Ph.D. Thesis

A HYBRID KNOWLEDGE BASE CHATBOT FRAMEWORK WITH ENHANCED INQUISITIVENESS AND LANGUAGE LOCALISATION

Submitted to

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

in partial fulfilment of the requirements for the award of the degree of

Doctor of Philosophy

by

RESHMI S
(Reg. No. 3516)

Under the guidance of

Dr. B. KANNAN

DEPARTMENT OF COMPUTER APPLICATIONS
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI- 682 022, INDIA

FEBRUARY 2019

A HYBRID KNOWLEDGE BASE CHATBOT FRAMEWORK WITH ENHANCED INQUISITIVENESS AND LANGUAGE LOCALISATION

Ph.D. Thesis

Author

Reshmi S.
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India
Email: reshmis@gmail.com

Supervising Guide

Dr. B. Kannan
Professor & Head
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022.
Email: mullayilkannan@gmail.com

February 2019

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER APPLICATIONS

Thrikkakara, Kochi - 682 022

CERTIFICATE

Certified that the work presented in this thesis entitled ***A Hybrid Knowledge Base Chatbot Framework with Enhanced Inquisitiveness and Language Localisation*** is based on the authentic record of research carried out by Mrs. Reshmi S. under my guidance in the Department of Computer Applications, Cochin University of Science and Technology, Kochi - 22 and has not been included in any other thesis submitted for the award of any degree. Also certified that thesis is adequate and complete for the award of the Ph.D. Degree.

Kochi - 22
12-02-2019

Prof. (Dr.) B. KANNAN
Supervising Guide

DECLARATION

I hereby declare that the work presented in this thesis entitled ***A Hybrid Knowledge Base Chatbot Framework with Enhanced Inquisitiveness and Language Localisation*** is based on the original research work carried out by me under the supervision and guidance of Dr. B. Kannan, Professor, Department of Computer Applications, Cochin University of Science and Technology, Kochi - 22 and has not been included in any other thesis submitted previously for the award of any degree.

Kochi - 22 **RESHMI S**

12-02-2019

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER APPLICATIONS
Thrikkakara, Kochi - 682 022

CERTIFICATE

Certified that the work presented in this thesis entitled ***A Hybrid Knowledge Base Chatbot Framework with Enhanced Inquisitiveness and Language Localisation*** submitted to Cochin University of Science and Technology by Mrs. Reshmi S. for the award of degree of Doctor of Philosophy under the Faculty of Technology, contains all the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar and recommended by the Doctoral Committee.

Kochi - 22
12-02-2019

Prof. (Dr.) B. Kannan
Supervising Guide

Acknowledgements

I would like to express my deepest sense of gratitude to my research guide, **Prof. (Dr.) B. Kannan**, Head, Department of Computer Applications, Cochin University of Science and Technology, for his excellent guidance and incessant encouragement. It has been a great pleasure and privilege to work under him and he was always there when I needed help.

I am greatly thankful to **Prof. (Dr.) K. V. Pramod** for his generous support and inspiration throughout my research. I extend my sincere gratitude to **Dr. M. Jathavedan**, **Dr. A. Sreekumar**, **Ms. Malathi S.**, **Dr. M. K. Sabu** and **Dr. M. V. Judy**, faculty members of the Department of Computer Applications for their solid support and necessary corrections to accomplish my research goals.

The endless support, affection and timely help extended by all the office staff, technical staff and librarian in the Department of Computer Applications are remembered with great sense of gratitude.

I take this opportunity to express my sincere thanks to Prof. (Dr.) Supriya M. H., Department of Electronics, Cochin University of Science and Technology, for the whole hearted support and constant encouragement. My heartfelt gratitude goes to Dr. Mohankumar K., with whom I had many insightful discussions, which have improved my understanding of various topics.

