a coherent, step-by-step resolution.

To achieve this, **Symphony** employs an iterative prompt-based approach with LLM to automate the decomposition process. In each round, the LLM generates a sub-question based on the previous sub-question and data source, or decides to terminate the process. First, an initial prompt is generated to identify the first sub-question and its corresponding data sources. **Symphony** then iteratively uses this information to generate subsequent prompts, guiding the LLM to create the next sub-question until the entire question is resolved. In cases where the LLM determines that a question cannot be effectively decomposed, **Symphony** bypasses decomposition. Instead, it directly leverages the information retrieved from multiple data sources to formulate a reasoning for the original question. For example, a question like “What is the population of France?” cannot be further decomposed into distinct sub-questions.

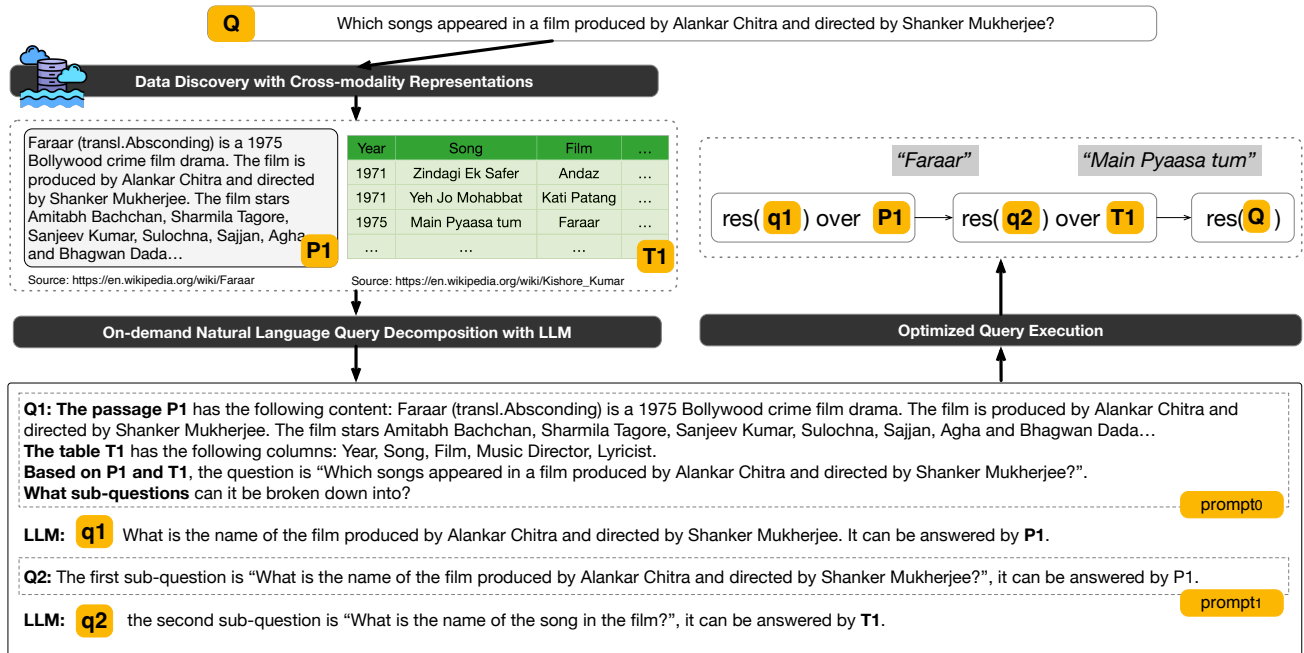


Figure 44: RAG-based Reasoning in **Symphony**

We illustrate the question decomposition process in Figure 44. Suppose we have a question **Q**: "Which song ...," with two relevant data sources retrieved from multimodal data lakes, a passage **P1** and a table **T1**. Using a template-based approach, **Symphony** constructs an initial prompt, **prompt<sub>0</sub>**, based on **Q**, **P1**, and **T1**. **Symphony** sends the prompt **prompt<sub>0</sub>** to LLM, and LLM generate a sub-question **q1** as well as the data source on which it should be utilized. Building on the first sub-question **q1** and its data source **P1**, **Symphony** uses the next prompt template to generate **prompt<sub>1</sub>**. Given **prompt<sub>1</sub>**, the LLM generates the second sub-question, **q2**, and assigns table **T1** as its data source. At this point, the LLM decides to stop, as it considers the original query **Q** fully addressed.

