A Conversational AI: LSTM Based Tamil Chatbot system for Academic Information System

P. Karthik
Dept. of Information Technology
Chennai Institute of Technology
Email: karthikpit2019@citchennai.net

Kishore Kumar.L
Dept. of Information Technology
Chennai Institute of Technology
Email:kishorekumarlit2019@citchennai.net

Venkateswar. S
Dept. of Information Technology
Chennai Institute of Technology
Email:venkateswarsit2019@citchennai.net

B. Sundarambal
Center for Artificial Intelligence & Research
Chennai Institute of Technology
E-mail: sundarambalb@citchennai.net

R. Ponnusamy
Center for Artificial Intelligence & Research
Chennai Institute of Technology
E-mail: ponnusamyr@citchennai.net

Abstract

A Chabot is a system used to conduct an online conversation via text to speech and give the impression of actual human conversation. It is designed to simulate the system like human communication. The system should automatically answer all the queries raised by the user. Nowadays, the number of queries raised by aspiring students in the academic system increasing dramatically. It is very hard for the user to answer the queries raised by the aspiring students about the various courses offered by the academic institution. It is good to have an automated chatbot system to answer the queries raised by the students. To solve this problem a chatbot system is designed using Conversational AI. Specifically, the design of LSTM based system to order the sequence of words spoken by the system is effective. In addition to that usage of the NLP, toolkit makes the system more meaningful. Tamil has to follow a strict grammar rule. If these words are rearranged, the original meaning is lost. Thus, it is hard for processing the data for the computers when the original meaning is lost. Thus, one requires to train the models with specific structures and discard others. Therefore, in this work, an attempt is made to design an LSTM/Deep Learning-based system ordering the sentence is effective. It is designed to measure the user satisfaction rate and evaluation rate to ensure the effectiveness of the system.

Keywords: Chatbot, Conversational AI, Deep Learning, LSTM, user satisfaction, evaluation rate.

1. Introduction

In the open economy service offered to the people are increasing day by day. Most of the companies like to have an IVRS (Interactive Voice Response System) because of the growing demand for products and services. This IVRS needs lots of interaction from the customer. Therefore, the customers avoid such kind of system. Alternatively, it is impossible to provide 24X7 customer services to enquire made by the customer. Therefore it is essential to find an alternative mechanism to solve this issue, which is more interactive and should provide all required information in understanding user requirements.

Artificial Intelligence and Deep Learning is modern emerging technology providing various types of powerful solutions to these kinds of problems. Conversational AI is a set of technologies behind automated messaging and voice-enabled applications that enable human-like interactions between computers and humans. Conversational AI [9] can communicate like a human by recognizing languages and texts, understanding intent, decoding different languages, and responding in a way that mimics human conversation increase. In particular, it can work in a variety of languages.

Several attempts were made by the early researchers to design and develop Tamil Chabot to provide an interactive system to answer the queries in Tamil. All the existing systems are developed with a rule-based system or database-oriented one. Most of the time it will be not able to answer the new type of queries raised by the user. In this system, an attempt is made to develop a dynamic Chatbot with Conversational AI using the Rasa tool, which is capable of dynamically accepting different kinds of queries in Tamil. It is one of the most advanced developments in the Chabot environment. Chabot is an application that allows you to start and continue conversations using auditory and/or textual methods, much like humans. Chatbots are either simple rule-based engines or intelligent applications that use natural language understanding. Today, many companies are beginning to use chatbots extensively. Chatbots are becoming popular because they are available 24 hours a day, 7 days a week, provide a consistent customer experience, serve multiple customers at the same time, and are inexpensive to improve the overall customer experience.

In this paper, Section 2 reviews the literature, Section 3 describes the basics of Chatbot system and System Architecture Design in detail. Section 4 provides an overview experimental setup in section, 5 which explains the result and discussion. Section 6 concludes the paper with the study of the impact of the Tamil Chabot system.