The constant support, sincere love and timely help extended by all

my co-researchers - Vijith T. K., Vinu V. S., Sukrith B., Jino P. J., Aneurin Salim A. L., Sariga Raj, Sunil Kumar R., Rajesh Kumar R., Soumya George, Sruthi S., Simily Joseph, Jomy John, Soumya T. V., Sruthi K. S., Rekha K. S., Ajlisa O. A., Haritha K., Shernasmol A. A. and Divya Sindhulekha are remembered with much pleasure and token of gratitude. A word of mention is deserved by my friends Dr. Subhash Chandra Bose M. R., Gigin Sunilkumar, Ravi Kumar, Sumi H., Divia P., Ranjith K. V., Shalini K. S. and Limna T. J. who were a constant source of motivation and energy for me.

I thankfully acknowledge the sincere co-operation and support I received from the staff and management of Grey Technolab India Pvt. Ltd., Cochin and HCL Technologies Ltd., Bengaluru.

It is beyond words to express my gratitude to my siblings, Mrs. Reshma S. and Major Harish S., and loving parents for their prayers and blessings. I am deeply indebted to my uncle Mr. Radhakrishnan P., my aunt Mrs. Jeeja I. and their two daughters, who gave me support in the form of deep care and motivation. I thank my father in law, Mr. Velayudhan K. K and mother in law Mrs. Satheerathnam E. K for their support.

I owe my loving thanks to my husband Rajesh K. V and my little one, Smera Raj for her understanding, full fledged support, sincere love, patience and inspiring words. Beyond all, I express my deep gratefulness to Almighty, who gave the health and courage to make this dream come true.

RESHMI S

ABSTRACT

One of the key research areas in the field of human-computer interaction is the design of natural and intuitive interaction modalities. In particular, many efforts have been devoted to the development of systems which are capable of interacting with the user in their natural language. Chatbots are the classical interfaces for natural language interaction. Chatbot is a computer program designed to simulate an intelligent conversation with one or more human users in natural language and attempts to hold a conversation in a way that imitates a real person. Chatbots generally communicate with their human partners through simple text interfaces, although some include speech recognition and text-to-speech features. Chatbots have gained considerable research interest due to its infotainment as well as commercial importance.

The proposed work envisages the implementation of a chatbot framework that aims to enhance the inquisitive capabilities of chatbots by intelligently identifying and collecting missing data from the users which are needed for the response generation. This research work mainly focuses on chatbots that can cater to the needs of small to medium businesses. Such systems are expected to be affordable, easily configurable and maintainable while being sufficiently interactive. This chatbot system uses a hybrid knowledge base that includes a modified AIML system, RDBMS along with an Integrated Big data framework as an external knowledge base. The potentials of a language localised chatbot in a regional language has also been investigated.

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
AIML	Artificial Intelligence Markup Language
ALICE	Artificial Linguistic Internet Computer Entity
ASE	Artificial Support Entity
CAT	Conversational Analysis Theory
CRF	Conditional Random Field
DL	Description Logic
GQM	Goal Question Metric
HDFS	Hadoop Distributed File System
HDP	Hortonworks Data Platform
HMM	Hidden Markov Model
HQL	Hive Query Language
HS2	HiveServer2
KB	Knowledge Base
LSTM	Long Short Term Memory
MEMM	Maximum Entropy Markov Model
NER	Named Entity Recognition
NLP	Natural Language Processing
NYSE	New York Stock Exchange
OWL	Web Ontology Language
PARADISE	PARAdigm for Dialogue System Evaluation
POS	Part-of-Speech
RDBMS	Relational Data Base Management System
RDF	Resource Description Framework
SVM	Support Vector Machine
YARN	Yet Another Resource Negotiator

Chapter 1

Introduction

The design of natural and intuitive interaction systems has always been one of the most important areas of research in the field of human-computer interaction. In particular, many efforts have been devoted to the development of systems which are capable of interacting with the user in their natural language and one of the classical interfaces for such natural language interaction is chatbots. In recent years, chatbots have gained considerable research interest due to its infotainment as well as commercial importance and are being widely employed across industries for various purposes.