## C.2 Reasoning

**Question Answering using Retrieval Augmented Generation (RAG).** We leverage the powerful reasoning capabilities of large language models (LLMs) to address complex questions alongside relevant retrieved data sources. Using a prompt-based approach, we guide LLMs to conduct nuanced reasoning and generate coherent answers. If necessary, we can also prompt the LLM to provide detailed explanations of the reasoning, enhancing transparency and interpretability. **Symphony** offers NL2SQL as another way to support queries over a single table or a database. In addition to the existing NL2SQL techniques [9], **Symphony** leverages LLMs [12], as well as the prompting techniques to convert NL questions to SQL queries, using similar ideas we introduced in question decomposition.

**Sub-Answers Aggregation.** For complex questions, answers to each sub-question need to be combined accurately for a complete response. Using a prompt-based approach, the LLM sequentially integrates individual answers, rephrasing them into a coherent response to the original question. For instance, if the task involves summing values from sub-answers, the LLM aggregates these values directly, producing a reliable and context-aware final result.

**Reasoning Optimization.** To ensure both efficiency and accuracy, reasoning optimization is applied to streamline execution plans and reduce response times. **Symphony** employs a multi-objective optimizer

that balances speed with precision, choosing the best approach based on retrieved data type and question complexity. For instance, if an exact result is critical and the data is highly structured, **Symphony** prioritizes Natural Language to SQL (NL2SQL) for accuracy; otherwise, Table Question Answering (TableQA) can be used for faster, approximate answers.

This optimization framework also manages cost-performance trade-offs by evaluating the computational demands of various query methods. For high-priority questions, accurate methods are selected, while non-critical evaluating may utilize quicker, approximate options. This flexible approach enables **Symphony** to handle complex, multimodal evaluating efficiently, effectively decomposing and aggregating responses across data sources.

## D Symphony: Answer Verification over Multimodal Data Lakes

This section describes our verification approach [17]. In **Symphony**, verification occurs when an answer is provided, with the objective of ensuring its correctness. Notably, **Reasoning** and **Verification** are loosely coupled, allowing **Verification** to validate answers regardless of whether they are generated by humans, large language models (LLMs), or other tools.

During verification, this answer is used as a query to retrieve supporting or contradicting data items from the multimodal data lake, with the aim of either validating or refuting the generated information. **Symphony** employs two types of verifiers. The first type is a one-size-fits-all model, such as an LLM, which can be conveniently utilized by sending prompts directly. The second type consists of task-specific models, designed for specialized scenarios, such as PASTA [9] for verifying facts based on tables. While using LLMs by default provides simplicity, we support task-specific models for two main reasons:

1. **Data privacy:** In sensitive domains like healthcare and government, using generic, externally hosted models can risk data exposure. Task-specific, localized models mitigate this by processing data internally, protecting privacy;
2. **Improved accuracy:** Our empirical results show that task-specific models, fine-tuned on relevant datasets, generally outperform generic models for specialized tasks, providing more reliable and precise results.

An example of this verification process is shown in Figure 45. Based on an answer generated by the LLM, **Symphony** retrieves a passage titled “Stomp the Yard” and a table “Meagan Good’s Filmography” from the data lake. Using these retrieved items as evidence, the verifier identifies the generated content as incorrect and provides an explanation for the error.

The multimodal data lakes used in verification can differ from those used in reasoning, such as private or enterprise-specific data lakes, which contain domain-specific knowledge and high-quality, reliable data tailored to organizational needs. Using such data sources in the verification process not only enhances reliability but also ensures that outputs adhere to specific content standards and align with user expectations. This is particularly valuable in enterprise applications where data quality and relevance are important.

## E Experiments

We conduct preliminary experiments to demonstrate the effectiveness of **Symphony**. We evaluate its performance in two key processes: Reasoning and Verification.

