Natural Language Querying on NoSQL Databases: Opportunities and Challenges [Vision Paper]

Wenlong Zhang
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA
wzhang71@stevens.edu

Tian Shi
BizLidar
Parsippany, NJ, USA
researchtianshi@gmail.com

Ping Wang
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA
pwang44@stevens.edu

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825998

Abstract—**Natural language querying (NLQ) is an important research direction in both natural language processing and database communities. Over the past few years, using modern deep learning language generation and semantic parsing techniques to translate natural language questions to SQL queries, namely Text-to-SQL, has become a promising research topic. Despite the many limitations of using SQL queries for searching due to the predefined data structures and functionality of SQL databases, few attempts have been made beyond SQL query generation. Although there are many well-known, efficient, and scalable NoSQL databases and search engines, such as MongoDB and Elasticsearch, very little work has been devoted to developing NLQ tools for them and exploiting their potential. This gap motivates us to forge and explore the new research direction of NLQ on NoSQL databases. This vision paper aims to investigate the unique characteristics of the NoSQL database in the context of NLQ, examine the integration of NLQ with NoSQL databases, identify emerging research opportunities, and outline key challenges and potential research directions. We hope to inspire and stimulate further research investigation into adopting NoSQL databases for NLQ tasks.**

Index Terms—**Natural language querying, NoSQL database**

I. Introduction

Natural language querying (NLQ) is a crucial natural language processing (NLP) task that can significantly improve the accessibility of specialized databases to a broader audience [1]–[5]. This task is also an important preliminary step in many advanced research fields, such as Retrieval-Augmented Generation [6] and Large Language Models (LLMs) [7]. Previous research has focused on one of the NLQ branches called Text-to-SQL, concentrating on the unique challenges related to Structured Query Language (SQL) query generation on relational databases [8]–[11]. Text-to-SQL aims to automate the process where individuals utilize natural language for querying and searching within relational databases. Specifically, this involves the translation of naturally phrased questions into executable SQL queries. Such a paradigm significantly enhances the accessibility of various specially tailored databases, such as Wikipedia [12], Electronic Health Records (EHR) [13] and Spider [8], [14] to a broader audience. For example, it is notably beneficial for professionals like doctors and scientists who lack expertise in database management or data structures by enabling them to access comprehensive data and address various problems efficiently. However, there are some limitations, with the functional constraints of SQL being a significant factor leading to unexpected results in various areas, such as full-text search and handling diverse information types. For instance, in healthcare domain, EHR data usually includes clinical notes and key components containing various types of patient information, such as discharge summaries and family history. Due to SQL's limited full-text search capabilities, it is challenging to search within these clinical notes using SQL queries effectively. Furthermore, SQL is inherently designed for static datasets, making it less effective at integrating dynamic external knowledge, which is a capability that is particularly valuable for users in fields like healthcare.

This problem has been effectively addressed in the industry through the use of non-relational databases, such as Elasticsearch and MongoDB, which excel at handling complex and dynamic data formats, thereby facilitating the extraction of more valuable information. Non-relational databases are widely employed in scenarios requiring rapid response and complex information extraction [15]. Particularly with the rapid growth of the recent LLMs, building advanced NLQ models from complex data is increasingly reliant on NoSQL databases, which offer efficient solutions for quicker information extraction. In our previous work, we introduced and formulated the Text-to-ESQ task for NLQ on the Elasticsearch database, making the first exploration of performing NLQ tasks on NoSQL databases [16], [17]. Our findings demonstrate both the effectiveness and efficiency of this new strategy while highlighting the significant potential to be further explored.

There are three primary challenges in this field. First, unlike the standardized SQL syntax, NoSQL databases come in many different types, each with its own advanced features. Selecting the most suitable NoSQL database for a specific task and designing the appropriate architecture are challenges still being explored. Second, the reliance on NoSQL queries for data extraction poses a significant barrier for researchers who are more familiar with the widely-used SQL syntax. The diversity of NoSQL queries can create additional challenges in real-world applications. For example, Elasticsearch queries tend to be much longer than SQL queries, which can lead to issues like long-tail generation problems. Third, the lack of training data and public datasets limits the development of tailored

979-8-3503-6248-0/24/$31.00 ©2024 IEEE

solutions that can effectively integrate NoSQL into various research methodologies.

