Journal of Artificial Intelligence and Capsule Networks (ISSN: 2582-2012)
www.irojournals.com/aicn/

Natural Language Driven Leave Management Using Lang Chain Agents and Few-Shot Learning

Vijayalakshmi S.[1], Ashwin Kumar R S.[2], Rakshana V.[3], Srivatsan R.[4]

Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, India.

E-mail: [1]svl.cse@psgtech.ac.in, [2]ashwinkumarrs13@gmail.com, [3]rakshanaworld0812@institute.com, [4]srivatsan.rajesh@gmail.com

Abstract

The traditional approaches for managing leaves involve form-driven procedures and processes, resulting in high levels of admin workload, delays, and low levels of flexibility for users. In this paper, we introduce a novel Natural Language Driven Leave Management System where employees are able to ask their requests about leaves through natural language. The introduced framework uses the capabilities of LangChain agents, Gemini large language models, few-shot learning, and MongoDB to translate natural language into NoSQL operations. A mechanism of validation and fallback is used to guarantee schema, data consistency, and query reliability. The framework architecture includes four layers that enable to implement automation of leave processing and notification management. An experiment was performed for testing the framework by applying a number of representative queries in leave management. Configurations include Zero-Shot, Few-Shot, and Few-Shot with Validation cases. According to the results obtained, the developed framework achieved Intent Accuracy of 91%, Field Extraction Accuracy of 88%, NoSQL Query Accuracy of 87% and Fallback Success Rate of 93%.

Keywords: Leave Management, Lang Chain, Few-Shot Learning, Natural Language Processing, NoSQL, Fast API, Gemini LLM.

DOI: https://doi.org/10.36548/jaicn.2026.3.001
Received: 08.04.2026, received in revised form: 14.05.2026, accepted: 08.06.2026, published: 17.06.2026

1. Introduction

Leave management is a essential process in organizations that allows workers to apply for their leaves, while managers and human resources can control the availability of employees. The existing leave management solutions are usually designed around form-based processes that force users to use various interfaces and strict work protocols. These systems usually result in lengthy delays in the process, gaps in communication between employees and managers, and higher effort from administrators.

The recent advances in artificial intelligence, NLP, and LLMs allowed designers to develop conversational user interfaces that facilitate automated organizational processes through natural language dialogue. Using AI solutions, developers create more accessible experiences for both non-technical and tech-savvy users by letting them interact with applications via natural language. When applied to leave management, conversational UI can decrease the complexity of forms and facilitate natural communication between employees and management.

Although these technologies have been advanced significantly, many leave management systems suffer from inflexibility due to their SQL architecture or domain-specific designs. Additionally, many existing systems struggle in dealing with vague and ambiguous natural language inputs. Furthermore, intelligent validation systems and flexible database operations are crucial to most contemporary enterprises. Hence, there is a gap within current leave management systems, which is not capable of understanding user queries and providing efficient workflow automations through structured database operations.

To solve the above problems, we propose a natural language-driven leave management system that integrates LangChain agents, few-shot learning techniques, and NoSQL database system. Under our design, employees would be able to initiate their leave requests through natural language and then turn them into structured JSON commands based on a NoSQL database. Moreover, the leave management system includes validation processes, notifications, customized leave policies, and analytics dashboards.

The major contributions of this work are as follows:

- The creation of a leave management system that is based on natural language interaction between users and the system.

- The employment of LangChain agents and the use of few-shot prompts to ensure proper translation of natural language instructions into NoSQL queries.
- The incorporation of validation techniques and error handling for better accuracy in query generation and execution.
- The availability of automatic alerts and analytical dashboards for facilitating effective leave management and workforce planning.
- This proposed system seeks to enhance the user-friendliness, adaptability, and automation of leave management processes through artificial intelligence.

2. Related Works

Employee absenteeism has long been viewed as a major organizational problem that impacts both productivity and efficiency. As such, effective absence management practices must be in place in order to enhance organizational performance while reducing expenses [1]. Yet, current leave management practices are mainly focused on process management and do not account for technological development and innovation.