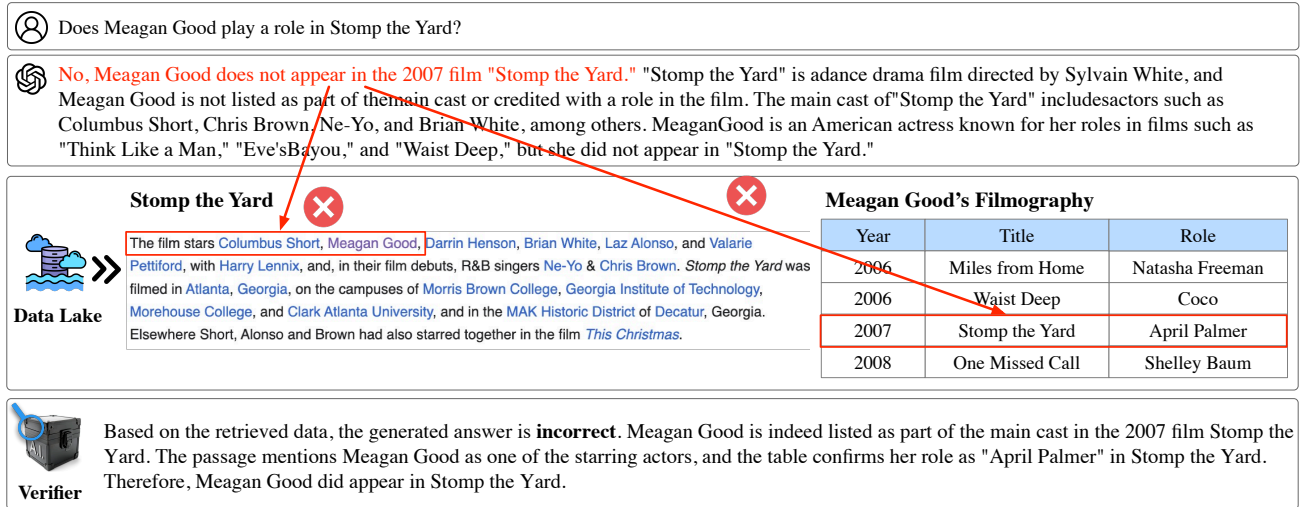


Figure 45: RAG-based Verification in **Symphony**

## E.1 Question Answering

**Experiment Setting.** In this experiment, we focus on evaluating question answering performance using a multimodal data lake consisting of 400K web tables and 6M English passages extracted from Wikipedia. The data lake includes both tables and texts, and each query is designed to retrieve relevant data items to answer a given question. We use 18 manually crafted user queries, each with corresponding ground truth annotations specifying the required data items, sub-queries for decomposition, and final answers.

**Data Discovery Evaluation.** The effectiveness of data discovery is measured using the recall at  $K$  ( $R@K$ ) metric, which calculates the proportion of relevant data items retrieved in the top- $K$  recommendations. The experimental results show that when  $K$  is 5, 10, 15, and 20, the  $R@K$  values are 40.8%, 46.3%, 59.3%, and 77.8%, respectively. For 12 out of the 18 queries, **Symphony** successfully discovers all the relevant items needed to answer the query. The remaining 6 queries show partial success. In total, 30 out of 38 related items are correctly discovered, demonstrating the potential of the proposed data discovery methodology, even though it is still in a preliminary stage.

**Query Decomposition Evaluation.** To decompose queries into manageable sub-queries, **Symphony** serializes the discovered data items and uses GPT-3 to generate sub-queries. The output includes the sub-queries and corresponding data item ids. Evaluation of the decomposition quality is based on two criteria: (1) whether each sub-query is useful for solving the original query, and (2) whether the sub-query can be answered correctly using the selected data item. The human evaluation results show that 77.8% of the queries scored 2 (both criteria met), 16.7% scored 1 (only the first criterion met), and 5.5% scored 0.

Table 22 shows the results of 8 instances. **Symphony** is able to handle different aggregation operations, such as sum (Instance 2) and comparison (Instance 3). Further, it correctly understands long sentences (Instance 1). However, **Symphony** has difficulty in dealing with sentences with complex syntactic structures (e.g., in Instance 4, GPT-3 mistakenly generates  $q_3$  because it takes the subject of "released" as Academy Award for Best Foreign Language Film").