This vision paper discusses the emerging opportunities of NoSQL databases for complex information, highlights the unique challenges, and identifies new research directions in this highly interdisciplinary area.

II. NoSQL Databases

In this section, we introduce several key NoSQL databases along with their advantages. This overview aims to provide valuable insights for those exploring this field.

A. Overview of NoSQL Databases

The emergence of NoSQL databases has significantly influenced various industry projects, frequently surpassing traditional SQL databases in applications. This shift is largely attributed to the advanced capabilities inherent in NoSQL technologies. However, within the academic domain, there remains a notable lack of research leveraging this evolving field for NLQ tasks. In this section, we provide a comprehensive overview of five prominent NoSQL databases and highlights their unique advantages.

1) Elasticsearch: Developed by Elasticsearch B.V., it is a robust search and analytics engine built on the Apache Lucene library [18]. It is extensively utilized for real-time distributed search and data analytics, providing high availability and scalability. With its schema-free JSON documents and advanced search capabilities, Elasticsearch is particularly well-suited for full-text search, log and event data analysis, and operational intelligence applications.

2) MongoDB: Created by MongoDB Inc., is a document-oriented database known for its flexibility and ease of use [19]. It stores data in JSON-like BSON documents, allowing for dynamic schemas. MongoDB excels in handling large volumes of unstructured data, making it suitable for applications requiring real-time analytics, content management, and Internet of Things (IoT) solutions. Its horizontal scaling and high availability features ensure robust performance and reliability.

3) Cassandra: Is an open-source project managed by the Apache Software Foundation and a highly scalable and distributed NoSQL database designed to handle large amounts of data across many commodity servers [20]. Its peer-to-peer architecture and support for multi-data center replication provide exceptional fault tolerance and high availability. Cassandra is particularly favored for handling time-series data, real-time analytics, and applications demanding high write throughput.

4) Couchbase: Combines the strengths of both document and key-value stores developed by Couchbase Inc [21]. It offers a flexible JSON document model with SQL-like querying, built-in caching for sub-millisecond data operations, and full-text search capabilities. Couchbase's distributed architecture ensures high performance, availability, and scalability, making it ideal for interactive fields that require real-time data access.

5) Redis: An in-memory key-value store renowned for its lightning-fast maintained by Redis Labs [22]. It supports a wide array of data structures, such as strings, hashes, lists, sets, and sorted sets. Redis's in-memory nature enables sub-millisecond response times, making it perfect for caching, session management, real-time analytics, and message brokering. Its simplicity, speed, and versatility have made Redis a critical component in many high-performance applications.

In conclusion, all these listed NoSQL databases represent the forefront of modern data management solutions, each offering unique features and advantages tailored to meet the diverse demands of today's data-driven tasks.

B. Differences of NoSQL Databases from SQL Databases

In the domain of database management, SQL and NoSQL databases embody two distinct paradigms for data storage and retrieval, each tailored to different application needs and workloads. A clear understanding of the key differences between SQL and NoSQL databases is essential for selecting the appropriate database technologies for a given task. This section details the primary distinctions between SQL and NoSQL, emphasizing key aspects of databases such as data models, scalability, schema flexibility, and consistency [23].

1) Data Models: SQL databases are based on a relational model, where data is organized into tables with rows and columns. Each table has a predefined schema that defines the structure and type of data that can be stored. In contrast, NoSQL databases employ a variety of data models, including document, key-value, column-family, and graph models. These models offer greater flexibility, allowing for the storage of unstructured or semi-structured data. Document databases, for instance, store data in JSON-like formats, which can vary from document to document within the same database.

2) Schema Flexibility: SQL databases enforce a rigid schema, where the structure of the data must be defined before data entry. Any changes to the schema require migration processes, which can be complex and time-consuming. NoSQL databases offer more flexible schema capabilities, allowing for the storage of data without a predefined structure. This flexibility makes it easier to accommodate evolving data requirements and supports agile development practices.

3) Scalability: SQL databases scale vertically, meaning they require more powerful hardware to handle increased data loads [24]. This vertical scaling can become costly and may eventually reach hardware limitations. NoSQL databases, on the other hand, are designed to scale horizontally. They can distribute data across multiple servers, allowing them to handle large volumes of data traffic by adding more nodes to the cluster, which provides greater scalability and fault tolerance.