The emergence of conversational artificial intelligence technologies has created new opportunities to address employee leave management and approve corresponding requests via conversational interfaces. Thus, one of the studies showed that leave management practices could benefit from incorporating natural language conversation into their functioning by automating certain processes, increasing responsiveness, and enhancing efficiency [2]. The same can be stated about the adoption of an automated leave management system that was designed to support efficient administration in the academic setting [3]. Moreover, the implementation of a multitenant leave management system based on conversational bots and natural language processing was proposed to facilitate the management of employee leaves and automate corresponding procedures [4].

The incorporation of natural language processing with database technologies has led to even more powerful leave management tools. An AI agent framework that used LangChain was developed to convert natural language inputs to executable SQL queries to allow non-technical employees to access organization-specific databases [5]. The ability to build conversational AI systems using LangChain has been shown to be an efficient way to apply

large language models in automating organizational operations and increasing interactivity through natural language [15]. Nevertheless, query reliability and accuracy, as well as schema adaptability, still remain key challenges for researchers.

Text-to-SQL generation research has made a solid contribution to building natural language applications for enterprises. Previous studies have utilized reinforcement learning methods to generate accurate SQL queries from natural language inputs, leading to a considerable increase in the accuracy of the queries [6]. Other studies have incorporated meta-learning and one-shot learning to lower the amount of data necessary to train models to generate queries [7], [8]. However, the latter approach had some difficulties in adapting the models to novel databases.

The increased attention toward few-shot learning has made further improvements in query generation from natural language. Few-shot learning methods and their applicability to text-to-SQL generation using deep learning algorithms have demonstrated high effectiveness due to the usage of schema-aware modeling, constrained decoding by grammatical constraints, and prompt-based training [9], [10]. The most recent works showed that a well-designed prompt and context-based learning method could help to achieve better results on the task of text-to-SQL conversion using smaller annotated datasets [11], [12]. Also, few-shot learning has been proven to be applicable for SQL errors correction and refinement of the queries [13]. Comparative analysis also proves that context-based learning and few-shot parameter optimization provide effective solutions for the deployment of big language models in specific cases with scarce amounts of data [14].

Overall, there is substantial advancement in the field of conversational artificial intelligence, leave management automation, natural language queries processing, and few-shot learning. Nevertheless, a great deal of solutions still lack generic nature or rely on a specific, rather inflexible, SQL approach, or need vast amounts of data for training purposes. The proposed natural language-driven leave management system overcomes all the mentioned problems through employing LangChain agents, few-shot learning technique, and NoSQL database operations within one solution. Thus, it allows for the submission of a leave request from an employee in natural language form, converting it into structured NoSQL operations, configuring policies, and performing automatic notification and analysis operations.

3. Proposed Work

3.1 System Architecture

Proposed Natural Language Driven Leave Management System incorporates conversational artificial intelligence, large language models (LLM) and NoSQL databases. The system architecture has been designed based on four different layers: User Interface Layer, Application Layer, Intelligent Processing Layer, and Data Layer.

Figure 1 illustrates the overall architecture of the proposed Natural Language Driven Leave Management System.