<p><u>Instance 1:</u> <math>Q</math>: Did the 2014 United States house of representatives elections take place on the same day in Rhode Island, South Carolina and Louisiana? <math>E = \{P_1, P_2, P_3\}</math>  <math>\rightarrow q_1</math>: Did the 2014 United States House of Representatives elections take place on November 4, 2014 in Rhode Island? <math>e_1 = P_1</math>; <math>q_2</math>: Did the 2014 United States House of Representatives elections take place on November 4, 2014 in South Carolina? <math>e_2 = P_2</math>; <math>q_3</math>: Did the 2014 United States House of Representatives elections take place on November 4, 2014 in Louisiana? <math>e_3 = P_3</math>  ★ Score: 2</p>
<p><u>Instance 2:</u> <math>Q</math>: How many representatives were elected in the 2014 United States house of representatives elections in Rhode Island and South Carolina? <math>E = \{P_1, P_2\}</math>  <math>\rightarrow q_1</math>: How many representatives were elected in the 2014 United States house of representatives elections in Rhode Island? <math>e_1 = P_1</math>; <math>q_2</math>: How many representatives were elected in the 2014 United States house of representatives elections in South Carolina? <math>e_2 = P_2</math>  ★ Score: 2</p>
<p><u>Instance 3:</u> <math>Q</math>: Which number is larger, the number of living former prime ministers of Tunisia or the number of former presidents of Ghana? <math>E = \{T_1, P_1\}</math>  <math>\rightarrow q_1</math>: What is the number of living former prime ministers of Tunisia? <math>e_1 = P_1</math>; <math>q_2</math>: What is the number of former presidents of Ghana? <math>e_2 = T_1</math>  ★ Score: 2</p>
<p><u>Instance 4:</u> <math>Q</math>: What year was the first German film that won the Academy Award for Best Foreign Language Film released? <math>E = \{T_1, P_1\}</math>  <math>\rightarrow q_1</math>: What year was the film The Tin Drum released? <math>e_1 = T_1</math>; <math>q_2</math>: What was the first German film that won the Academy Award for Best Foreign Language Film? <math>e_2 = T_1</math>; <math>q_3</math>: What year was the first Academy Award for Best Foreign Language Film given out? <math>e_3 = T_1</math>  ★ Score: 0</p>
<p><u>Instance 5:</u> <math>Q</math>: Did The Tin Drum and Kolya both win the Academy Awards for Best Foreign Language Film? <math>E = \{P_1, P_2\}</math>  <math>\rightarrow q_1</math>: whether The Tin Drum has ever won the Academy Award for Best Foreign Language Film? <math>e_1 = P_1</math>; <math>q_2</math>: whether Kolya has ever won the Academy Award for Best Foreign Language Film? <math>e_2 = P_2</math>  ★ Score: 2</p>
<p><u>Instance 6:</u> <math>Q</math>: How many years elapsed between the premiere of season 10 and the premiere of season 5 of Law &amp; Order : Special Victims Unit? <math>E = \{T_1, T_2\}</math>  <math>\rightarrow q_1</math>: What is the premiere date of season 10 of Law &amp; Order : Special Victims Unit? <math>e_1 = T_2</math>; <math>q_2</math>: What is the premiere date of season 5 of Law &amp; Order : Special Victims Unit? <math>e_1 = T_1</math>  ★ Score: 2</p>
<p><u>Instance 7:</u> <math>Q</math>: Did You Let Me Walk Alone participate in the Eurovision Song Contest in 1996? <math>E = \{T_1, P_1\}</math>  <math>\rightarrow q_1</math>: What year was the Eurovision Song Contest held in Lisbon, Portugal? <math>e_1 = T_1</math>; <math>q_2</math>: What year was You Let Me Walk Alone released? <math>e_2 = P_1</math>  ★ Score: 1</p>
<p><u>Instance 8:</u> <math>Q</math>: Are the tallest building in the united kingdom and the tallest building in poland above 200 meters? <math>E = \{T_1, T_2\}</math>  <math>\rightarrow q_1</math>: What is the height of the tallest building in the United Kingdom? <math>e_1 = T_1</math>; <math>q_2</math>: What is the height of the tallest building in Poland? <math>e_2 = T_2</math>  ★ Score: 2</p>

Table 22: Example sub-queries generated by **Symphony**.  $q_i$  and  $e_i$  represent the  $i_{th}$  sub-query and its corresponding data item.  $T_i$  represents a table and  $P_i$  represents a text.