Chatbot also known as chatterbot or conversational agent is a computer program designed to simulate and carry forward an intelligent conversation with one or more human users without any time or location related barriers. They are software frameworks that can respond to natural language input and it converse in such a way that it imitates a real person. Chatbots communicate with their human partners through simple text interfaces, although some include

speech recognition and text-to-speech features.

Chatbots try to understand what the user is saying, analyse it and attempt to provide a suitable response. Some chatbots strive to be indistinguishable from humans, while others try and stand out from humans with super-human knowledge or features. While some chatbots simply look for keywords, phrases and patterns that have been programmed into their knowledge base [1], some others use more advanced techniques like natural language processing [2]. Apart from the active research activities in this field, as of yet, no chatbot has been able to ultimately make the users believe that it is one of them through its knowledge of the natural language and the way of interaction. Many research works are being carried out to improve the conversational capabilities to make chatbots converse more sensibly and naturally. This chapter gives an overview of chatbots which is the area of focus of this thesis. This chapter also introduces the nature and significance of the problem domain, motivation behind the research and various contributions.

1.1 Overview of Chatbots

Artificial Intelligence (AI) technology provides techniques for developing computer programs for carrying out a variety of tasks, simulating the intelligent way of problem-solving by humans. Chatbots are considered to be a serious branch in AI research, where they act as a media to provide information to the users as well as assist them in fulfilling their tasks by utilizing the power of AI techniques. The power of such systems are yet to be assessed entirely and applied to the maximum possible extent.

Natural language text is the input for almost all dialogue agents and they respond with the best possible answer for user queries, by text or speech. This process continues until the conversation winds up [3]. Fig. 1.1 depicts the flow of the conversational process between humans and chatbots. One of the essential features of a chatbot is the ability to carry out uninterrupted communication with the user in the form of either text or speech [4]. This can be achieved only if the chatbot can analyse the user query and provide a suitable response.

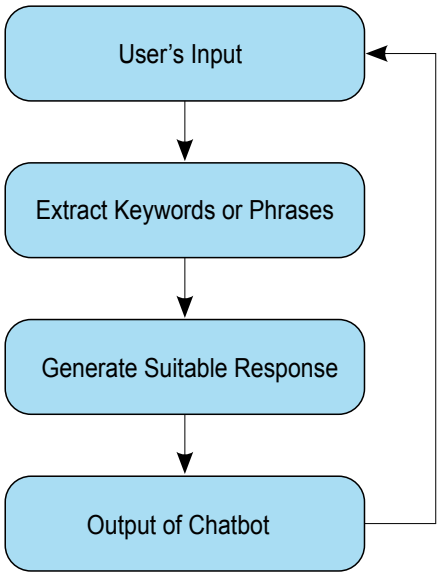


Figure 1.1: Conversational process flow between humans and chatbots

Chatbots mainly consists of three parts: a knowledge base, an interpreter program and chat engine [5]. The knowledge base encapsulates the intelligence of the system and is composed of keywords/phrases and responses associated with each keyword/phrase. The common implementation of knowledge base involves the use of dat files or text files, databases and XML files or even a combination of these [6].

The interpreter program facilitates the communication process with the user and includes two submodules namely analyser and generator. The analyser reads the input text from a human partner and analyses the syntax and semantics of the sentence. It acts as a pre-processor to the user input and uses different normalisation techniques like pattern fitting, substitution and sentence splitting [7].

The chat engine tries to match the pre-processed output of the analyser using various techniques like pattern matching, sentence reconstruction, indexing, etc. with the data contained in the knowledge base and identifies the suitable response. Generator processes the response given by the chat engine and generates appropriate grammatically correct sentence to display. Fig. 1.2 illustrates the typical components of a chatbot and the relation between these components.

