QUESTION AND ANSWER BASED COMMUNICATION SYSTEM FOR INDIAN RAILWAYS IN HINDI

Dr. Lovely Sharma
Assistant Professor
PG Department of Computer Science & IT, PCM S.D College for Women, Jalandhar, India

ABSTRACT

Purpose: *To discover information, users often rely on the wide-ranging Google search results and sit through quite many long documents as well as web links. Meanwhile, the widespread use of these results shows the increasing importance of Question Answering (QA) dialogue systems: where an immediate and concise answer can be given.*

Method: *We will pay attention to part semantic and syntactic analysis combined with a keyword-based approach. The Dialogue Manager serves as the central component, facilitating interaction between the user and the system, because this is where such challenges as anaphora, co-reference resolution take place.*

Conclusions: *Hence, the most widely used Indian Railway system is a more click based interface and is time consuming. We motivate further investigation in this direction and conclude that the developed QA dialogue system provides users targeted, concrete responses. At the moment, the system responds to relevant inquiries, but future improvements could enable it to tell callers, for instance, when trains will arrive and depart.*

Keywords: Question &Answering System, Natural Language Processing, Artificial Intelligence, Railway, Normalization, Keyword based Approach, Dialogue Manager

1. INTRODUCTION

QA dialogue systems are designed to deliver concise and precise answers to arbitrary queries expressed in natural language. Unlike traditional search engines, which return lengthy documents containing relevant information, QA systems focus on providing direct responses to user inquiries. For instance, when a user asks, " How much is the fare for Amritsar Express from Jalandhar to Amritsar?" a QA system would ideally respond with "250 Rs," eliminating the need for users to sift through extensive text. This intuitive approach not only enhances efficiency but also significantly reduces the time required to obtain relevant information.

Due to these advantages, substantial research has been conducted in various fields, including biology [1] and chemistry [2], to develop domain-specific QA systems. However, despite the progress, many practical QA systems remain underutilized in everyday life. Recognizing this gap, we chose to focus on the railway domain, specifically in Northern India [1], where users frequently seek information related to train schedules, fares, and routes. These users often prefer receiving information in their native language without the need for extensive linguistic proficiency. This need is effectively addressed through a QA dialogue system, which facilitates seamless interactions by delivering accurate and succinct answers in Hindi.

To achieve this goal, we developed the Question Answering Railways System (QuARS), a QA system tailored for handling railway-related inquiries in Northern India. QuARS follows a method based on keyword identification, incorporating limited linguistic processing on natural language queries. The system is designed to understand user queries from a linguistic perspective, ensuring it accurately interprets and responds to user intentions.

QA dialogue systems primarily function through interactions between users and the system. The objective of implementing the QA system architecture is to provide precise answers while promoting the usage of Hindi language on digital platforms. With an estimated 4,000 spoken languages worldwide, Hindi ranks as the fifth most widely spoken language globally among the top 100 languages [3]. Given its widespread use, developing a robust QA system in Hindi enhances accessibility for millions of users.

The QA system comprises multiple processing stages, as illustrated in Figure 1. These include Query Processing, Query Frame Analysis, Dialogue History Management, Dialogue Manager, SQL Query Generation, and Answer Generation. This paper introduces QuARS and outlines the motivations behind designing a QA dialogue system. The subsequent sections provide an overview of the QA system, a detailed explanation of query processing techniques, evaluation methodologies, and concluding remarks.

The Text Retrieval Conference (TREC) is a series of workshops dedicated to advancing research in Information Retrieval (IR). TREC plays a pivotal role in supporting and motivating the IR research community by setting challenges for participating teams. These challenges encompass various tasks, such as document retrieval, question answering, and access to specific data points. Thanks to TREC, modern QA systems have evolved to answer over two-thirds of factual questions with high accuracy. Researchers utilize these workshops to present innovative ideas, share research findings, and collaborate on enhancing QA technologies.

2. LITERATURE REVIEW

Over time, numerous developers have proposed systems to assist users in retrieving information, though often within restricted domains. One of the earliest and most well-known domain-specific QA systems is BASEBALL [4], which provided answers related to American League statistics for single period of time. Another notable Question Answering systems include START [2] and Wolfram Alpha [3], both of which focus on providing direct answers to user queries.