	<p><b>Claim:</b> In 1954 u.s. open (golf), the cash prize for tommy bolt, fred haas, and ben hogan was 960 in total.  <b>(Ground Truth:</b> a false claim that should be <b>Refuted</b>)</p>																																																																		
<p><b>Retrieved Evidence and Verification</b></p> <p><b>Table E1:</b> 1954 u.s. open (golf)</p> <table border="1"> <thead> <tr> <th>place</th> <th>player</th> <th>country</th> <th>score</th> <th>to par</th> <th>money</th> </tr> </thead> <tbody> <tr> <td>t1</td> <td>ed furgol</td> <td>united states</td> <td>71 + 70 + 71 + 72 = 284</td> <td>+ 4</td> <td>1 6000</td> </tr> <tr> <td>t2</td> <td>gene littler</td> <td>united states</td> <td>70 + 69 + 76 + 70 = 285</td> <td>+ 5</td> <td>3600</td> </tr> <tr> <td>t3</td> <td>lloyd mangrum</td> <td>united states</td> <td>72 + 71 + 72 + 71 = 286</td> <td>+ 6</td> <td>1500</td> </tr> <tr> <td>t3</td> <td>dick mayer</td> <td>united states</td> <td>72 + 71 + 70 + 73 = 286</td> <td>+ 6</td> <td>1500</td> </tr> <tr> <td>t5</td> <td>bobby locke</td> <td>south africa</td> <td>74 + 70 + 74 + 70 = 288</td> <td>+ 8</td> <td>960</td> </tr> <tr> <td>t6</td> <td>tommy bolt</td> <td>united states</td> <td>72 + 72 + 73 + 72 = 289</td> <td>+ 9</td> <td>570</td> </tr> <tr> <td>t6</td> <td>fred haas</td> <td>united states</td> <td>73 + 73 + 71 + 72 = 289</td> <td>+ 9</td> <td>570</td> </tr> <tr> <td>t6</td> <td>ben hogan</td> <td>united states</td> <td>71 + 70 + 76 + 72 = 289</td> <td>+ 9</td> <td>570</td> </tr> <tr> <td>t6</td> <td>shelley mayfield</td> <td>united states</td> <td>73 + 75 + 72 + 69 = 289</td> <td>+ 9</td> <td>570</td> </tr> <tr> <td>t6</td> <td>billy joe patton (a)</td> <td>united states</td> <td>69 + 76 + 71 + 73 = 289</td> <td>+ 9</td> <td>1 0 1</td> </tr> </tbody> </table> <p><b>Verification result: Refuted. Explanation:</b> The cash prize for Tommy Bolt, Fred Haas, and Ben Hogan was \$570 each, totaling \$1710.</p>		place	player	country	score	to par	money	t1	ed furgol	united states	71 + 70 + 71 + 72 = 284	+ 4	1 6000	t2	gene littler	united states	70 + 69 + 76 + 70 = 285	+ 5	3600	t3	lloyd mangrum	united states	72 + 71 + 72 + 71 = 286	+ 6	1500	t3	dick mayer	united states	72 + 71 + 70 + 73 = 286	+ 6	1500	t5	bobby locke	south africa	74 + 70 + 74 + 70 = 288	+ 8	960	t6	tommy bolt	united states	72 + 72 + 73 + 72 = 289	+ 9	570	t6	fred haas	united states	73 + 73 + 71 + 72 = 289	+ 9	570	t6	ben hogan	united states	71 + 70 + 76 + 72 = 289	+ 9	570	t6	shelley mayfield	united states	73 + 75 + 72 + 69 = 289	+ 9	570	t6	billy joe patton (a)	united states	69 + 76 + 71 + 73 = 289	+ 9	1 0 1
place	player	country	score	to par	money																																																														
t1	ed furgol	united states	71 + 70 + 71 + 72 = 284	+ 4	1 6000																																																														
t2	gene littler	united states	70 + 69 + 76 + 70 = 285	+ 5	3600																																																														
t3	lloyd mangrum	united states	72 + 71 + 72 + 71 = 286	+ 6	1500																																																														
t3	dick mayer	united states	72 + 71 + 70 + 73 = 286	+ 6	1500																																																														
t5	bobby locke	south africa	74 + 70 + 74 + 70 = 288	+ 8	960																																																														
t6	tommy bolt	united states	72 + 72 + 73 + 72 = 289	+ 9	570																																																														
t6	fred haas	united states	73 + 73 + 71 + 72 = 289	+ 9	570																																																														
t6	ben hogan	united states	71 + 70 + 76 + 72 = 289	+ 9	570																																																														
t6	shelley mayfield	united states	73 + 75 + 72 + 69 = 289	+ 9	570																																																														
t6	billy joe patton (a)	united states	69 + 76 + 71 + 73 = 289	+ 9	1 0 1																																																														
<p><b>Table E2:</b> 1959 u.s. open (golf)</p> <table border="1"> <thead> <tr> <th>player</th> <th>country</th> <th>year (s)</th> <th>won</th> <th>total</th> <th>to par</th> <th>finish</th> </tr> </thead> <tbody> <tr> <td>ben hogan</td> <td>united states</td> <td>1948, 1950, 1951, 1953</td> <td>287</td> <td>+ 7</td> <td>t8</td> </tr> <tr> <td>cary middlecoff</td> <td>united states</td> <td>1949, 1956</td> <td>294</td> <td>+ 14</td> <td>t19</td> </tr> <tr> <td>liack fleck</td> <td>united states</td> <td>1955</td> <td>294</td> <td>+ 14</td> <td>t19</td> </tr> <tr> <td>liulus boros</td> <td>united states</td> <td>1952</td> <td>297</td> <td>+ 17</td> <td>t28</td> </tr> <tr> <td>tommy bolt</td> <td>united states</td> <td>1958</td> <td>301</td> <td>+ 21</td> <td>t38</td> <td>V2:</td> </tr> </tbody> </table> <p><b>Verification result: Not related.</b></p>		player	country	year (s)	won	total	to par	finish	ben hogan	united states	1948, 1950, 1951, 1953	287	+ 7	t8	cary middlecoff	united states	1949, 1956	294	+ 14	t19	liack fleck	united states	1955	294	+ 14	t19	liulus boros	united states	1952	297	+ 17	t28	tommy bolt	united states	1958	301	+ 21	t38	V2:																												
player	country	year (s)	won	total	to par	finish																																																													
ben hogan	united states	1948, 1950, 1951, 1953	287	+ 7	t8																																																														
cary middlecoff	united states	1949, 1956	294	+ 14	t19																																																														
liack fleck	united states	1955	294	+ 14	t19																																																														
liulus boros	united states	1952	297	+ 17	t28																																																														
tommy bolt	united states	1958	301	+ 21	t38	V2:																																																													