2. Background Literature

The Tamil chatbot is an important tool for communicating with the system to enable the user to get any information from the system in all 24 x 7 hours. A vast amount of work has been carried out to classify the text worldwide by considering its importance using different modern tools. Several frameworks and algorithms are available on the web, leading to the development of Tamil Chatbot. This section deliberates works related to Chatbot systems development.

1. T. Kalaiyarasi, et al., in 2003 [1] developed a chatbot called Poongkuzhali that communicates in Tamil on a basis of giving an appropriate response to the given question or statement as input using artificial intelligence. The response will be given even when the question phrase is not given incorrect phrase using a set of decomposition rules and a set of reassembly rules in the knowledge base.
2. Raphael Meyer von Wolff, et al., in 2019 [2] attempt to conduct a structured literature review on research papers based on chatbots and says their current potentials, their objectives, the gap present in the research and to tackle them.
3. B.Galitsky, in 2019 [3] explains the use of Explainable ML and its features in the field of chatbot. It also show's a problem and its solution by using the transparent rule-based or ML method.
4. Daniil Sorokin, et al., in 2018 [4] approaches the entity linking in the context of question answering task and jointly optimized neural architecture for entity mention detection and entity disambiguation that models the surrounding content on a different level of granularity. The Wikipedia knowledge base is used as a dataset for question answering data training.
5. Hamid Zafar, et al., in 2018 [5] studies the question answering system over knowledge graphs which map an input into queries. By using Tree-LSTM, it exploits the syntactical structure of input questions as well as candidate formal queries to compute the similarities.
6. Senthilkumar, M., & Chowdhary, C. L. in 2019 [6] developed a sequence to sequence neural network model human to machine chatbot to replace the human. It is very productive compared to other normal chatbots.
7. Jungwook Rhim, et. al., in 2021 [7] surveyed different kinds of Chatbots and used three factors for surveying, such as respondents' perceptions of chatbots, interaction experience, and data quality. It also explained a different kind of Chatbot used in the environment.
8. Prissadang Suta, et. al., in 2021[8] developed a machine learning-based chatbot which handles three steps effectively, understanding the natural language input; generating an automatic, relevant response; and, constructing realistic and fluent natural language responses. The main issue of the natural language understanding problem is solved in this method.

3. RASA System Architecture

The following figure gives an overview of the Rasa open-source architecture. It is a readymade framework available in python for effective chatbot deveoopment and it is a machine learning framework. It has several components such as an action server, tracker store, lock store, file system, dialog policy, NLU pipeline, agent, and so on. The two main components are Natural Language Understanding (NLU) and Dialog Management. NLU is the part that handles intent classification, entity extraction, and response retrieval. It is shown below as the NLU pipeline because it uses the NLU model generated by the trained pipeline to handle the user's utterances.

The dialog management component determines the next action in the conversation, depending on the context.