SHRDLU [5] is recognized as one of the first AI systems capable of accepting user queries and generating responses based on contextual understanding. Another significant system in this domain is LUNAR [6], which facilitated easy access to data from the chemical examination of lunar rock samples. Early dialogue systems such as UC (Unix Consultant) [7] and The Berkeley UNIX Consultant project were designed to assist users by answering questions related to the UNIX operating system [8].

Beyond these historical developments, considerable progress has been made in developing QA systems that support Indian languages. Researchers have actively contributed to creating multilingual QA models that cater to diverse linguistic needs. For example, Sravanthi et al. [11] developed a model aimed at improving data management and query processing in the tourism sector. More recently, Vignesh et al.[12] introduced the Hidden Markov Model (HMM) for speech recognition in Tamil. Several other researchers have also contributed to developing multilingual QA systems, including Hindi-Dogri [14], Punjabi [15], and Hindi-Kannada language models [16].

The need for Indian language QA systems is particularly crucial given the diverse linguistic landscape of the country. Users often prefer engaging with systems that communicate in their native language, ensuring a seamless and efficient information retrieval experience. The advancement of such systems promotes digital inclusivity, making information more accessible to non-English speakers.

QuARS is built to address these specific requirements, providing an effective solution for railway-related inquiries in Hindi. The system ensures that users receive accurate and relevant responses without requiring proficiency in English or technical knowledge of query structures. By employing a structured approach to query processing, QuARS enhances user experience and accessibility in the Indian railway domain.

To summarize, QA dialogue systems serve as a valuable tool for providing succinct answers to user queries, reducing reliance on lengthy documents and complex search results. QuARS exemplifies this approach by offering a Hindi-language railway inquiry system tailored to the needs of users in Northern India. The system follows a keyword-based methodology and incorporates linguistic analysis to ensure accurate and meaningful responses. Through continued research and development, QA systems like QuARS can further improve digital accessibility, making information retrieval more efficient for diverse user groups.

A Question-Answering Dialogue System for emergency operations enhances crisis response by providing accurate, real-time information. It processes natural language queries, aiding decision-making through automated responses, ensuring efficiency, and improving communication during emergencies for faster and more effective disaster management [17].

The paper [19] explores how interactive question answering systems (IQAS) merge the concepts of QA and dialogue systems. It covers the architecture, evaluation, and different modalities of interaction, such as text or speech. The paper also classifies QA systems based on tasks and challenges. Evaluation can be either offline or online, using datasets like TREC or WikiQA.

A Comprehensive Survey[20] on Open-domain Question Answering reviews recent trends in Open-domain Question Answering (OpenQA) within NLP. It highlights how neural Machine Reading Comprehension (MRC) techniques have advanced performance, introducing the Retriever-Reader architecture. The paper also discusses challenges in OpenQA, benchmarks used, and aims to guide researchers through the evolution of OpenQA systems. This work provides a comprehensive overview of architecture, techniques, and key issues in the field.

AgriBot, an agriculture-specific question-answering system [21] helps farmers by providing information related to weather, market rates, plant protection, and government schemes. It uses a sentence embedding model, achieving about 56% accuracy initially, which improves to roughly 86% after incorporating entity extraction. The system is available 24/7 and is accessible via electronic devices. It aims to ease the burden on Kisan Call Center workers and improve agricultural productivity.

3. Components of QuARS

The systematic diagram consists of various parts as shown in Figure 1 provides the detailed mechanism of Question Answering Dialogue system in Hindi language.