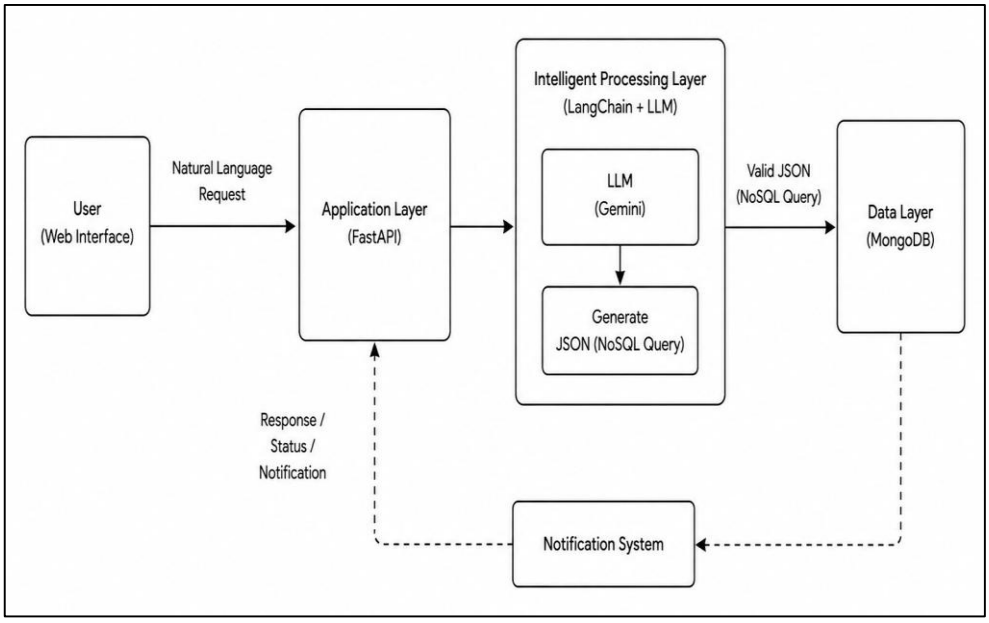

Figure 1. Architecture of the Proposed System

User Interface Layer will provide two dashboards for employees and HR managers. Employees will be able to send leave requests, keep track of leave statuses and receive notifications through a web application interface. HR managers will be able to review leave requests, approve and reject applications along with monitoring statistics regarding leaves from an administrative dashboard.

Application Layer is developed with the help of FastAPI and acts as an orchestrator. This layer will manage authentication, route requests, notification services, enforce leave policies and perform communication between the frontend and AI components as well as database services.

Intelligent Processing Layer uses LangChain Agent in conjunction with Gemini LLM. This layer helps in understanding natural language leave requests, performing intent recognition, identifying relevant entities such as leave category and dates, and creating structured JSON objects. Few-shot prompting is used to achieve accuracy and consistency while generating queries in accordance with the predefined database structure.

The Data Layer makes use of MongoDB as the backend NoSQL database. Information about the employees, leave applications, approvals, and the logs of the system will be stored in a document collection. Prior to executing the JSON output generated, the schema validation process will take place.

3.2 NL-to-NoSQL Processing Workflow

The complete interaction sequence among the user interface, LangChain agent, Gemini LLM, validation module, and MongoDB database are illustrated in Figure 2.

