Question Answering Technology Based on Text to SQL

Chetan Arekar[1], Vivek Jadhav[2] , Rushikesh Gaikwad[3] , Ajay Ghumre[4]
D. Y. Patil Institute Of Engineering & Technology, Pune[1-4]

Abstract:

Question answering (QA) systems have gained significant attention in recent years due to their ability to retrieve relevant information and provide precise answers to user queries. The advent of text-to-SQL projects has further enhanced the capabilities of QA systems by enabling them to understand and execute SQL queries on structured databases. This research paper aims to explore the development and advancements of question answering technology based on text-to-SQL projects. We discuss the key components, challenges, and methodologies involved in building effective QA systems with text-to-SQL capabilities. Additionally, we evaluate the performance and limitations of existing approaches and propose potential directions for future research.

Keywords: Question answering system, Text-to-SQL, Natural language processing, Structured query language

I. INTRODUCTION

A. Background and significance of Question Answering (QA) technology:

In recent years, there has been a growing interest in developing intelligent systems that can understand and respond to natural language questions, known as Question Answering (QA) technology. QA systems aim to bridge the gap between human language and machine understanding, enabling users to interact with computers more naturally and efficiently.

The significance of QA technology lies in its ability to facilitate information retrieval and knowledge extraction. Traditional search engines often provide a list of documents or web pages that may contain relevant information, requiring users to manually extract the answers. In contrast, QA systems aim to directly provide precise and concise answers to user queries, saving time and effort.

QA technology has practical applications across various domains. In customer support, QA systems can provide instant responses to frequently asked questions, improving customer satisfaction and reducing the workload of support agents. In the education sector, QA systems can assist students in finding relevant information for assignments and studying. In the research field, QA systems can help researchers quickly retrieve information from large volumes of scientific literature.

B. Overview of Text to SQL and its role in QA:

Text to SQL is a subfield of QA technology that focuses on transforming natural language questions into structured SQL queries, which can be executed on databases to retrieve specific information. The goal of Text to SQL is to enable users to interact with databases using natural language queries instead of requiring them to have knowledge of SQL syntax.

Text to SQL plays a crucial role in QA systems, as it enables users to retrieve information from databases by formulating queries in a more intuitive and human-like manner. Instead of manually constructing complex SQL queries, users can express their information needs in natural language, and the Text to SQL component of a QA system will convert these queries into executable.

II. OBJECTIVE

By addressing these objectives, this research paper aims to contribute to the existing body of knowledge on question answering technology, specifically focusing on the integration of text-to-SQL projects. The findings and insights presented in this paper can serve as a valuable resource for researchers, practitioners, and developers working in the field of natural language processing, information retrieval, and database management.

III. METHODOLOGY

The methodology of question answering technology based on text to SQL involves several steps. It starts with data preparation, where a structured database is acquired and the data is preprocessed for efficient querying. Training data is collected, consisting of natural language questions and their corresponding SQL queries. NLU techniques are then applied to process the user's question and understand its structure. Query generation algorithms or models are used to convert the processed question into SQL queries. Schema mapping establishes connections between the natural language representation and the structured database schema. The generated SQL queries are executed against the database, and the relevant data is retrieved. Post-processing techniques refine the data, and a response is generated to provide a clear answer to the user. The system's performance is evaluated, and improvements are made based on the results. Throughout the process, machine learning and NLP techniques are employed to enhance accuracy.