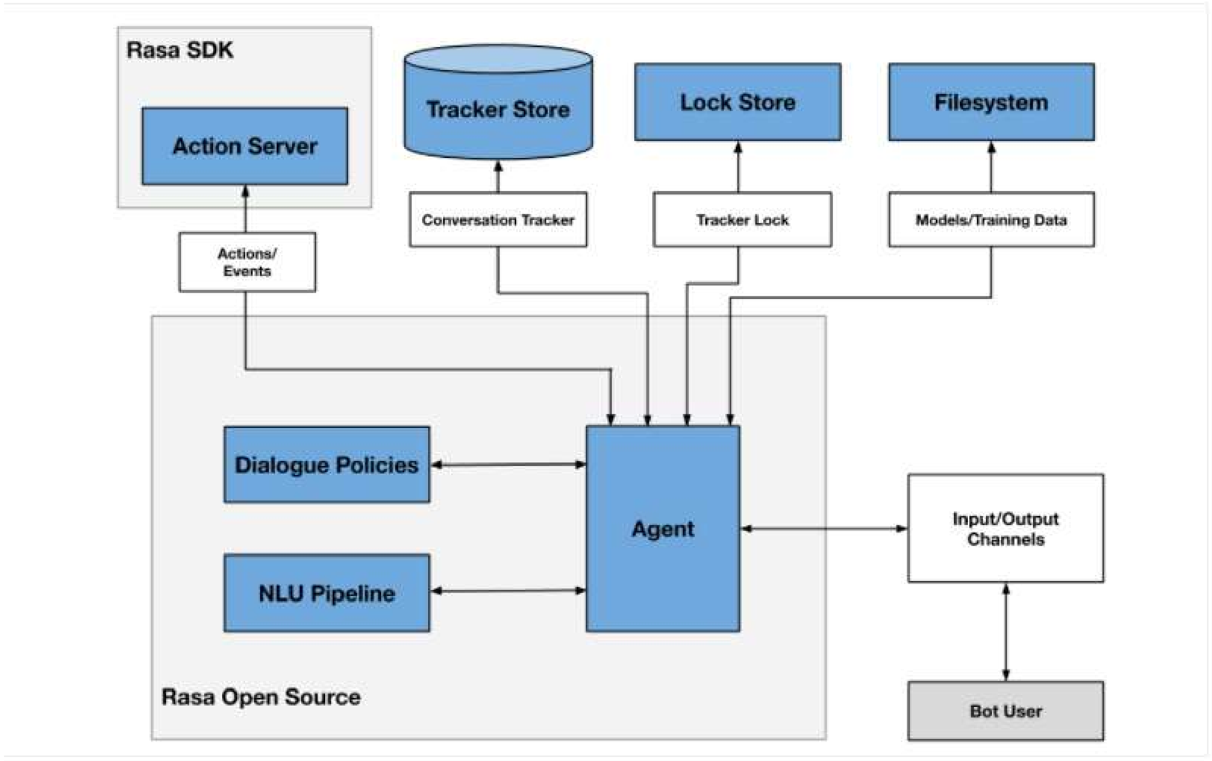


Figure 1. RASA Architecture

Action Server:

Custom actions in the rasa project run using this action server. When an action is predicted using rasa, then the rasa server sends a POST request in the form of JSON payload. When the server finishes, it returns the responses and events in the form of a JSON payload. A conversion tracker is created for the user's responses and adding the events. A webhook endpoint is accepted for HTTP POST requests in the case of using other action servers.

Track Store:

The conversations made in rasa are stored in a place called tracker stores. This open-source tool helps to create custom storage boxes. The default track store is In Memory Tracker Sore. Databases that can be used in rasa are PostgreSQL, Oracle, and SQLite. Mongo Tracker Store and Dynamo TrackerStore are also used to store the history of conversation in rasa.

Lock Store:

Lock store is used in rasa to ensure whether the incoming messages are processed in the correct order, then they are directly stored in lock stores for active processing. Multiple servers are used and no need for conversation sent by the client. The default lock store in rasa is In Memory LockStore. If the need for a persistence layer in conversation locks, RedisLockStore can be used.

File system:

Rasa collects and loads the training data and this data is imported using a custom importer. RasaFileImporters and MultiProjectImporter are used for importing the data. Then these data are stored in the File system.

Dialogue Policies:
Machine-learning and rule-based policies are used by rasa to decide which action to take place in a conversation. Actions can be selected on each configuration rasa takes, priority for each policy can also be given to make it higher or lower case. Machine learning policies are the TED policy, UnexpecTED Intent policy, Memoization policy, and Augmented Memoization policy. Rule-based policies are Rule policy and Configuring policies.

NLU Pipeline

The components of the NLU pipeline are Tokenizers, Features, Intent Classifiers, and Entity Extractors. Tokenizers will split the text into a text known as tokens. Featured is used for generating the numeric features in rasa. Two types of features like sparse and dense features are used. In rasa Intent Classification, the DIET algorithm is used. Entities don't need an algorithm to detect, so RegexEntityExtractor is used. Thus in NLU, actions are predicted, thus an NLU pipeline can be developed.