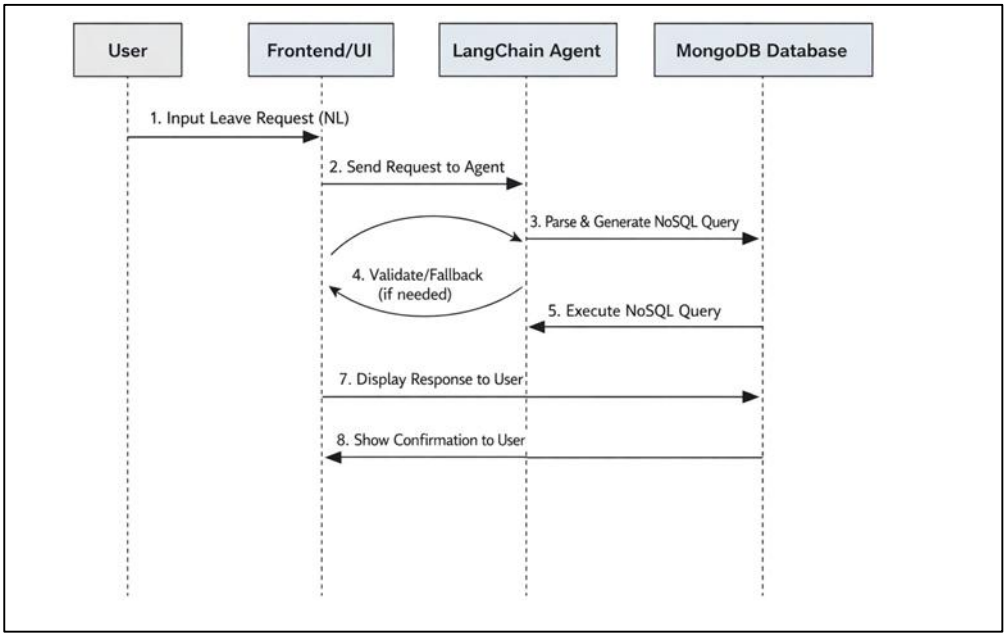

Figure 2. Sequence Diagram for NL-to-NoSQL Processing

In this process, the request comes from an employee in natural language via the web interface. In contrast to previous form-based systems, users can give their requests in plain English, for example, “Apply casual leave from March 10 to March 10.” LangChain Agent receives the request and constructs a prompt from the system instruction, database schema details, and few-shot examples.

This prompt is fed to Gemini LLM, where the system finds the intention behind the request and identifies the important entities mentioned therein. According to this information, it generates a structured JSON object representing the database operation to be performed. However, before executing the task, the created JSON object is verified for mandatory fields, type of data, date validation, and organizational leave policy restrictions.

In case of any error, the system triggers a clarification dialogue and asks the user for more information. The validated JSON then gets transformed into an action in MongoDB and stored in the leave database. Upon successful execution, the notification system alerts the respective manager about the request. Approval or rejection results in update in the database and a respective message to the employee.

3.3 System Implementation

The design of the proposed system will be accomplished using advanced software tools and AI technologies. The main software packages utilized in the building of the proposed system and their functions are presented in table 1 below.

Table 1. Software Components Used in the Proposed System

Package/Tool	Category	Function/Role in Project
Python 3.x	Programming Language	The main development language was selected because it has strong support for data science, AI, and web development frameworks.
Gemini API	Large Language Model (LLM)	The special NLP engine for the project is used for strong natural language understanding (NLU) and for creating structured NoSQL queries in JSON or BSON format.
LangChain	Orchestration Framework	Acts as the AI application layer. It coordinates the Gemini API, few-shot examples, and the NoSQL database connection to create the complete NL-to-NoSQL agent pipeline.
Few-Shot Learning Setup	AI Technique (Prompting)	Uses in-context learning to give the Gemini model specific, labeled examples. This greatly improves the accuracy and consistency of the generated NoSQL queries.

NoSQL Connector	Database Interface	Specific Python libraries, like PyMongo for MongoDB, are used to set up and run the NoSQL query output produced by the LangChain agent.
Flask / FastAPI	Backend Framework	Used to create the high-performance API endpoints that handle employee requests and manage the application logic.
Git/GitHub	Version Control	Essential for managing source code, working together, and keeping the deployment pipeline intact.

The proposed solution consists of Python, FastAPI, MongoDB, Gemini, and LangChain. Specifically, MongoDB will serve as the database storing the information about users' activities. Gemini will represent the model utilized for processing the incoming request, while LangChain will be responsible for the orchestration of prompts creation, model invocation, validation, and database interaction.

One of the crucial aspects of the implementation of the designed system is the few-shot prompting approach. In this way, there is no need to retrain the large language model. Instead, several samples of natural language leave requests along with the resulting JSON documents will be provided in the input prompt, which allows the model to learn the output format in context. The JSON document created will include the detected action, the user's information, type of leaves, dates, and other required information. The validation step is performed based on structural and logical criteria.

The NL-to-NoSQL transformation process is formally described in Algorithm 1.