4) Consistency: SQL databases prioritize Atomicity, Consistency, Isolation, and Durability (ACID). properties to ensure reliable transactions and consistency of data [25]. This strict consistency model is essential for applications requiring precise and accurate data handling. NoSQL databases often follow the theorem Consistency, Availability, Partition Tolerance (CAP) and may sacrifice strict consistency for availability and partition tolerance. Many NoSQL systems implement eventual consistency, where updates propagate to all nodes over time, making them suitable for applications where absolute real-time consistency is not critical.

In real-world applications, many fields require more flexible databases. For example, Electronic Health Records (EHR) data in healthcare often cover unstructured data, such as clinical notes and discharge summaries, and don't fit neatly into predefined formats [26]. Some areas, like COVID-19 vaccine data, require rapid response and dynamic management due to the large volume of data being updated daily [27]. Other fields, such as legal documents, need different structures for storage [28]. The growing complexity and volume of data have outpaced the capabilities of traditional SQL databases [29]. These challenges also extend to data access and retrieval, creating opportunities to explore the potential of NoSQL databases and adopt them for more effective solutions in handling complex problems, such as NLQ for automatic information retrieval.

III. New Opportunities for Natural Language Querying on NoSQL Databases

A. Why NLQ for NoSQL is Different?

Three key aspects make NLQ on NoSQL databases transformative compared to NLQ on SQL databases for information retrieval.

First, the flexibility of NoSQL databases allows NLQ to handle diverse types of information within a single database, such as key-value pairs, document stores, column-family stores, and graph data formats [30]. This capability supports the management of semi-structured or unstructured data, enabling NLQ to address more complex real-world problems. For example, in healthcare, where tasks frequently involve the integration of numerical data with descriptive information, NoSQL databases streamline the process by eliminating the complexity found in existing approaches that require handling different data types separately and subsequently linking the results [31]. This greatly simplifies the architecture for tasks involving multiple data formats.

Second, using NoSQL query languages tailored for JSON documents facilitates seamless integration with other advanced data processing frameworks, such as Hadoop [32], [33]. These approaches significantly enhance the speed of processing large datasets, break through traditional limits, and allow the NLQ to deal with problems that need huge computing resources. This is a critical requirement for the preliminary step in many advanced research fields like preparing training data for LLMs [34]. Additionally, it plays a crucial role in real-life events such as COVID-19 prevention and control, due to the large number of patients and the complexity of the information that needs to be processed [35].

Third, the dynamic nature of NoSQL databases offers significant advantages for emerging tasks that involve evolving data, such as medical histories or online learning in LLMs. Unlike the rigid table structures of SQL, NoSQL databases can easily accommodate changes, enabling support for real-time online learning. This adaptability allows NLQ task models to continuously learn and update with new data in real time, a critical requirement for these applications.

B. Transformative Applications

With the above advantages, this emerging field is poised to drive innovation across various domains due to its crucial role in data preprocessing. Below are key areas expected to be most significantly impacted by this new strategy.

1) Information Retrieval: Many previous information retrieval approaches have primarily focused on SQL databases, retrieving standard information from tables by searching through indexes or column names [36]. The strategy of natural language querying on NoSQL databases presents an innovative solution to handle unstructured or semi-structured data, providing a user-friendly solution for navigating and extracting insights. Unlike traditional SQL databases that apply a uniform schema, NoSQL databases offer flexible, schema-less structures that make the extraction processing more efficient, it also enables NoSQL databases to handle diverse data types, facilitating their use in more complex and realistic applications.

2) NLQ for Multi-modal Data: NLQ for semantic searching on multi-modal data plays a critical role in retrieval systems [37]. Beyond traditional text and structured data, modern applications generate vast amounts of unstructured content, such as images and videos. By incorporating image and video captions as textual metadata into NoSQL databases, NLQ can be extended to facilitate rich semantic searches across multiple modalities. This approach enables users to retrieve visual content based on specific contexts described in natural language queries. For instance, a user could search for videos of "sunsets over mountains" or images of "people at a concert" using NLQ, leveraging the contextual captions stored within the database. NoSQL's flexibility in handling diverse data formats, combined with the NLQ, allows for a more intuitive and accurate search experience, bridging the gap between text-based queries and non-textual content, thus offering a more dynamic way to interact with multi-modal datasets.