Agent

An agent in rasa will help us to train a model, load and will help us to use it. In simple words, it's a simple API that will help us to access Rasa's core functionality.

Bot User.

After the creation of input and output channels, that is when a complete rasa project is developed, a bot user will be created, and thus it helps us to solve the questions asked by humans.

Rasa Framework Layout: Basically, Rasa has the following built-in modules that are used when implementing Chabot:

1. RASANLU for understanding user messages. This part of the framework is a tool/library for intent classification and entity extraction from query text. Entities and intents also allow for the retrieval of responses and the composition of spoken text. You can use this component alone to create a simple and minimal AI chatbot yourself. This is usually the case when creating an AI chatbot that answers FAQs, simple queries, and so on. This blog also uses it to code chatbots.
2. Have a conversation with RASA Core and decide what to do next. This component is the framework's dialog engine that helps build more complex AI assistants that can handle

context (previous requests and responses in a conversation) during a response. RASA recommends using both NLU and Core, but these can be used separately.

3. Integration with Deployment Channels: These components allow you to connect and deploy your bot to popular messaging platforms. Learn more about this here. These help developers focus on bot functionality rather than the installation required for a real deployment. RASA X is a toolset that takes bots (developed with RASA open source) to the next level. It's free, but it's a closed toolset available to all developers. The following figure illustrates how the Rasa chatbot works.

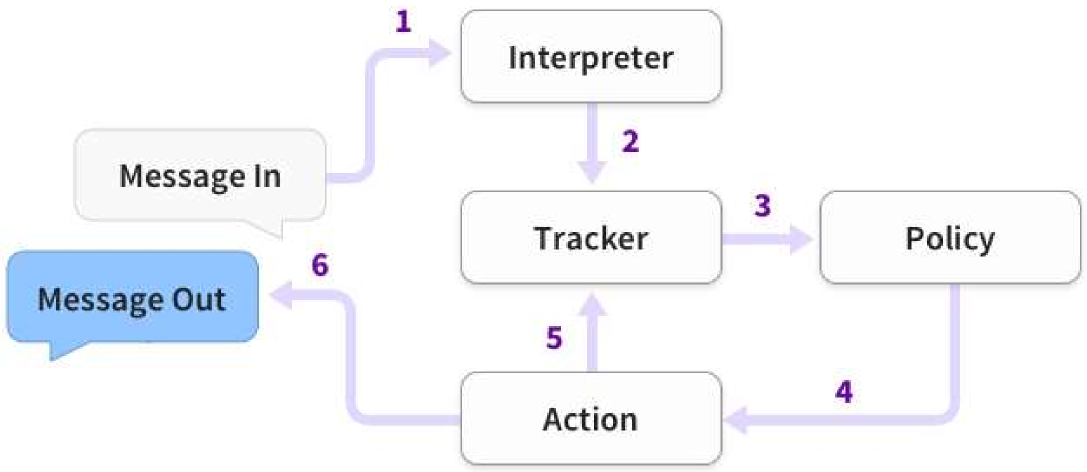


Figure 2 Method of working of Rasa Chatbot

The following steps are described below.

1. First, the message is received from the user in the bot and passed to the interpreter. The interpreter translates the message into a dictionary containing the original text, intents, and all found entities. This part is handled by NLU.
2. Next, the tracker is an object that tracks conversation status. You will receive information that a new message has arrived. It also helps you keep track of the entities extracted by the user and how the conversation is managed. NS. Dialogue flow.
3. The policy receives the current status of the tracker.
4. The policy determines the next action to take.
5. The selected action is logged by the tracker to help you follow the path or flow of the conversation.
6. The response is sent to the user. The user then responds to the response received from the bot.