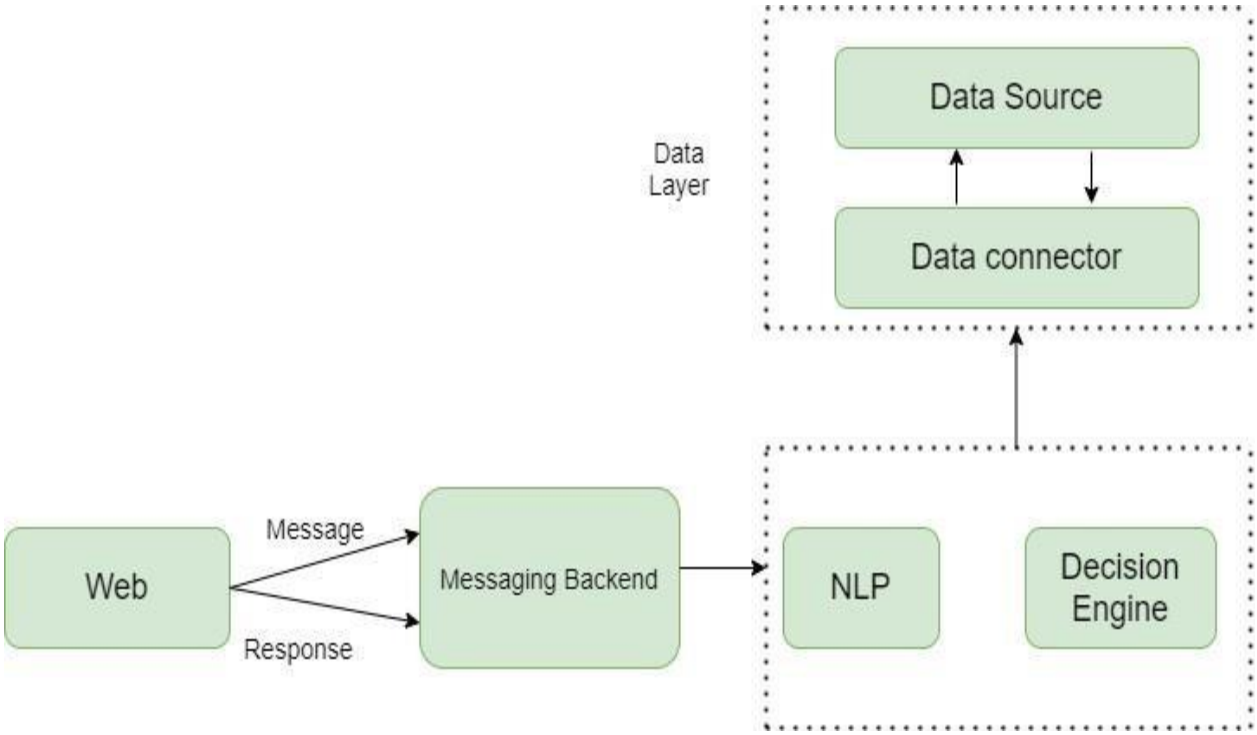

Fig. 1: System Architecture

i. **Web Component:** The web component serves as the user interface, typically in the form of a web application. Users interact with the system by inputting their natural language questions through the web interface.

ii. **Backend Messaging:** The backend messaging component acts as an intermediary between the web component and the other system components. It receives user queries from the web component and forwards them to subsequent components for processing.

iii. **Natural Language Processing (NLP):** The NLP component processes the user queries received from the backend messaging component. It employs various techniques such as tokenization, part-of-speech tagging, syntactic parsing, named entity recognition, and semantic role labeling. These NLP techniques help extract the meaning and structure of the questions.

iv. **Decision Engine:** The decision engine component receives the processed user queries from the NLP component. It applies rule-based or machine learning algorithms to understand the intent of the queries and make decisions based on predefined criteria. The decision engine determines the appropriate actions to be taken based on the analyzed queries.

v. **Data Connector:** The data connector component handles the interaction between the system and the data source. It establishes a connection with the structured database that contains the relevant information to be queried. The data connector receives the processed user queries and translates them into SQL queries that can be executed against the data source.

vi. **Data Source:** The data source component represents the structured database where the relevant information resides. It can be a relational database management system (RDBMS) or any other structured data storage system. The data source stores the necessary data and responds to SQL queries by providing the requested information to the system.