Figure 46: Verifying a textual claim using retrieved tables.

## E.2 Answer Verification

We showcase preliminary experimental results that highlight the initial achievements of **Symphony** in facilitating the verification of generative AI.

**Experiment Setting.** We perform a controlled study to assess textual claims, employing 1,300 textual claims from the TabFact [3] benchmark, which is currently the most advanced benchmark for verifying the credibility of textual hypotheses by utilizing a given table. The data lake consists of 16,573 tables from the TabFact and 2,925 tables sourced from WikiTable-TURL [6].

**Evaluation for Retrieval.** We use Elasticsearch [8] to retrieve the top-5 tables for each textual claim. Given the limited amount of relevant data, we focus on the recall metric for evaluation. Each textual claim is associated with a corresponding table in the original dataset, which we consider relevant evidence, while other retrieved tables are deemed irrelevant. The retrieval performance, measured by R@5, is 0.88.

**Evaluation for Verification.** We evaluate the verification process using two different verifiers: GPT-3.5, the default verifier for both data types, and PASTA [9], a specialized model for text verification. The performance of the verifiers is measured by accuracy. When the retrieved data cannot support or refute a claim, the verifier outputs “not related”. However, in this case, since PASTA that only offers two

different answers: “true” or “false”, we consider it’s also correct when PASTA outputs “false”.

We conduct experiments in two settings. When a relevant table is retrieved and provided as evidence to the verifier, PASTA achieves higher accuracy than GPT-3.5 (0.89 vs. 0.75) in verifying the textual claim based on the table. However, in cases where many of the retrieved tables are irrelevant to the claim, the verifier must accurately determine which tables are not related. In this setting, PASTA’s accuracy drops to 0.72 because it has not encountered this scenario during training, while GPT-3.5 improves to 0.91. Thus, when the retrieved data is highly related to the generative data, local models like PASTA have higher accuracy while protecting privacy. In contrast, GPT-3.5 is better at generalizing and providing explanations for further judgments. Users can select the appropriate model based on their requirements.