4. Experimental

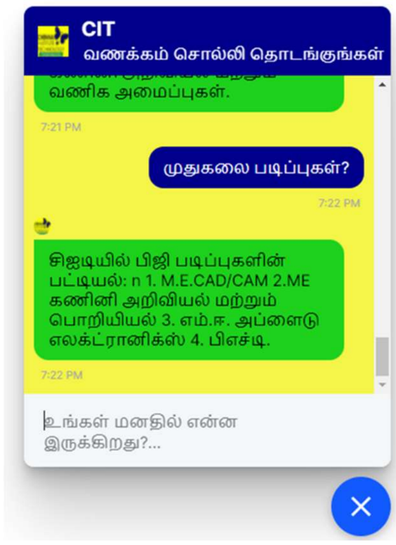
Intents are developed in Tamil to make the system customize the Tamil language. Responses were also created for the intents. Using these intents and responses Rasa trains data in the Tamil language. A set of stories are created which explain the situation sequence of activities to perform in the system. Rules are created to adapt the queries of different formats. In this system, nearly 46 intents and 140 responses are created in Tamil. Also, 20 stories and rules are created for intents and responses.

4.1 Experiment Setup

A python program is developed with an open-source machine learning framework is called the Rasa tool used for this system development. This uses python dependency packages like tensor flow, Spacy. It is configured to run through the GUI or Command prompt. In the experiment, the system is simulated using a PC environment connected to the Internet.

4.2 Simulation Screen

The Simulated system is configured to work for the Chennai Institute of Technology for admission seekers on the campus. This Chatbot system enables the user to chat with the system in Tamil and its screen snapshots are shown in the following figure 3.


CIT
வணக்கம் சொல்லி தொடங்குங்கள்
உங்கள் மனதில் என்ன இருக்கிறது?...

வணிக அமைப்புகள்.

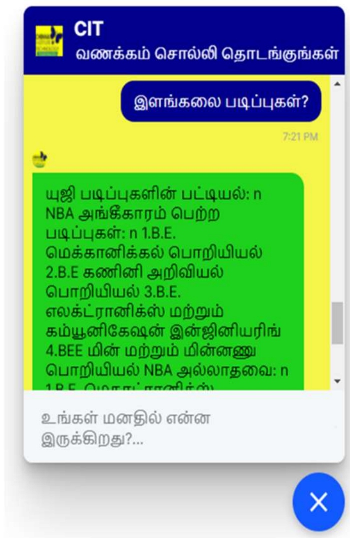
7:21 PM

முதுகலை படிப்புகள்?

7:22 PM

சிஐடியில் பிஜி படிப்புகளின் பட்டியல்: n 1. M.E.CAD/CAM 2.ME கணினி அறிவியல் மற்றும் பொறியியல் 3. எம்.ஈ. அப்ளைடு எலக்ட்ரானிக்ஸ் 4. பிஎச்டி.

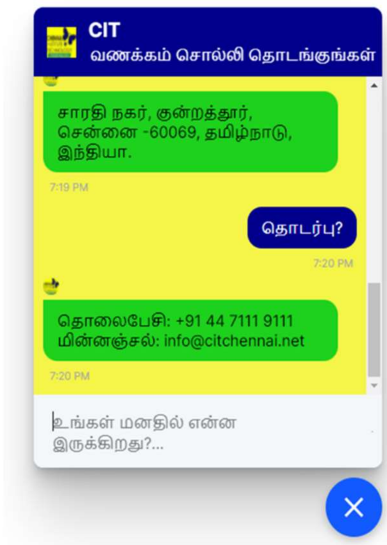
7:22 PM


CIT
வணக்கம் சொல்லி தொடங்குங்கள்
உங்கள் மனதில் என்ன இருக்கிறது?...

இளங்கலை படிப்புகள்?