Algorithm for NL to NoSQL Transformation

Input : Natural language query Q
Output: Execution result R
Begin
Preprocess(Q)
P ← BuildPrompt(Q, Schema, FewShotExamples)
J ← LangChainAgent(P)
if Validate(J) = FALSE then
C ← GenerateClarification()
Q ← GetUserResponse(C)
Repeat process
end if

M ← ConvertToMongoOperation(J)
R ← Execute(M)
Notify(Employee, Manager)
Return R
End

3.4 Evaluation Metrics and Validation

The performance of the proposed Natural Language Driven Leave Management System has been assessed based on four metrics: Intent Accuracy (IA), Field Extraction Accuracy (FEA), NoSQL Query Accuracy (NQA), and Fallback Success Rate (FSR).

Intent Accuracy measures the ability of the system to correctly identify the user's intent from a natural language request.

$$IA = \frac{N_{CI}}{N_Q} \tag{1}$$

where N_{CI} is the number of correctly identified intents and N_Q is the total number of queries.

Field Extraction Accuracy measures how accurately the system extracts required information such as leave type, dates, and duration.

$$FEA = \frac{N_{CF}}{N_{TF}} \tag{2}$$

where N_{CF} is the number of correctly extracted fields and N_{TF} is the total number of required fields.

NoSQL Query Accuracy measures the proportion of generated NoSQL operations that are syntactically correct and consistent with the intended database operation.

$$NQA = \frac{N_{VQ}}{N_Q} \tag{3}$$

where N_{VQ} is the number of valid NoSQL queries and N_Q is the total number queries.

Fallback Success Rate evaluates the effectiveness of the clarification mechanism when incomplete or ambiguous inputs are encountered.

$$FSR = \frac{N_{SC}}{N_{FC}} \quad (4)$$

where N_{SC} is the number of successfully resolved queries and N_{FC} is the total number of fallback cases.

In order to perform database queries with accuracy and reliability, every JSON query that is generated is first verified before being executed. This verification involves checking for compliance with schema, mandatory fields, data types, date consistency, and organization-specific rules of leave policy. Failure to comply with any of these aspects results in the triggering of a clarification cycle.

4. Results and Discussion

The proposed Natural Language Driven Leave Management System was implemented using representative leave-management scenarios. Figures 3 and 4 illustrate the developed web interfaces designed for employees and managers.

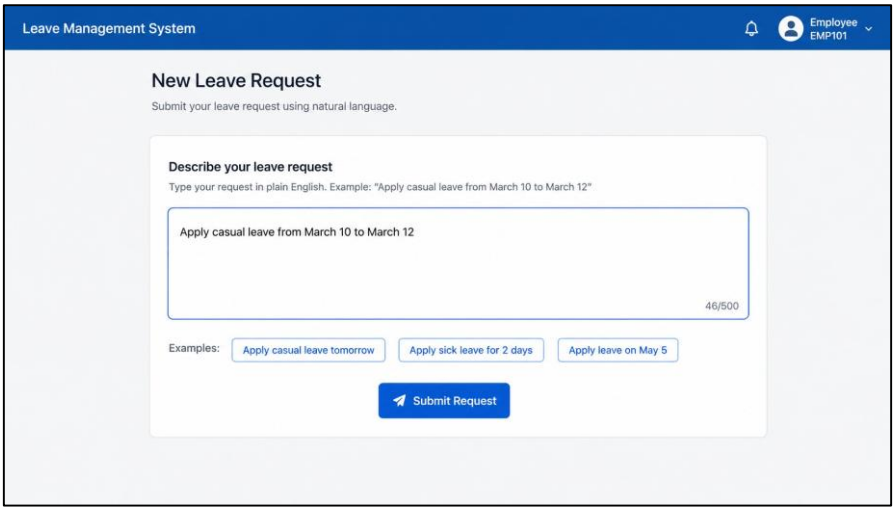

Figure 3. Employee Leave Request Interface

Figure 3 presents the employee leave request interface, where users can submit leave applications through natural language inputs. The interface simplifies leave submission by allowing conversational requests instead of conventional form-based interactions.