3) Large Language Models Finetuning: NLQ plays a crucial role in fine-tuning LLMs because it allows people to interact with the data in a more intuitive manner without requiring complex programming or processing. NLQ on NoSQL databases benefits fine-tuning by providing flexibility in handling diverse unstructured data, enabling more efficient training on large-scale datasets. Adopting NoSQL databases for NLQ to support the fine-tuning of LLMs requires a completely new architectural design [38] and benefits the whole processing by two crucial points. Unlike traditional SQL databases, the new strategy allows seamless integration with other big data processing techniques, significantly improving the accuracy of data extraction and the overall efficiency of the fine-tuning pipeline. Furthermore, NoSQL databases offer a variety of data formats that can be utilized directly, eliminating the need for extensive preprocessing of data processing.

IV. Unique Challenges

Despite the numerous advantages of applying NLQ on NoSQL databases for various tasks, some unique challenges still need to be addressed. These challenges are distinct from those encountered with traditional methods for SQL databases, requiring novel approaches to address them effectively.

First, selecting appropriate NoSQL databases is challenging since unlike relational databases that follow a standardized query language, NoSQL databases come in diverse structures (e.g., key-value stores, document stores, column-family stores, or graph databases) and use different query languages or APIs. Each type has its own strengths and weaknesses, which require careful evaluation during selection for intended tasks. It is essential to select a NoSQL database that aligns well with the needs of the specific NLQ task, particularly in terms of data structure and scalability. Overcoming this challenge will ensure the selected database can effectively handle natural language processing and query translation tasks for NLQ.

Second, handling complex data structures in NoSQL databases presents a significant challenge for NLQ. NoSQL databases often handle unstructured or semi-structured data, which can have complex nesting, hierarchical relationships, or varying schemas. Translating natural language into queries that can navigate these complex structures requires advanced capabilities. The NLQ system must be able to accurately interpret user intent and map it to the appropriate data model, which can be both intricate and non-trivial due to the diversity and complexity of the data formats. Therefore, it requires robust and advanced algorithms to ensure that the queries are generated accurately and that the data is finally correctly retrieved from these complex structures in NoSQL databases.

Third, natural language queries are inherently ambiguous, as users often use different keywords, synonyms, or fuzzy terms to express their needs, which complicates the task of translating these queries into accurate queries in NoSQL databases. Unlike SQL databases, which operate with structured queries, NLQ on NoSQL databases must accommodate this variability and support advanced search features such as keyword expansion, synonym handling, and fuzzy matching. The flexibility of NoSQL databases allows multiple ways to achieve similar query outcomes, resulting in numerous similar results. This introduces complexity into the final analysis and increases the risk of errors. Implementing these features requires sophisticated NLP algorithms and seamless integration with NoSQL database search capabilities, which may not be inherently supported.

V. Future Research Directions

Given that NLQ on NoSQL databases is a novel paradigm of NLQ, it holds substantial potential for future research and applications. Below, we highlight several areas where it could advance current methodologies and drive new innovations.

A. Training Task-Specific NLQ Models on NoSQL Databases

Many existing models are developed and trained specifically for NLQ tasks on SQL databases, but they often perform poorly on NLQ tasks on NoSQL databases due to the latter's complex and varied data formats. However, performance can be significantly improved by adapting and training these models on the tailored specific tasks of NoSQL data and queries for querying hierarchical structured data or extracting information from document-based databases. Therefore, a potential direction is to develop and train the task-specific models for NLQ on the NoSQL databases to enhance task performance. Traditional NLQ models on SQL databases struggle with handling complex data structures and processing tasks. This new NLQ paradigm offers potential opportunities for improved accuracy and efficiency in diverse NoSQL environments.