7:21 PM

யுஜி படிப்புகளின் பட்டியல்: n NBA அங்கீகாரம் பெற்ற படிப்புகள்: n 1.B.E. மெக்கானிக்கல் பொறியியல் 2.B.E கணினி அறிவியல் பொறியியல் 3.B.E. எலக்ட்ரானிக்ஸ் மற்றும் கம்யூனிகேஷன் இன்ஜினியரிங் 4.BEE மின் மற்றும் மின்னணு பொறியியல் NBA அல்லாதவை: n


CIT
வணக்கம் சொல்லி தொடங்குங்கள்
உங்கள் மனதில் என்ன இருக்கிறது?...

சாரதி நகர், குன்றத்தூர், சென்னை -60069, தமிழ்நாடு, இந்தியா.

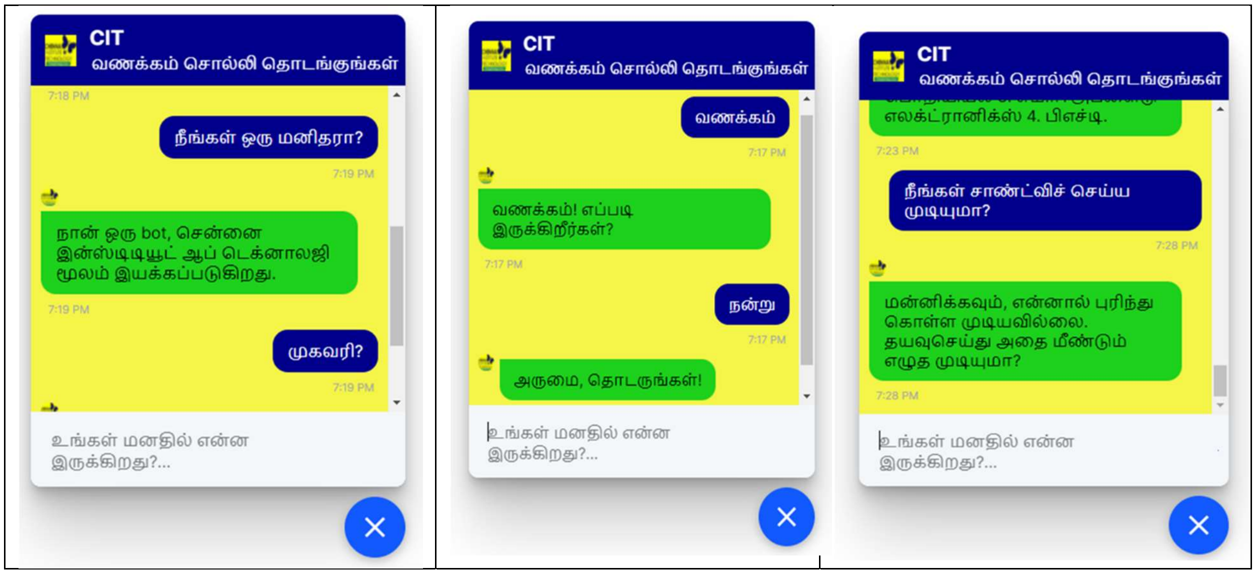
7:19 PM

தொடர்பு?

7:20 PM

தொலைபேசி: +91 44 7111 9111 மின்னஞ்சல்: info@citchennai.net

7:20 PM



CIT
வணக்கம் சொல்லி தொடங்குங்கள்

7:18 PM

நீங்கள் ஒரு மனிதரா?

7:19 PM

நான் ஒரு bot, சென்னை இன்ஸ்டிடியூட் ஆப் டெக்னாலஜி மூலம் இயக்கப்படுகிறது.

7:19 PM

முகவரி?

7:19 PM

உங்கள் மனதில் என்ன இருக்கிறது?...

CIT
வணக்கம் சொல்லி தொடங்குங்கள்

வணக்கம்

7:17 PM

வணக்கம்! எப்படி இருக்கிறீர்கள்?

7:17 PM

நன்று

7:17 PM

அருமை, தொடருங்கள்!