In Figure 46, we present a case of verifying a textual claim based on retrieved tables using GPT-3.5. **Symphony** retrieves two tables  $E_1$  and  $E_2$ , where  $E_1$  can be used with an aggregation query to refute the claim while  $E_2$  is not related because it is for the year 1959. The red boxes in Figure 46 show that GPT-3.5 can provide not only a verification result but also some explanation.

## F Open Problems

**Cross-Modal Data Discovery.** Data discovery presents a significant challenge within data preparation, especially when dealing with data lakes that store diverse types of data across various formats, including structured data (e.g., tables), semi-structured data (e.g., graphs), and unstructured data (e.g., images and videos). Unlike data lakes containing only relational tables, discovering relevant data across multiple modalities requires addressing the inherent heterogeneity of these data types. One promising direction for tackling this challenge is to explore cross-modal representation learning, which encodes data from different modalities into a unified vector space. This approach can enable a streamlined data discovery process by supporting embedding-based similarity search. While we have made initial strides in cross-modal representation, our current work has not touched the surface of modeling relationships across different data modalities. Further research is needed to deepen our understanding and improve cross-modal data discovery methods.

**Cross-Modal Data Reasoning and Verification.** One of the complexity of cross-modal reasoning and verification stems from the intricate relationships between different data modalities, such as text, images, and structured data (e.g., tables and knowledge graphs). Each modality often possesses unique characteristics and contextual information that can complicate the verification process. For instance, verifying a claim made in textual data may require correlating it with relevant knowledge graph entities or structured data, where mismatches in representation and interpretation can lead to inaccuracies. Current large language models, such as GPT, demonstrate reasonable performance in reasoning across diverse data types; however, there remains significant room for improvement, particularly regarding privacy and accuracy. To address these challenges, promising directions include the development of domain-specific models that focus on the interactions between specific modalities, improved representation learning techniques for better alignment of data types, and hybrid approaches that combine local and large language models. Additionally, privacy-preserving techniques, such as federated learning and iterative feedback mechanisms, could enhance the robustness and reliability of cross-modal reasoning and verification. These strategies aim to create a more effective framework for ensuring the accuracy and trustworthiness of generative AI outputs across different modalities.

**Trustworthiness of Data Sources.** The accuracy of discovering and verifying data across different modalities in a data lake can be influenced by the quality and reliability of the underlying data sources. Therefore, it is crucial to assess the trustworthiness of different sources accurately to enhance the overall

accuracy and reliability of the entire verification process.

## G Related Works

**Retrieval Augmented Generation Question Answering.** Large language models sometimes generate factually incorrect or misleading information, often due to a lack of real-time knowledge or limited access to external facts beyond their training data. RAG-based Question Answering addresses this by integrating external knowledge retrieval into the generation process. By retrieving relevant document chunks through semantic search, RAG ensures that the model’s responses are grounded in accurate, real-world information, effectively reducing the likelihood of hallucinations. Early approaches focused on jointly training the retriever and generator, ensuring that the retrieved content aligned with the generation model’s intent to provide more accurate answers [10]. With the success of in-context learning, more recent work has treated the retriever as a separate module, directly providing retrieved information to the model via prompts [18]. As retrieval technologies have advanced, RAG-based systems now support multimodal retrieval, enabling answers that draw from diverse data sources [1, 2, 13].

**Trustworthiness of Large Language Models.** The trustworthiness of LLMs is essential for their effective deployment in real-world applications. To assess LLM trustworthiness, researchers have proposed various approaches. For example, TrustLLM [7] provides a comprehensive framework for evaluating LLMs across different trust dimensions. However, evaluating LLM trustworthiness remains challenging, with gaps in holistic assessment approaches. Some studies suggest that self-evaluation, where LLMs assess their confidence in the generated outputs, can help improve selective generation and mitigate inaccuracies [15]. Additionally, understanding the internal mechanisms of LLMs, such as the use of local intrinsic dimension (LID) for predicting truthfulness, has been proposed as a way to measure model reliability [19]. In our work, we aim to improve the trustworthiness of LLMs through post-verification, ensuring that generated outputs are validated against reliable sources after generation to minimize inaccuracies and enhance their overall reliability.