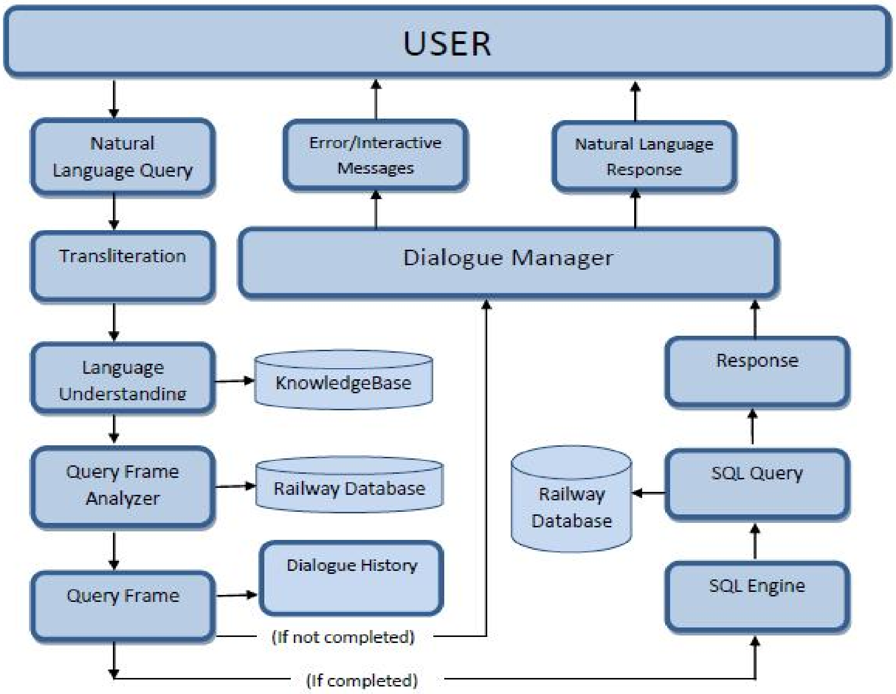

Fig 1. Detailed Structure of QuARS

QUERY PROCESSING

Processing natural language is an intricate and challenging task. The system must incorporate linguistic knowledge to analyze queries effectively. It involves interconnected modules that work together to generate quick and concise responses. The initial phase of this pipeline is the query input stage. This step comprises two key modules: the NL query and Transliteration as shown in above Figure 1. The module Natural Language query processes user input, while the Transliteration module converts it into Hindi Language. Below is an example illustrating how a user query is input and transliterated into Hindi.

Amritsar Express main mera kitna kiraya lagega?

↓

After Transliteration

↓

अमृतसर एक्सप्रेस में मेरा कितना किराया लगेगा ?

Following this, linguistic processing operates at three levels within the QA architecture: (i) Tokenization (ii) Normalization and (iii) Keyword Selection as illustrated in Figure 2.

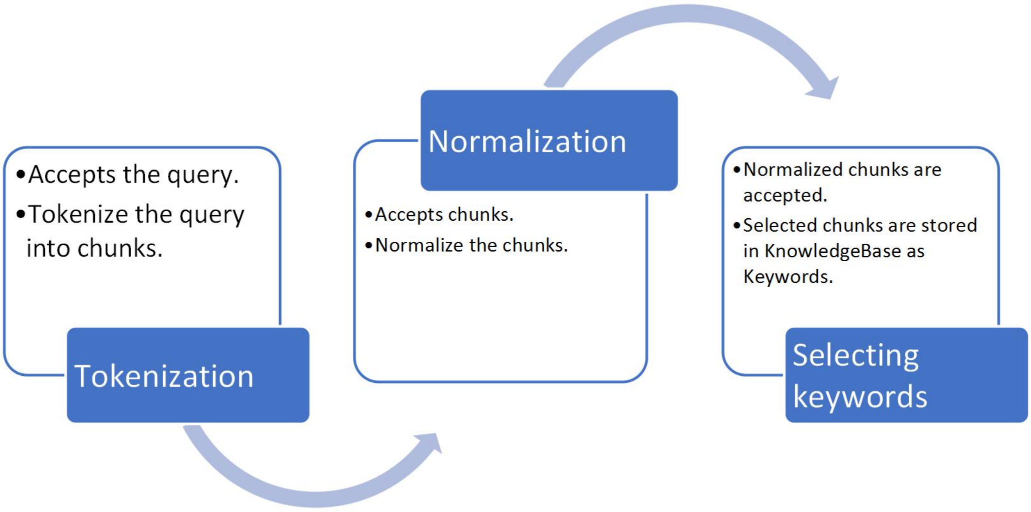

Figure 2. Module of Query Processing.

TOKENIZATION

Developing natural language processing is highly complex, requiring linguistic knowledge for accurate query analysis. The system must effectively interpret language patterns to ensure precise understanding and response generation. According to our QA architecture, linguistic processing occurs at three levels: (i) Tokenization, (ii) Normalization, and (iii) Keyword Selection, as shown in Figure 2.

अमृतसर एक्सप्रेस में मेरा कितना किराया लगेगा ?