உங்கள் மனதில் என்ன இருக்கிறது?...

CIT
வணக்கம் சொல்லி தொடங்குங்கள்

எலக்ட்ரானிக்ஸ் 4. பிஎச்டி.

7:23 PM

நீங்கள் சாண்ட்விச் செய்ய முடியுமா?

7:28 PM

மன்னிக்கவும், என்னால் புரிந்து கொள்ள முடியவில்லை. தயவுசெய்து அதை மீண்டும் எழுத முடியுமா?

7:28 PM

உங்கள் மனதில் என்ன இருக்கிறது?...

Figure 3 Chatbot screen snapshots

5. Experimental Results and Discussion

In general, there is different kinds of measurements [10] are used for measuring the value of the Chatbot systems [9]. These attributes are 1. User Satisfaction 2. Dialog Efficiency Metric and 3. Several Queries and Response, etc. In this system, explicit feedback is collected from the user to get their user satisfaction level and satisfaction level is plotted as a bar chart shown in Figure 4. Secondly, the efficiency is measured of 8 sample dialogues in terms of the atomic match, First-word match, Most significant match, and No match. To measure the efficiency of the adopted learning mechanisms to see if they increase the ability to find answers to general user input as shown in Figure 5.

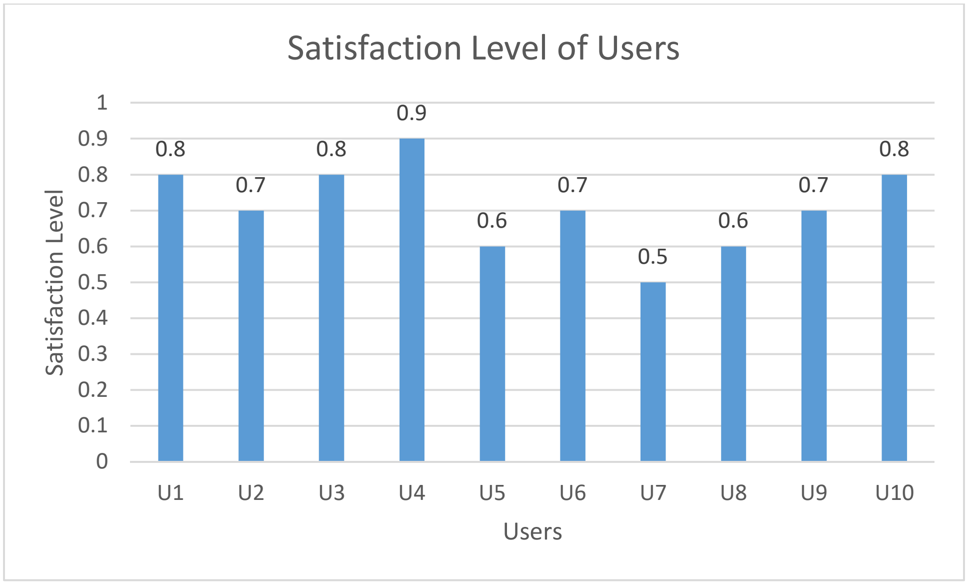


Figure 4 – User Satisfaction Level Measures.