Figure 4 shows the manager dashboard, which provides a consolidated view of leave requests, approval statistics, pending applications, and leave trends. The dashboard supports

decision-making by offering real-time insights into workforce availability and leave utilization. The developed system was evaluated using a set of representative leave-management queries covering leave application, leave cancellation, leave-status enquiry, and leave modification scenarios.

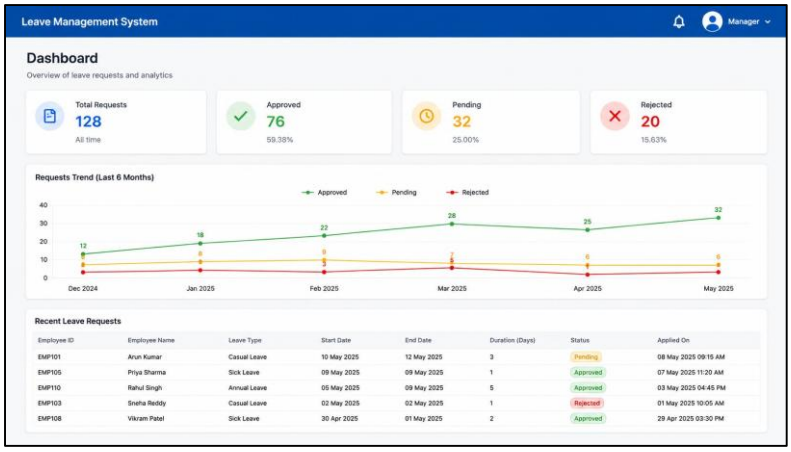

Figure 4. Manager Dashboard for Leave Analytics

The evaluation focused on the system's ability to correctly identify user intent, extract relevant information, and generate valid NoSQL operations.

Three experimental configurations were considered:

1. Zero-Shot – Gemini LLM without in-context examples.
2. Few-Shot – Gemini LLM with curated few-shot examples embedded in the prompt.
3. Few-Shot + Validation – Few-shot prompting combined with schema validation and fallback clarification mechanisms.

As illustrated in Table 2, the performance of each configuration was evaluated using IA, FEA, NQA and FSR.

Table 2. Performance Comparison of Different Approaches

Method	IA	FEA	NQA	FSR
Zero-Shot	0.72	0.65	0.6	0.52
Few-Shot	0.86	0.81	0.78	0.74

Few-Shot + Validation	0.91	0.88	0.87	0.93

The results shown in Table 2 indicate that few-shot prompting greatly boosts the effectiveness of the system regarding interpreting leave request and generating correct NoSQL queries. Relative to the Zero-Shot configuration, few-shot prompts enhance Intent Accuracy from 72% to 86%, Field Extraction Accuracy from 65% to 81%, and NoSQL Query Accuracy from 60% to 78% and Fallback Success Rate from 52% to 74%. As it can be seen, the Few-Shot + Validation configuration performs better than other configurations, delivering Intent Accuracy of 91%, Field Extraction Accuracy of 88%, NoSQL Query Accuracy of 87% and Fallback Success Rate of 93%.

Validation and fall-back capabilities increase system reliability as they help detect lack of information, date inconsistency, and schema breaches prior to database processing. Consequently, errors are avoided, and accuracy of generated queries is improved. Out of all assessed metrics, NoSQL Query Accuracy shows the highest gains, improving by 27 percentage points compared to the Zero-Shot setup. Hence, the integration of few-shot learning and validation mechanisms enables reliable generation of natural language-based NoSQL queries.