1.1.1 Scope and Application of Chatbots

Chatbots have evolved considerably from their earlier generations where they were used as simple scripted answering machines. Currently, chatbots are efficient enough to carry out conversations with multiple users without any human support or intervention. Chatbots are becoming more popular across many industries, especially in those industries where human-computer interactions take place, like banking, insurance, health care, education, e-commerce, etc. [8].

Conventionally, the task of user/customer interaction is carried out by human experts through telephonic calls or direct meetings. The operator who is assigned to assist the user is expected to possess the necessary skill to identify the issues and provide solutions or

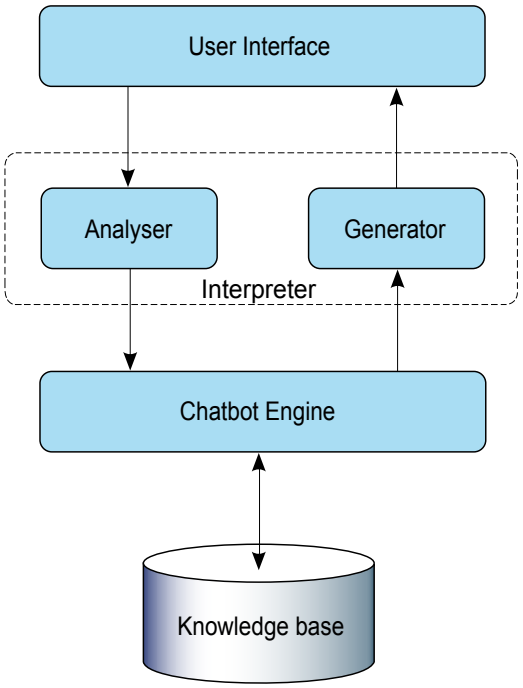


Figure 1.2: General chatbot architecture

information to the user. However, as the number of users handled by the operator increases, the task of taking care of each customer becomes very difficult. There will be chances of misinterpreting the query and the operator may return a negative/incorrect response. Such systems demand extreme alertness for a prolonged period from the operator and will result in fatigue and boredom. The monotonous nature of the task results in the decline of performance and quality of work. Due to these confounding factors, there is a growing need for an automatic query responder in different domains for user interactions.

Presently, most of the business fraternities have started employing chatbots for answering customer queries so that they can assign existing manpower to other tasks, which will, in turn, result in

improved productivity and will help in reducing the cost. Chatbots can be employed in different levels of business processes, starting from customer service to lead generation to lead fulfillment. They can act as an artificial tutor, an agent for selling products, help desk in large libraries or institutions, for assisting in booking hotels, schedule meetings, paying bills, finding places to visit, or to automate daily tasks like ordering groceries, etc. [6, 9, 10].

According to a survey conducted by Oracle, in 2016 [11], 80% of the major companies around the world have confirmed that they are either using chatbot services or have plans to use chatbots for customer interactions by 2020. It is also found that business fraternities who have employed chatbots for consumer-facing products have reported increased sales.

1.2 Motivation of the Work

Chatbots are one of the upcoming technologies which would reduce the operational costs associated with manpower in areas where man to man interaction takes place. Chatbots have opened an arena for business and various industries to provide an unprecedented experience to the customers through exceptional customer support. For any business to be successful, they have to be customer-centric. Apart from the proven fact that chatbots can reduce the cost of operation and can provide uninterrupted service throughout the day, resulting in more visits/sales. Chatbots offer an interface to the customer that will help them to avoid repeat visits and can get their requirements addressed in lower costs. This makes them more satisfied and in turn results in more business for the service provider.

Smaller businesses are reluctant to engage themselves in continued customer service, because of concerns regarding elevated cost. In such situations, chatbots can provide an efficient and easily deployable solution, that can help to reduce costs and resources. This is valuable as it enables the businesses to showcase their capabilities by sharing the details of their products and services. This work mainly focuses on chatbots that can cater the needs of such small to medium businesses, which needs to be cost effective and user friendly.