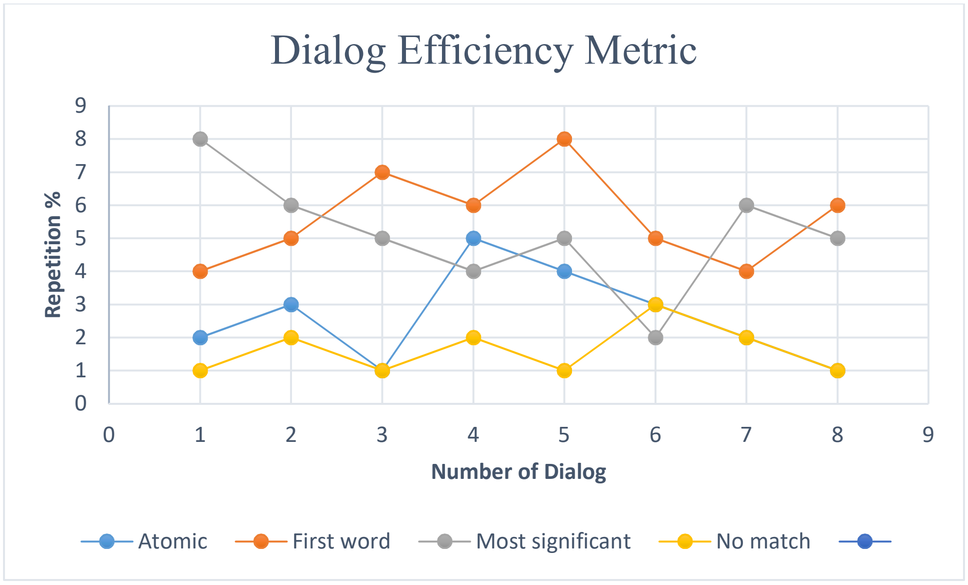


Figure 5– Dialog Efficiency Relative Frequency

6. **Conclusion**

Tamil Chatbot is one of the important systems that enables users to communicate with the system and enable to get more information about the system. In this work, a Chatbot is developed for Chennai Institute of Technology (Higher Education Institute) for providing various information about the Institute. The system is configured with Rasa Framework which is a conversational AI system. It uses the LSTM Architecture for sequencing the dialogue. The performance of the system is also relatively good compared to a regular system.

7. **References**

1. T. Kalaiyarasi, R. Parthasarathi, T.V.Geetha, (2003) Poongkuzhali-An Intelligent Tamil Chatterbot, Proceedings of Tamil Internet Conference 2003, 86 – 95.
2. Meyer von Wolff, R., Hobert, S., & Schumann, M. (2019). How may I help you? – State of the art and open research questions for chatbots at the digital workplace. Proceedings of the 52nd Hawaii International Conference on System Sciences. https://doi.org/10.24251/hicss.2019.013
3. Galitsky, B., & Goldberg, S. (2019). Explainable machine learning for chatbots. Developing Enterprise Chatbots, 53-83. https://doi.org/10.1007/978-3-030-04299-8_3.
4. Sorokin, D., & Gurevych, I. (2018). Mixing context Granularities for improved entity linking on question answering data across entity categories. Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics. https://doi.org/10.18653/v1/s18-2007.
5. Zafar, H., Napolitano, G., & Lehmann, J. (2019). Deep query ranking for question answering over knowledge bases. Machine Learning and Knowledge Discovery in Databases, 635-638. https://doi.org/10.1007/978-3-030-10997-4_41.
6. Senthilkumar, M., & Chowdhary, C. L. (2019). An AI-based chatbot using deep learning. Intelligent Systems, 231-242. https://doi.org/10.1201/9780429265020-12.
7. Rhim, J., Kwak, M., Gong, Y., & Gweon, G. (2022). Application of humanization to survey chatbots: Change in chatbot perception, interaction experience, and survey data quality. Computers in Human Behavior, 126, 107034. https://doi.org/10.1016/j.chb.2021.107034.

8. Shawar, B. A., & Atwell, E. (2007). Different measurements metrics to evaluate a chatbot system. Proceedings of the Workshop on Bridging the Gap Academic and Industrial Research in Dialog Technologies - NAACL-HLT '07. https://doi.org/10.3115/1556328.1556341.
9. https://www.interactions.com/conversational-ai/
10. Shawar, B. A., & Atwell, E. (2007). Different measurements metrics to evaluate a chatbot system. Proceedings of the Workshop on Bridging the Gap Academic and Industrial Research in Dialog Technologies - NAACL-HLT '07. https://doi.org/10.3115/1556328.1556341