↓

After Tokenization

↓

अमृतसर ,एक्सप्रेस, मेरा ,कितना, किराया ,लगेगा

NORMALIZATION

Input queries may contain many non-standard word representations. Non-Standard Words (NSW) can arise from various dialects, foreign language influences, and alphabet variations, leading to different forms of equivalent terms. These phonetic deviations or non-standard lexical representations impact language processing and require effective handling for accurate interpretation. As seen in the previous example, variations may arise in किराया (Fare) word as shown below:

किराया (kiraya), किरया (kiaraya), किराय(Kiraya), कीराय (kiiraya), कीराया (Kiiraya)

Another example highlights common spelling variations of the station name 'Jalandhar,' as below.

जालंधर (Jaalandhar), जालन्धर (Jaalandhar), जलन्धर (jalandhar), जलंधर (jalandhar)

We identify various word variations in the language and develop solutions by formulating specific rules to address them. This approach enhances system accuracy by selecting the correct word from the knowledge base while eliminating all other variations.

KEYWORD SELECTION

In the previous stage, chunks are normalized. Before acceptance, each chunk is compared against the knowledge base using a lookup table method until a match is found. The matched word is then converted into a keyword, helping the system determine the purpose of the natural language query.

अमृतसर एक्सप्रेस में मेरा कितना किराया लगेगा ?

(Amritsar Express main mera kitna kiraya lagega?)

Keyword = किराया (Fare)

In the preceding example, किराया serves as keyword, indicating the User's query purpose that user seeks information about the train fare.

QUERY FRAME ANALYSER

The accuracy of the system heavily depends on selecting the correct query frame. In this phase, keywords are identified and processed, such as system recognizing किराया (Fare) (as explained earlier). This keyword is then mapped to the knowledge base using a lookup table technique to determine the adequate query frame [10]. Every natural language query is associated with a distinct query frame. In the given example, the किराया (Fare) query frame includes the following slots:

Table 1: Query Frame for Fare

Fare	Assigned values
Train Name/Code	अमृतसर एक्सप्रेस
Source	जालंधर

Destination	अमृतसर
Age	30
Class	स्लीपर

Similarly, additional examples of pre-defined query frames include class of coach, discounts and waiting list etc. The appropriate of pre-defined query frames is carefully selected from already existed knowledge base using these keywords.

DIALOGUE MANAGER

In some instances, slot-filling may be incomplete due to inadequate information provided by user. When this occurs, the system must seek additional details to ensure accurate query processing. To address this, the Dialogue Manager (DM) plays a crucial role by initiating an interactive conversation with the user. It sends prompts or clarification requests, encouraging the user to provide the necessary missing details. This interaction ensures that the system collects complete and relevant data before proceeding with query execution.

As shown in Table 2, the DM not only facilitates user-system communication but also regulates the overall dialogue flow. It determines when and how the system should respond based on the user's input, ensuring smooth and logical interactions. By controlling the sequence of exchanges, the DM enhances the system's ability to understand user queries accurately and generate precise responses.

This dynamic interaction mechanism significantly improves the efficiency and usability of the QA system. By actively engaging with users to resolve incomplete inputs, the DM enhances the overall user experience, leading to more accurate query resolutions and improved satisfaction levels. Thus, the DM serves as a bridge between the user and the system, ensuring effective and meaningful communication.

Table 2: SQL Generation

User's Query	Related Frame	Assigned Value
मैं: अमृतसर एक्सप्रेस में मेरा कितना किराया लगेगा ?	रेलवे. अमृतसर एक्सप्रेस (**ट्रेन नाम/ कोड**,?,?,?,?)	रेलवे. अमृतसर एक्सप्रेस (**अमृतसर एक्सप्रेस**,?,?,?,?)
सिस्टम :आपने कहाँ से चलना है? मैं: जालंधर	रेलवे. अमृतसर एक्सप्रेस (ट्रेन नाम/ कोड,**चलना**,?,?,?)	रेलवे. अमृतसर एक्सप्रेस (अमृतसर एक्सप्रेस, जालंधर,?,?,?)
सिस्टम :आपने कहाँ तक जाना है? मैं: अमृतसर	रेलवे. अमृतसर एक्सप्रेस (ट्रेन नाम/ कोड,चलना, **पहुंचना**,?,?)	रेलवे. अमृतसर एक्सप्रेस (अमृतसर एक्सप्रेस, जालंधर, अमृतसर,?,?)
सिस्टम :आपकी कितनी आयु है? मैं : 30	रेलवे. अमृतसर एक्सप्रेस (ट्रेन नाम/ कोड,चलना, पहुंचना,**आयु**,?)	रेलवे. अमृतसर एक्सप्रेस (अमृतसर एक्सप्रेस, जालंधर, अमृतसर, 30,?)
सिस्टम: आपने किस श्रेणी	रेलवे. अमृतसर एक्सप्रेस	रेलवे. अमृतसर एक्सप्रेस