Communication in local language is an important factor that can ensure user engagement. This work also focuses on a language localised chatbot in Malayalam. Malayalam is one among the twenty-two scheduled languages of India with the official language status in the State of Kerala and Union Territories of Lakshadweep and Puducherry. It is a classical language with a rich literary tradition and is spoken by 33 million people. Also, we have observed the unavailability of a chatbot that can communicates in Malayalam. Therefore, a chatbot which can communicate in Malayalam will be beneficial to Malayalam speaking people who don't have expertise in English.

1.3 Problem Definition

Response generation is indisputably the most vital process of a chatbot system. Response generation task is mainly dependent on the knowledge base of the system. Once the query is received from the user, the chatbot, with the help of knowledge base, generates the response message that has to be delivered to the user. Depending on the response generating mechanism, two models namely, retrieval based model and generative based model are in use. While the retrieval based

models generate the response simply by accessing a predefined knowledge base, the generative based models create new sentences depending on the context for response generation.

The retrieval based model extracts most probable Query-Response (Q-R) pair from a set of available Q-R pairs stored in the knowledge base which is matching with the current user query Q and provide the response R from the selected Q-R pair [12]. Generative based methods employ an encoder-decoder framework that initially encodes Q as a vector representation and feeds the same to the decoder to produce response R. This approach depends upon the existing Q-R pairs as training data, which is, in fact, similar to the knowledge base of retrieval based models [13].

As the retrieval based models generally fetch the responses by considering matching features in the query with those stored in the knowledge base, the design of a general purpose chatbot demands very extensive knowledge base construction of which is often a difficult task. Thus they are generally employed for closed domain dialogue systems which are very limited to specific applications [2]. However, even for closed domains, collecting query response pairs for different specific areas could be intractable. In the case of generative models, the learning/training phase also demands sufficiently large data sets to achieve acceptable levels of performance.

Thus, the success of a chatbot depends highly on the robustness of the knowledge base. It can be generally said that a chatbot is as good as its knowledge base [14]. As it is required to have a very large knowledge base and subjectivity of response, developing a perfect chatbot is very difficult. Different technologies and formats are

available to frame knowledge base for chatbots which include manual writing, learning from corpus [15] etc.

As mentioned before, both the models retrieve the information from the user's query and try to provide an appropriate response with the help of the knowledge base. However, in many scenarios even if the knowledge base is robust, the information specified in user query may not be sufficient enough to generate the response. Conventionally this is handled very generically in chatbots, especially in rule-based chatbots. In such contexts, chatbots respond with some insipidity response stored in the system like "Can't find an answer", "I do not understand" etc. [16, 17]. This situation is not handled properly by existing systems and may lead to the abrupt end of the communication flow. The existing systems lack the intelligence to identify the missing information required for answering the query and would not be capable enough to probe the user further to collect the missing information. This is a significant aspect of chatbot implementation.

Retrieval-based model stores a knowledge base of predefined responses and employs a heuristic method to select an appropriate response. One of the most noticeable drawbacks of a retrieval based method is the lack of customized response generation as the responses are pre-existing and not dynamic. Thus, responses may seem to be artificial. In the case of generative models, automated language generation tasks like machine translation and paraphrasing often become a bottleneck [18]. It can also be noted that the machine-generated text often lacks fluency and naturality. Another major drawback with such pre-canned response generation is that it cannot provide the responses containing frequently changing data without knowledge updation. When there is a change in the data that

is required for the response generation, the corresponding change should also be reflected in the knowledge base. Manually updating such changes frequently in the knowledge base would turn out to be a time consuming and tedious task [19–21].

Most of the research works pertaining to chatbots are based on English. Even though there are some works based on Chinese [22], German [23] and Spanish [24] languages, they are comparatively very few. The main reason for not having the focus on non-English languages is the absence of quality corpus and data processing tools. Development of such systems that can interact with the users in the native language is very important, which can lead to seamless and more effective human-computer interactions.