IV. SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how— and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

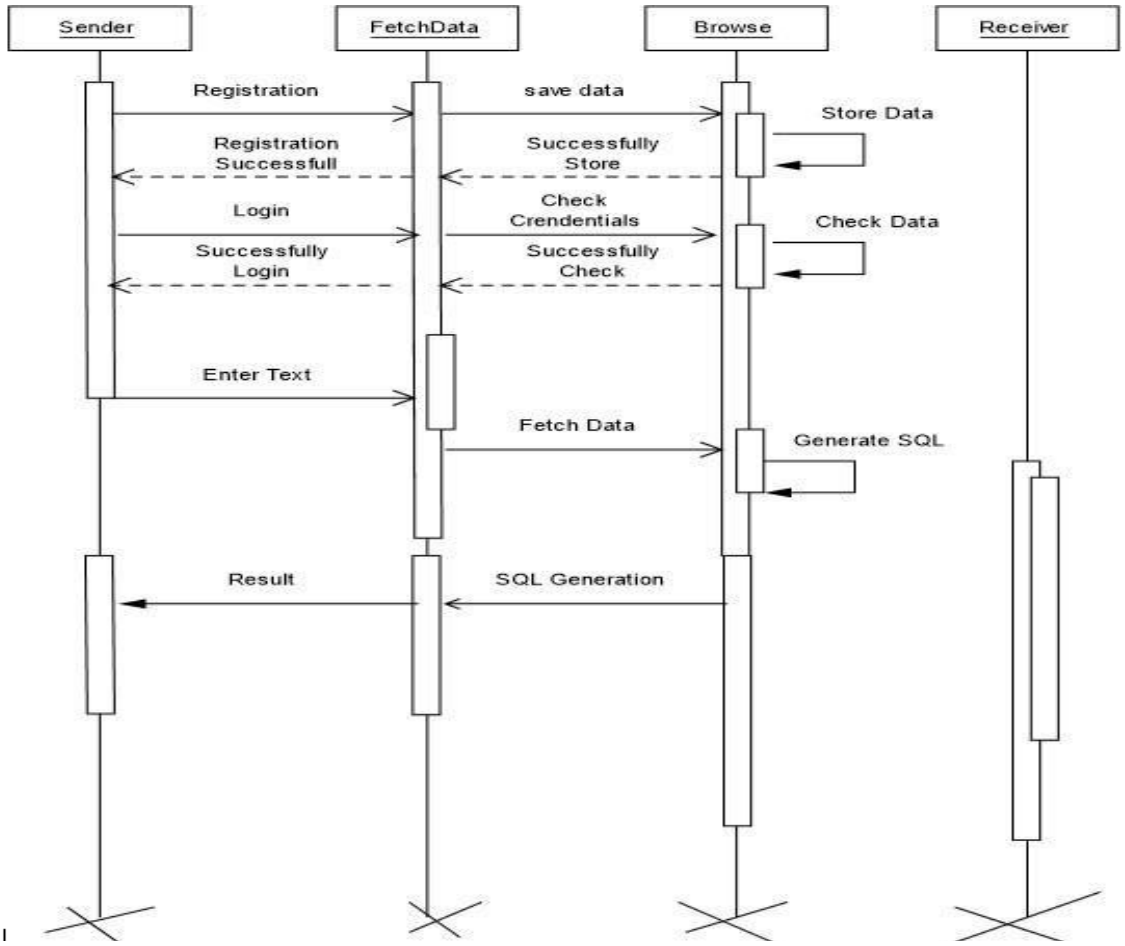

Fig. 2: Sequence Diagram

v. ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- Rounded rectangles represent actions;
- Diamonds represent decisions;
- Bars represent the start (split) or end (join) of concurrent activities; • A black circle represents the start (initial state) of the workflow; • An encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen. Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for ex-pressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