में सफ़र करना है? मैं :स्लीपर	(ट्रेन नाम/ कोड,चलना, पहुंचना, आयु, **श्रेणी**)	(अमृतसर एक्सप्रेस, जालंधर, अमृतसर,30,स्लीपर)

In this phase, The formulation of an SQL query is contingent upon the comprehensive and accurate completion of the query frame. Two possible scenarios can occur during this process: (a) the query frame remains incomplete, or (b) the query frame is fully constructed. If the query frame is incomplete, the system cannot generate an accurate SQL query. In such cases, the Dialogue Manager (DM) plays a crucial role by engaging with the user to request additional details. Through an interactive exchange, the DM ensures that all necessary information is collected before proceeding further.

On the other hand, if the query frame is complete, the system moves forward with generating a structured SQL query. This query is then executed on the database to retrieve the most relevant information, ensuring the user receives a precise and concise response. The generated SQL query plays a key role in translating natural language queries into structured database interactions. Table 3 demonstrates how a natural language query is systematically converted into an SQL query. By structuring user input into well-defined query frames, the system ensures accurate and efficient data retrieval, ultimately enhancing the overall performance and reliability of the QA system.

Table 3:Answer Generation

Steps/Modules	Example
User's Query	अमृतसर एक्सप्रेस में मेरा कितना किराया लगेगा ?
Keyword	किराया (Fare)
Query Frame	ट्रेन नाम/ कोड,चलना, पहुंचना, आयु, श्रेणी (train name/code, source, destination, age, class)
Assigned values	रेलवे. अमृतसर एक्सप्रेस (अमृतसर एक्सप्रेस, जालंधर, अमृतसर,30,स्लीपर)
SQL Query	select fare from TrainFare where (train_name= 'अमृतसर

	एक्सप्रेस',source_stat= 'जालंधर',dest_stat= 'अमृतसर', age=30,class= 'स्लीपर')

Users prefer concise answers over web links or lengthy documents. As explained earlier, An SQL query is formulated and subsequently executed on the database to facilitate the precise retrieval of relevant information. The retrieved data is then forwarded to the Dialogue Manager (DM), which converts the SQL output into a natural language statement for user-friendly interaction. This ensures that the user receives clear and direct information rather than raw database results. For example, in the previous case, the system processes the query and provides the response as "230 Rupees," as demonstrated in Table 4, ensuring a seamless and intuitive user experience.

Table 4. Evaluation and Result

select fare from TrainFare where (train_name= 'अमृतसर एक्सप्रेस',source_stat= 'जालंधर',dest_stat= 'अमृतसर', age=30,class= 'स्लीपर')
↓
230 रूपए

The true measure of a QA system's success lies in user satisfaction. Higher user satisfaction directly correlates with a higher success rate of the system. However, commonly used evaluation metrics include Recall and Precision values [8].

Recall = (Number of correct responses produced by the system / Number of Natural queries given to the system) * 100.

Precision = (Number of correct responses produced by the system / Number of responses produced by the system) * 100.

Number of correct responses produced by the system=146

Number of responses produced by the system=149

Number of Natural queries given to the system=150

Precision= (146/149)*100=96.5%

Recall= (146/150)*100=95%

The implemented system was evaluated by inputting 150 natural language queries into the QA system. The evaluation results demonstrated high performance, with the system achieving a Precision rate of 96.5% and a Recall rate of 95%.

4. CONCLUSION

This paper presents the implementation of QuARS, a keyword approach-based system in Hindi language is based on Northern Indian Railway. The entire system primarily focuses on the Query Processing module to ensure accurate and efficient responses. A detailed analysis of the input query is conducted to achieve this. The process involves three key steps: Tokenization, Normalization, and Keyword Selection. The extracted keywords play a crucial role in identifying the adequate query frame. SQL query is produced in the subsequent stage based on that query frames, allowing QuARS to retrieve the most relevant information. Finally, the processed information is converted into a natural language response and delivered to the user in Hindi. This ensures that the system is both user-friendly and efficient in handling railway-related queries. By employing this structured approach, QuARS enhances the accuracy and accessibility of railway information for Hindi-speaking users, streamlining the process of obtaining essential travel details. The integration of keyword-based processing with structured query generation significantly improves the effectiveness and usability of the system, providing a seamless experience for users seeking railway-related information.