B. Leveraging LLMs for NLQ on NoSQL Databases

LLMs possess the ability to understand complex natural language queries in NLQ tasks, allowing users to interact with databases without requiring proficiency in specific database queries or syntax. Moreover, LLMs can understand the intent behind the query, identify key entities, relationships, and conditions from the natural languages, and translate natural language inputs into NoSQL queries. Therefore, leveraging LLMs for NLQ on NoSQL databases offers a promising research direction, with the potential to improve user accessibility and efficiency in interacting with NoSQL databases. Key areas of investigation include evaluating the capability of LLMs to manage complex queries involving nested conditions and multiple entities, as well as examining how NoSQL schema design impacts the performance of NLQ tasks. Understanding these aspects will be crucial in optimizing LLMs-driven NLQ on NoSQL databases.

C. Knowledge-Guided NLQ Task

Incorporating domain-specific knowledge into NLQ tasks is crucial in improving query accuracy and relevance. One potential direction is to build knowledge-guided NLQ systems that leverage external knowledge bases to inform the query interpretation process, enabling the system to better understand and disambiguate complex queries [39]. This approach will be particularly valuable in domains with rich and specialized vocabularies. Traditional NLQ systems might struggle to deal with different formats of user intent and reorganize them. NLQ on NoSQL databases with specific knowledge can directly extract information from multiple data types and form the knowledge base in different formats like tree diagrams or graph databases to support the query generation.

D. NLQ Enhanced Retrieval Augmented Generation (RAG)

Typically, LLMs generate responses for the user input based on the information they were trained on. RAG enhances this process by incorporating an external knowledge base to improve the quality of the generated responses. To achieve this, it is essential to use high-quality external data, which may come from diverse data sources and cover multiple formats. A major challenge is how to perform a relevancy search to extract pertinent information. Unlike traditional rule-based approaches, which often face difficulties with diverse and unstructured data, NLQ on NoSQL databases offers an effective way to address this challenge by translating natural language queries to database queries for retrieving high-quality, precise data. This new strategy in data retrieval not only enhances query accuracy but also lays a solid foundation for further steps in RAG, such as enriching LLMs' prompts with additional context. By utilizing the accurate and relevant data retrieved from NoSQL databases, subsequent generation processes in LLMs can benefit from more reliable input, leading to enhanced overall performance in applications that require dynamic and real-time data processing.

VI. Acknowledgments

This work was supported in part by the US National Science Foundation grant IIS-2245907 and the startup funding from the Stevens Institute of Technology.

References

[1] V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," *arXiv preprint arXiv:1709.00103*, 2017.

[2] H. Kim, B.-H. So, W.-S. Han, and H. Lee, "Natural language to sql: where are we today?" *Proceedings of the VLDB Endowment*, vol. 13, no. 10, pp. 1737–1750, 2020.

[3] F. Li and H. V. Jagadish, "Constructing an interactive natural language interface for relational databases," *Proceedings of the VLDB Endowment*, vol. 8, no. 1, pp. 73–84, 2014.

[4] D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan, "Athena: an ontology-driven system for natural language querying over relational data stores," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1209–1220, 2016.

[5] J. Sen, F. Ozcan, A. Quamar, G. Stager, A. Mittal, M. Jammi, C. Lei, D. Saha, and K. Sankaranarayanan, "Natural language querying of complex business intelligence queries," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 1997–2000.

[6] P. Hu, C. Lu, F. Wang, and Y. Ning, "Dualmar: Medical-augmented representation from dual-expertise perspectives," *arXiv e-prints*, pp. arXiv–2410, 2024.

[7] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[8] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman *et al.*, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," *arXiv preprint arXiv:1809.08887*, 2018.

[9] T. Scholak, N. Schucher, and D. Bahdanau, "Picard: Parsing incrementally for constrained auto-regressive decoding from language models," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 9895–9901.

[10] R. Zhang, T. Yu, H. Er, S. Shim, E. Xue, X. V. Lin, T. Shi, C. Xiong, R. Socher, and D. Radev, "Editing-based sql query generation for cross-domain context-dependent questions," in *Proceedings of the 2019 Conference on EMNLP*, 2019, pp. 5338–5349.

[11] X. V. Lin, R. Socher, and C. Xiong, "Bridging textual and tabular data for cross-domain text-to-sql semantic parsing," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4870–4888.

[12] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 509–518.