5. Conclusion

This study discussed an NLP Driven Leave Management System that utilizes LangChain Agents, Gemini LLM, few-shot learning, and MongoDB for leave request automation using natural language processing techniques. The model proposed in this paper helps the employees make leave requests in the form of conversational English, which is automatically translated to NoSQL operations. This system has employed a validation process in order to make sure that the queries are accurate and consistent with the database schema. The experiments performed showed that the few-shot prompting technique outperforms zero-shot by improving intent recognition, field extraction, and NoSQL query generation accuracy. The optimal experimental configuration was Few-Shot with Validation, which had an Intent Accuracy of 91%, Field Extraction Accuracy of 88%, NoSQL Query Accuracy of 87% and Fallback Success Rate of 93%. Further research will aim to implement multilingual capabilities and integration with corporate human resources management systems.

References

[1] J. Gajda, “Employee Absence Management in Modern Organization,” Journal of US-China Public Administration 2015, vol. 12, no. 7, 540-547.

[2] Crave InfoTech, “Case Study: Conversational AI for Leave Management – Helped Boost Efficiency,” 2022. https://www.craveinfotech.com/blogs/case-study-conversational-ai-for-leave-management-helped-boost-efficiency/

[3] M.Vikash, S.V.Pirathik, Mrs.M.Gayathri Devi. “AI Powered Institutional Management System For Leave Request Processing”. International Journal of Sciences and Innovation Engineering 2025, vol. 2, no. 5, 498-501.

[4] Aayush Bhaskarwar, Akshat Dhanuka, Ziyaad Parkar, S.P. Shintre, “Multitenant Leave System,” International Journal of Innovative Research in Technology (IJIRT) 2025, vol. 11, no. 12, 8663-8672.

[5] V. Shanker, M. A. Khan, I. Karovalia, H. A. Shaikh, and A. Jha, “Natural Language SQL Querying via LangChain and AI Agent,” International Journal of Creative Research Thoughts (IJCRT) 2025, vol. 13, no. 4, 723-731.

[6] Zhong, Victor, Caiming Xiong, and Richard Socher. "Seq2sql: Generating structured queries from natural language using reinforcement learning." arXiv preprint arXiv:1709.00103 (2017).

[7] Huang, Po-Sen, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. "Natural language to structured query generation via meta-learning." Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2018, vol 2, 732-738.

[8] Lee, Dongjun, Jaesik Yoon, Jongyun Song, Sanggil Lee, and Sungroh Yoon. "One-shot learning for text-to-sql generation." arXiv preprint arXiv:1905.11499 (2019).

[9] Yin, Wenpeng. "Meta-learning for few-shot natural language processing: A survey." arXiv preprint arXiv:2007.09604 (2020).

[10] Katsogiannis-Meimarakis, George, and Georgia Koutrika. "A survey on deep learning approaches for text-to-SQL." The VLDB Journal 2023, vol. 32, no. 4, 905-936.

[11] Nan, Linyong, Yilun Zhao, Weijin Zou, Narutatsu Ri, Jaesung Tae, Ellen Zhang, Arman Cohan, and Dragomir Radev. "Enhancing few-shot textto-sql capabilities of large language models: A study on prompt design strategies." arXiv preprint arXiv:2305.12586 (2023).

[12] Sun, Ruoxi, Sercan O. Arik, Rajarishi Sinha, Hootan Nakhost, Hanjun Dai, Pengcheng Yin, and Tomas Pfister. "Sqlprompt: In-context text-to-sql with minimal labeled data." In Findings of the Association for Computational Linguistics: EMNLP 2023, 542-550.

[13] Jain, Divyansh, and Eric Yang. "Context-Aware SQL Error Correction Using Few-Shot Learning--A Novel Approach Based on NLQ, Error, and SQL Similarity." arXiv preprint arXiv:2410.09174 (2024).

[14] Liu, Fengchen, and Gary Jung. "Exploring Few-Shot Learning: Fine-Tuning vs. In-Context Learning and Parameter-Efficient Adaptations." In Practice and Experience in Advanced Research Computing: The Power of Collaboration 2025, 1-4.

[15] Pandya, Keivalya, and Mehfuza Holia. "Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations." arXiv preprint arXiv:2310.05421 (2023).