REFERENCES

1. Barker, K., Chaudhri, V. K., Chaw, S. Y., Clark, P.,Fan, J., Israel, D., Mishra, S., Porter, B. W., Romero, P., Tecuci, D. , and Yeh, P. Z., A question-answering system for AP Chemistry: Assessing KR&R technologies. In Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning , pages 488–497, 2004.
2. Black, A., Richards, M. O. C. , Sproat, R., Chen, S., Kumar, S., “Normalization of Non-Standard Words”, Computer Speech and Language, 15(3):287.333, 2001.

3. Chin D.N. Knowledge Structures in UC, the UNIX Consultant, In Proceedings of Twenty-first ACL, 1983, pp 159-63.
4. Dave S., Bhattachary P. & Klakowya D., (2001), “Knowledge Extraction from Hindi Text”, Journal of Institution of Electronic and telecommunication engineers, 18(4). http://www.cse.iitb.ac.in/~pb/papers/iete.pdf.
5. Green B. F., Wolf A. K., Chomsky C., and Laughery K. Baseball: An automatic question answerer, Computers and Thought. 1963, pages 207–216.
6. Harman D. The Text REtrieval Conferences (TRECs). In Proceedings of a workshop, Vienna, Virginia, May 6-8, 1996.
7. Neves, M.& Leser, U., Question answering for Biology, Methods, vol. 74, 2015, p.36-46 .
8. Reddy, R., Reddy, N.and Bandyopadhyay, S. “Dialogue based Question Answering System in Telulgu”, Proceedings of EACL Workshop on Multilingual Question Answering, *Association for Computational Linguistics*, 2006, pp. 53-60.
9. SHRDLU, Galley: Article-00056.
10. Sinha R.M.K., “On design of a question-answering interface for hindi in a restricted domain”, International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, 2006, June 26-29.
11. Sravanthi M.C., Prathyusha K. and Mamidi R. A Dialogue System for Telugu, a Resource-Poor Language. Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science, Volume 9042, 2015, pp 364-374.
12. Vignesh G. and Ganesh S.S. , Tamil Speech Recognizer Using Hidden Markov Model for Question Answering System of Railways, Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, Advances in Intelligent Systems and Computing, Volume 325, 2015, pp 855-862.

13. Voorhees E. and Harman D.,Overview of the Eighth Text REtrieval Conference (TREC-8).In D.Harman, editor,The Eighth Text REtrieval Conference (TREC-),Gaithersburg, MD, USA, 2000. U.S. Government Printing O .ce,Washington D.C.

14. Webster J. J., Kit C., Tokenization as the initial phase in NLP, Proceedings of the 14th conference on Computational linguistics, August 23-28, 1992, Nantes, France [doi>10.3115/992424.992434].

15. Wilensky R., Chin D.N., Luria M., Martin J., Mayfield J., Wu D. The Berkeley UNIX consultant project. Computational Linguistics, Vol. -14 Issue no.-4, December 1988, pp 35-84.

16. Woods W. Progress in natural language understanding -an application to lunar geology. American Federation of Information Processing Societies (AFIPS), Conference Proceedings, 1973, pp 441-450.

17. Chan, Hao-Yung, and Meng-Han Tsai. "Question-answering dialogue system for emergency operations." *International Journal of Disaster Risk Reduction* 41 (2019): 101313.

18. Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. "Survey on evaluation methods for dialogue systems." *Artificial Intelligence Review* 54 (2021): 755-810.

19. Biancofiore, G. M., Deldjoo, Y., Noia, T. D., Di Sciascio, E., & Narducci, F. (2024). Interactive question answering systems: Literature review. *ACM Computing Surveys*, *56*(9), 1-38.

20. Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., & Chua, T. S. (2021). Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

21. Jain, N., Jain, P., Kayal, P., Sahit, J., Pachpande, S., & Choudhari, J. (2025). Agribot: agriculture-specific question answer system. *arXiv preprint arXiv:2509.21535*.