[13] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[14] P. Wang, T. Shi, and C. K. Reddy, "Text-to-sql generation for question answering on electronic medical records," in *Proceedings of The Web Conference 2020*, 2020, pp. 350–361.

[15] J. Han, H. E, G. Le, and J. Du, "Survey on nosql database," in *2011 6th International Conference on Pervasive Computing and Applications*, 2011, pp. 363–366.

[16] W. Zhang, K. Zeng, X. Yang, T. Shi, and P. Wang, "Text-to-esq: A two-stage controllable approach for efficient retrieval of vaccine adverse events from nosql database," in *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2023, pp. 1–10.

[17] P. Sood, X. Yang, and P. Wang, "Natural language querying on domain-specific nosql database with large language models," in *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2024.

[18] Elasticsearch, https://www.elastic.co/.

[19] MongoDB, https://www.mongodb.com/.

[20] Cassandra, https://cassandra.apache.org/ /index.html.

[21] Couchbase, https://www.couchbase.com/.

[22] Redis, https://redis.io/.

[23] Y. Li and S. Manoharan, "A performance comparison of sql and nosql databases," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2013, pp. 15–19.

[24] D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden *et al.*, "The design and implementation of modern column-oriented database systems," *Foundations and Trends® in Databases*, vol. 5, no. 3, pp. 197–280, 2013.

[25] D. G. Chandra, "Base analysis of nosql database," *Future Generation Computer Systems*, vol. 52, pp. 13–21, 2015.

[26] W. Zhang, B. Ingale, H. Shabir, T. Li, T. Shi, and P. Wang, "Event detection explorer: An interactive tool for event detection exploration," in *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, ser. IUI '23 Companion. New York, NY, USA: Association for Computing Machinery, 2023, p. 171–174. [Online]. Available: https://doi.org/10.1145/3581754.3584178

[27] E. Mathieu, H. Ritchie, E. Ortiz-Ospina, M. Roser, J. Hasell, C. Appel, C. Giattino, and L. Rodés-Guirao, "A global database of covid-19 vaccinations," *Nature human behaviour*, vol. 5, no. 7, pp. 947–953, 2021.

[28] J. Piskorski and R. Yangarber, "Information extraction: Past, present and future," *Multi-source, multilingual information extraction and summarization*, pp. 23–49, 2013.

[29] P. Wang, T. Shi, and C. K. Reddy, "Text-to-sql generation for question answering on electronic medical records," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 350–361. [Online]. Available: https://doi.org/10.1145/3366423.3380120

[30] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: Nosql and newsql data stores," *Journal of Cloud Computing: advances, systems and applications*, vol. 2, pp. 1–24, 2013.

[31] A. Quamar, V. Efthymiou, C. Lei, F. Özcan *et al.*, "Natural language interfaces to data," *Foundations and Trends® in Databases*, vol. 11, no. 4, pp. 319–414, 2022.

[32] hadoop, https://hadoop.apache.org/.

[33] S. Sagiroglu and D. Sinanc, "Big data: A review," in *2013 international conference on collaboration technologies and systems (CTS)*. IEEE, 2013, pp. 42–47.

[34] E. Nijkamp, H. Hayashi, C. Xiong, S. Savarese, and Y. Zhou, "Codegen2: Lessons for training llms on programming and natural languages," *arXiv preprint arXiv:2305.02309*, 2023.

[35] J. Antas, R. Rocha Silva, and J. Bernardino, "Assessment of sql and nosql systems to store and mine covid-19 data," *Computers*, vol. 11, no. 2, p. 29, 2022.

[36] M. Kobayashi and K. Takeda, "Information retrieval on the web," *ACM computing surveys (CSUR)*, vol. 32, no. 2, pp. 144–173, 2000.

[37] H. Zhao, H. Liu, and Y. Fu, "Incomplete multi-modal visual data grouping." in *IJCAI*, 2016, pp. 2392–2398.

[38] X. Lin, W. Wang, Y. Li, S. Yang, F. Feng, Y. Wei, and T.-S. Chua, "Data-efficient fine-tuning for llm-based recommendation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 365–374.

[39] A. Karpatne, R. Kannan, and V. Kumar, *Knowledge guided machine learning: Accelerating discovery using scientific knowledge and data*. CRC Press, 2022.