1.4 Thesis Objectives

The primary objective of this research is the design of a chatbot framework that aims to enhance the inquisitive capabilities of chatbots by intelligently identifying and collecting missing data from the users query which are needed for response generation. This research work mainly focuses on chatbots that can cater the needs of small to medium businesses. Such systems are expected to be affordable, easily configurable and maintainable while being sufficiently interactive. Another objective is the design of the language localised chatbot which can converse in Malayalam and its knowledge base composition.

To achieve this, the current research work focuses on the following goals,

- Design and implement an inquisitive chatbot framework capable

of analysing the query and intelligently identify the missing information. Collect the missing information by probing the user and generate the response based on the collected details

- Improve the inquisitiveness feature of chatbot by using natural language processing techniques and ontology
- Integration of database and big data framework to enable the generation of dynamic responses
- Developing a chatbot that can communicate in Malayalam language and the implementation of Malayalam knowledge base generation tool

1.5 Thesis Contributions

- Design of new inquisitive chatbot that accepts a query from users, finds out missing information in queries and probe the user to provide missing information
- Enhanced the inquisitiveness of chatbot by incorporating Named Entity Recognition (NER) technique
- Introduced an ontology-based chatbot for addressing multilevel inquisitiveness
- Integration of multiple knowledge bases to the chatbot for generating dynamic responses and dynamic data fetch from different data sources
- Empowered the chatbot with mass knowledge analysis capability from a distributed environment through big data integration
- First exclusive work on a language localised chatbot in Malayalam
- A novel attempt is made to create a knowledge base synthesiser for the Malayalam language

1.6 Thesis Outline

In this chapter, we presented an overview of chatbots, with its scope and application in different domains along with the research objective and motivation behind this work. The remaining chapters are organized as follows:

Chapter 2: Background and Review of Literature covers various research works reported in literature in the areas of chatbot implementation and background study of related topics. The chapter also discusses a brief history of chatbots and detailed architecture of ALICE chatbot framework along with an overview of the AIML knowledge base. Different evaluation methodologies for evaluating the chatbot's performance have also been included in this chapter.

Chapter 3: Inquisitive Chatbot Framework discusses the methodology adopted for the realization of the proposed inquisitive chatbot framework. This chapter details the implementation of the inquisitive chatbot which finds the missing data in the query and probes the users to collect the data that is required to provide appropriate response. The second section of this chapter addresses the methodology adopted for enhancing the inquisitiveness and interaction of the chatbot by identifying entities through Named Entity Recognition.

Chapter 4: Ontology based Inquisitive Chatbot discusses various steps involved in the implementation of an multilevel inquisitive chatbot using ontology. This implementation can find multilevel missing values and can probe the user for additional data when the input is ambiguous or insufficient.

Chapter 5: Hybrid Knowledge based Chatbot discusses the integration of big data as knowledge base into the chatbot framework. An integrated chatbot framework with hybrid knowledge base is described towards the end of this chapter.

Chapter 6: Regional Language Chatbot envisages the need and relevance of a regional language chatbot. A retrieval based closed domain chatbot which can converse in Malayalam with the user has been proposed along with its development phases. This chapter also presents a Malayalam knowledge base synthesiser that facilitates the Malayalam knowledge base generation.

Chapter 7: Conclusions and Future Works recapitulate the thesis, points out the salient highlights of the work and the inferences gathered along with the scope and direction for future research work in this area.

Chapter 2

Background and Review of Literature

2.1 Introduction

A comprehensive review of literature related to the area of research is presented in this chapter. Various research works reported in the open literature in the areas of chatbot design, architecture, implementation, etc. have been covered. This chapter also discusses the evolution of chatbots along with in-depth coverage of various chatbots in different domains with their applications. A comprehensive study of the architecture of ALICE chatbot framework, which serves as the platform for the works presented in this thesis, along with the overview of Artificial Intelligence Mark-up Language (AIML) is also presented. Different evaluation methodologies for evaluating the chatbot's performance have also been consolidated and presented.