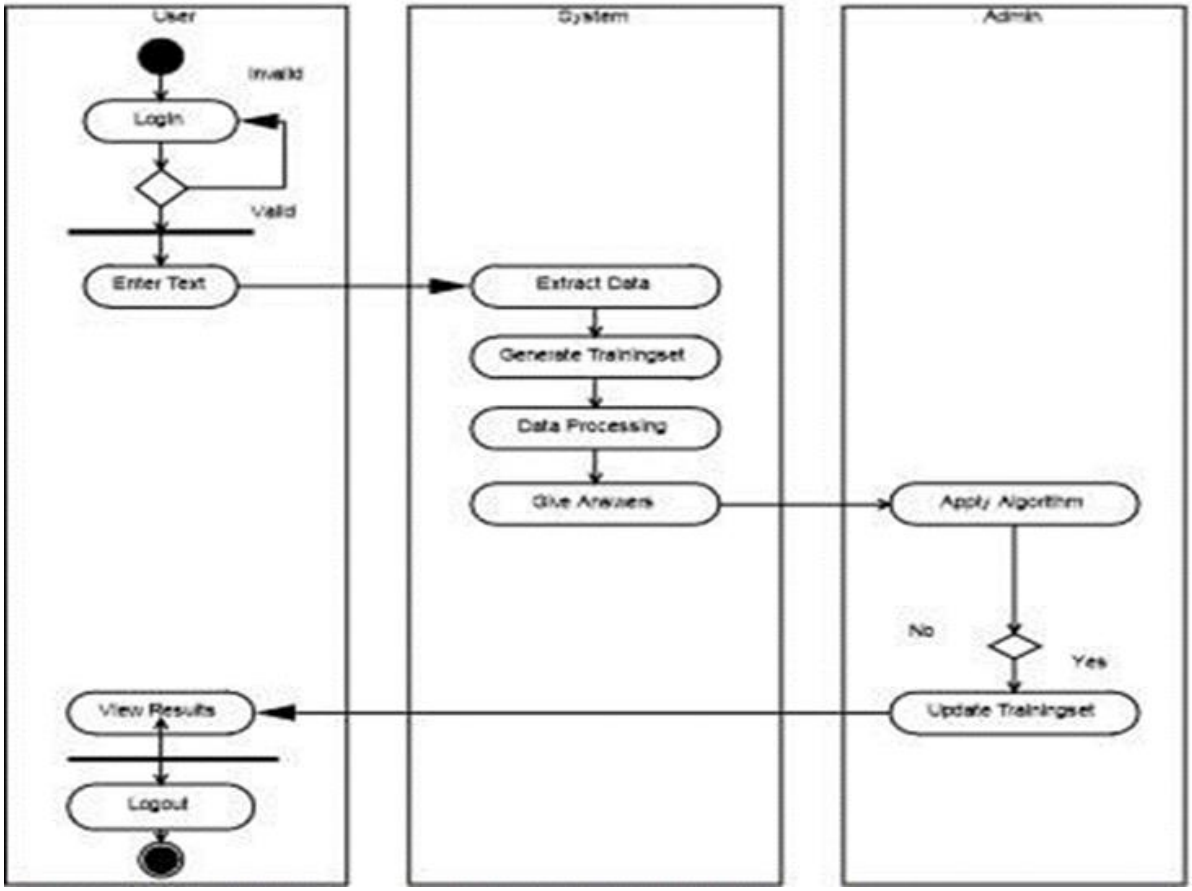

Fig. 3: Activity Diagram

VI. COMPONENT DIAGRAM

A Component Diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems thathave many components. Components communicate with each other using interfaces. The interfaces are linked using connectors.