## H Concluding Remarks

In conclusion, **Symphony** represents a significant advancement in the pursuit of trustworthy question answering over multimodal data lakes. By harnessing the power of RAG, **Symphony** effectively addresses the critical challenge of hallucinations inherent in LLMs. Its dual functionality caters to diverse user needs, facilitating both reasoning and verification processes. Through the decomposition of complex queries and the retrieval of relevant information from various data sources, **Symphony** generates grounded answers that can be rigorously cross-checked against reliable datasets. This collaborative approach not only enhances the accuracy of responses but also fosters confidence in the decision-making processes that rely on such information. As we continue to explore the potential of multimodal data and LLMs, **Symphony** stands out as a versatile tool that can adapt to a wide range of applications, paving the way for more reliable and informed use of LLMs in various domains.

## References

- [1] W. Chen, M.-W. Chang, E. Schlinger, W. Y. Wang, and W. W. Cohen. Open question answering over tables and text. In International Conference on Learning Representations, 2021.
- [2] W. Chen, H. Hu, X. Chen, P. Verga, and W. Cohen. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In EMNLP, 2022.

- [3] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Y. Wang. Tabfact: A large-scale dataset for table-based fact verification. arXiv preprint arXiv:1909.02164, 2019.
- [4] Z. Chen, Z. Gu, L. Cao, J. Fan, S. Madden, and N. Tang. Symphony: Towards natural language query answering over multi-modal data lakes. In CIDR, 2023.
- [5] A. de Wynter, X. Wang, A. Sokolov, Q. Gu, and S.-Q. Chen. An evaluation on large language model outputs: Discourse and memorization. Natural Language Processing Journal, 4:100024, 2023.
- [6] X. Deng, H. Sun, A. Lees, Y. Wu, and C. Yu. Turl: Table understanding through representation learning. ACM SIGMOD Record, 51(1):33–40, 2022.
- [7] Y. H. et al. Trustllm: Trustworthiness in large language models, 2024.
- [8] C. Gormley and Z. Tong. Elasticsearch: The Definitive Guide. O’Reilly Media, Inc., 1st edition, 2015.
- [9] Z. Gu, J. Fan, N. Tang, P. Nakov, X. Zhao, and X. Du. Pasta: Table-operations aware fact verification via sentence-table cloze pre-training. In EMNLP, 2022.
- [10] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave. Atlas: Few-shot learning with retrieval augmented language models. Journal of Machine Learning Research, 24(251):1–43, 2023.
- [11] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In EMNLP, pages 6769–6781, Nov. 2020.
- [12] B. Li, Y. Luo, C. Chai, G. Li, and N. Tang. The dawn of natural language to sql: Are we fully ready? Proc. VLDB Endow., 17(11):3318–3331, Aug. 2024.
- [13] H. Luo, Y. Shen, and Y. Deng. Unifying text, tables, and images for multimodal question answering. In EMNLP (Findings), 2023.
- [14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.
- [15] J. Ren, Y. Zhao, T. Vu, P. J. Liu, and B. Lakshminarayanan. Self-evaluation improves selective generation in large language models. In ICBINB, 2023.
- [16] N. Tang, J. Fan, F. Li, J. Tu, X. Du, G. Li, S. Madden, and M. Ouzzani. RPT: relational pre-trained transformer is almost all you need towards democratizing data preparation. Proc. VLDB Endow., 14(8):1254–1261, 2021.
- [17] N. Tang, C. Yang, J. Fan, L. Cao, Y. Luo, and A. Y. Halevy. Verifai: Verified generative AI. In CIDR, 2024.
- [18] X. Wang, Q. Yang, Y. Qiu, J. Liang, Q. He, Z. Gu, Y. Xiao, and W. Wang. Knowledgept: Enhancing large language models with retrieval and storage access on knowledge bases, 2023.
- [19] F. Yin, J. Srinivasa, and K.-W. Chang. Characterizing truthfulness in large language model generations with local intrinsic dimension. ArXiv, abs/2402.18048, 2024.