2.2 Evolution of Chatbots

One of the first established chatbots, ELIZA was created in 1966 by Joseph Weizenbaum at Massachusetts Institute of Technology (MIT) and emulated a psychotherapist [25]. It was inspired by the ideas of Turing [26], who argued that it was possible to build machines which are capable of acting like humans. The ELIZA chatbot accepted inputs from the users and employed some basic pattern matching techniques to select a possible response from a stored list. In case if a matching response is not found, ELIZA used some static phrases to continue the conversation. The program proved to be amazingly efficient in sustaining people's attention during the conversation and the success of the original program has influenced the development of many other chatbots. It can be noted that the principles used in ELIZA have laid the foundation for defining the basic theoretical framework of chatbots with specific phrases, keywords and preprogrammed responses.

The next noticeable chatbot was Parry, developed by Kenneth Colby in 1972 at Stanford University. In contrast to ELIZA, Parry simulated a patient suffering from Schizophrenia rather than a psychotherapist [27]. Parry was envisioned to help in the study of the nature of paranoia and was attuned to express different emotions like fears, beliefs and anxieties [28, 29]. Parry worked through a complicated system of attributions, assumptions and emotional responses triggered by changing weights assigned to verbal inputs. Parry's knowledge was defined using condition-action production rules, which an output response associated with a pattern.

Jabberwacky was another significant chatbot developed by Rollo

Carpenter in 1988. It was designed in such a way that it makes the conversation entertainable by simulating a natural human conversation [30]. It kept track of each user inputs and tried to provide an appropriate response employing contextual pattern matching techniques [31].

Following the footsteps of Jabberwacky, Dr. Sbaitso chatbot was created in the year 1992 for MS-DOS based machines [32]. Dr. Sbaitso was designed to showcase voice operated chatbot and personified the role of a psychologist.

ALICE (Artificial Linguistic Internet Computer Entity) [33] was the most prominent chatbot of 20^{th} century. Its ability to process natural language made ALICE exceptional and left the impression to the users that they are chatting with a real human. It personified a young lady and could carry out dialogues with users to provide information regarding her age, hobbies and other interesting facts. In order to converse with the users, heuristic pattern matching rules were applied by ALICE to the user input queries. A unique characteristic of ALICE was that it used AIML (Artificial Intelligence Mark-up Language) [34], an XML dialect developed by Dr. Richard Wallace in 1995, for its knowledge base implementation. A detailed study of ALICE is portrayed in section 2.5.

In 2006, IBM introduced, Watson, a query answering system which uses natural language processing and machine learning techniques [35]. Over the next decade, chatbots started gaining much more attention and many speech based intelligent personal assistant application came from big tech companies, starting with Apple's Siri [36] in 2010, Google Now in 2012, Amazon's Alexa [37] in 2015 and Microsoft's Cortana in 2015

[38]. However implementation like Alexa, Watson, Cortana, etc. require huge data sets for learning and also require sophisticated computational resources. These may not always be practically feasible for small-scale chatbot implementations.

Another AIML based chatbot namely Mitsuku was developed by Steve Worswick in 2012. Mitsuku is the four-time winner of Loebner prize in the years 2013, 2016, 2017 and 2018. The differentiating characteristic of this chatbot is its ability to reason with specific objects by checking its properties. Based on the rules and features configured in it, Mitsuku can learn herself and respond to new topics which have not already learned [39].

Microsoft Corporation launched Tay in March 2016; it was a machine learning Twitter chatbot designed to learn from interactions with Twitter users. However, due to controversy tweets posted by the bot, Microsoft shutdown the service within hours after the launch [40]. Later it was replaced with ZO [41].

From the last few years, chatbots have gained the interest of the general masses. In the upcoming years, it is expected that chatbots will be an integral part of our lives, in different facets of our day to day activities. Chatbots will be able to analyze information and take decisions in an effective and efficient manner which will help the users to find solutions to their queries.