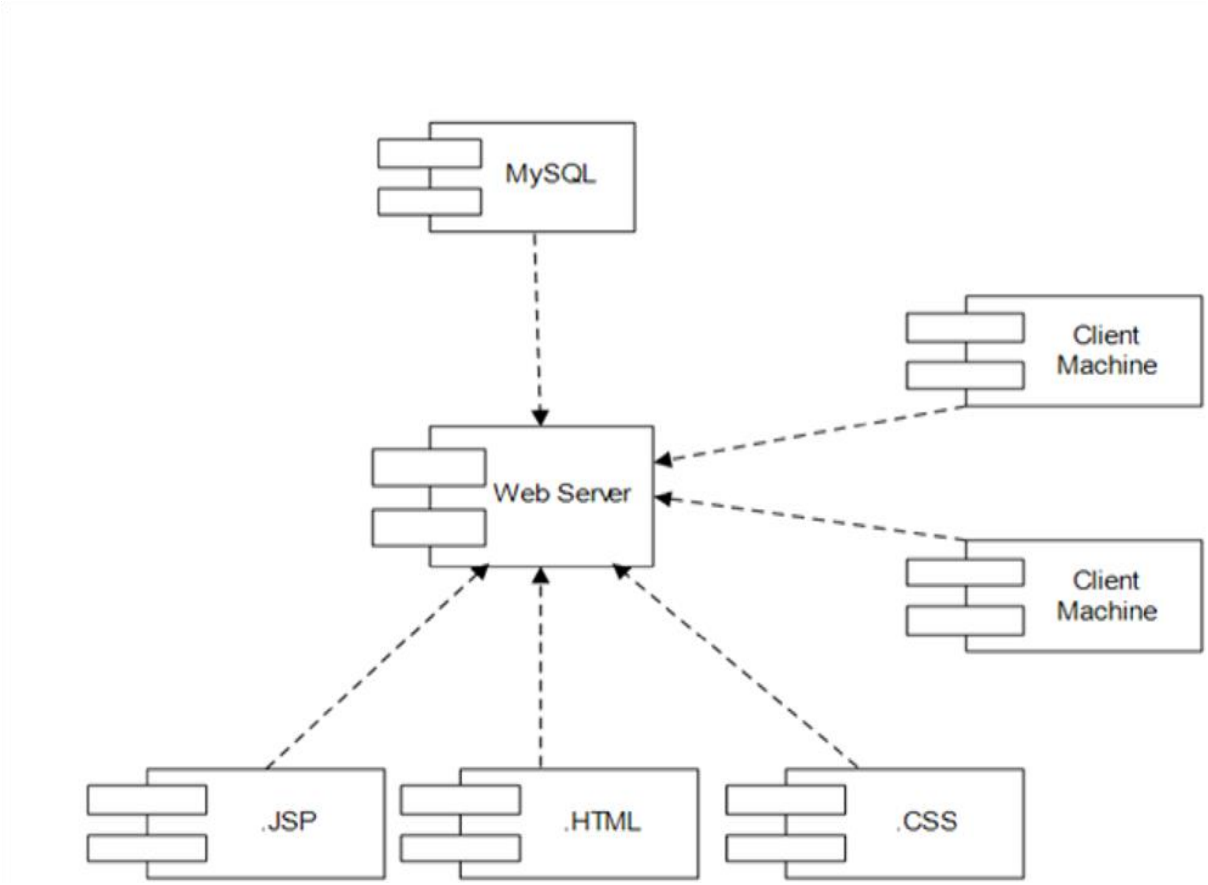

Fig. 4: Components Diagram

In a question answering technology based on text to SQL, the key components involved are MySQL (as the data source), client machine (user's device), CSS (for styling), HTML (for web page structure), and JSP (for dynamic content generation). MySQL serves as the database to retrieve relevant information. The client machine accesses the system through a web browser. CSS is used to define the visual appearance, HTML structures the web page, and JSP enables dynamic content and server-side logic. The system processes user queries, generates SQL queries, executes them on the MySQL database, retrieves data, and presents responses through the web interface. Additional components like NLP modules may enhance system functionality.

VII. CONCLUSION

The work of converting text to SQL is currently being employed more and more inthe realm of practical application, which has caught the interest of academia. The model's decoding techniques, which include intermediate representation, treedecoding, graph network modelling, etc., have produced some results, but the model's impact on complicated data is not immediately apparent. At the same time,we should take into account the standardisation of database tables, the utilisationof outside information, and the incorporation of progressive discussion in practicalimplementation.

VIII. ACKNOWLEDGEMENT

We're really indebted to D.Y Patil Institute of Engineering and Technology, Ambi, Pune for providing us an opportunity to undertake this project work as partial fulfilment of the BACHELOR's Degree in BACHELOR OF ENGINEERING curriculum. We would like to express our heartiest gratitude to Prof. Vishal Walunj and all the faculties of the BE Computer Department for their encouraging support and guidance in carrying out this project. We express our sincere thanks to D.Y Patil Institute of Engineering Technology, Ambi, Pune for permitting us to take this project work and for them instance of the good programming technique, which helped us to design and develop a successful **QUESTION ANSWERING TECHNOLOGY BASED ON TEXT TO SQL**. Finally, sincere thanks to our project members, mentors and all well-wishers for their esteemed guidance, support, valuable suggestions and constructive criticism.

IX. REFERENCES

1. Qing Li , Member, IEEE, Lili Li , Qi Li, and Jiang Zhong "A Comprehensive Exploration on Spider with Fuzzy Decision Text-to-SQL Model" Proces. 2020

2. Run-Ze Wang , Zhen-Hua Ling , Senior Member, IEEE, Jing-Bo Zhou, andYu Hu"A Multiple-Integration Encoder for Multi-Turn Text-to-SQL Semantic Parsing", VOL. 29, 2021

3. Sulochana Deshmukh, Marwan Bikdash "Automatic Text-to-SQL MachineTranslation for Scholarly Publication Database Search", Greensboro,NC. 27411

4. Huajie Wang , Lei Chen†, Mei Li , Mengnan Chen "GuideSQL: Utilizing Tables to Guide the Prediction of Columns for Text-to-SQL Generation"