Table 2.1 gives a summary of the chatbots discussed so far which have attracted the attention of the research community.

Table 2.1: Some notable chatbots

Chatbot	Year	Description	Approach
Eliza	1966	Simulate psychotherapist. Rephrasing response with few grammar rules.	Pattern matching and substitution.
Parry	1972	Simulate a patient with paranoid schizophrenia.	System of assumptions, attributions.
Jabberwacky	1988	Simulate natural human chat in an interesting, entertaining and humorous manner.	Contextual pattern matching, learns by association, learn new responses based on user interactions.
ALICE	1995	Applies heuristic pattern matching rules to input.	AIML
Watson	2006	Question answering system uses NLP and machine learning to reveal insight from large amount of data.	IBM's DeepQA and Apache UIMA.
Siri	2010	Work as an intelligent personal assistant. Part of Apple's OS.	Java, C, NLP
Mitsuku	2012	Claims to be an 18 years old female chatbot from Leeds. Four time Loebner prize winner in 2013, 2016, 2017, 2018.	AIML
Tay	2016	Chatbot that was originally released by Microsoft corporation via Twitter which caused controversy on Twitter and was taken offline shortly after.	Python, Java

2.3 Approaches for Chatbot Enhancement

2.3.1 AIML based Chatbots

This thesis mainly concentrates on AIML based chatbots which is suitable for small to medium business requirements. AIML is one of the popular ways of implementing rule-based chatbots and is flexible and easy to configure. Many variants of AIML chatbot has been reported in the open literature.

The development of a generic architecture for the creation of chatbots with personality using AIML has been discussed in [4]. The authors propose a flexible architecture that allows the creation of different and coherent personality models for the integration of personality in AIML chatbots. The Persona-AIML architecture contains four components: Categories Base, Personality Component, Dialogue Log and Reasoning Component. With these components, it is possible to describe attitudes, emotions, mood, physical states and traits.

Graphical Artificial Markup Language, an extension of AIML allowing the merge of verbal and graphical interaction modalities has been presented in [42]. The language can define personalized interface patterns that are suitable for the type of data exchanged between the user and the system during the process of conversation. A chatbot system namely Graphbot is also presented which can support this language and to interact with users in a mixed mode. The Graphbot system has been designed according to the well-known client-server three-tier architecture and the interface has been implemented with AJAX.

In [43], the authors have presented iAIML (intentional AIML), a mechanism to add intentional information to AIML chatbots based on the Conversational Analysis Theory (CAT). In this work, CAT is used as a linguistic base and considers intentionality in adjacent pairs in dialogue for facilitating consistent dialogues with the user. To incorporate the intention information the authors proposed to redefined the AIML base by adding new elements and rules. The results of two experiments carried out with real interlocutors and with a chatbot expert to validate the iAIML mechanism have also been presented.

An adaptive chatbot with corpus-training approach has been proposed in [44], which helps the students in active learning of conversational skills. The corpus-training approach is developed by retraining an AIML based chatbot with a corpus, to chat in the language and topic of the training corpus. An evaluation of the chatbot shows that the students are much more interested in conversing with the bot as they could repeat the same material over and again without getting monotonous.

2.3.2 KB Enhancements

The work done by Augello et al. [45] focuses on enhancing the traditional chatbot knowledge base with the encyclopedic knowledge coming from DBpedia and also to provide it with associative capabilities. For this, two different but interconnected area in the chatbots "brain" has been proposed. The first one is a rational reasoning area, based on the integration of two types of structured KBs-the standard AIML KB and a DBpedia-based KB. The second one is an associative reasoning area obtained by building a Latent

Semantic Analysis (LSA) inspired semantic space in which Wikipedia entries are coded as vectors and are connected by geometric similarity relationships.

Most of the chatbot's knowledge base is constructed manually, which generally is a time-consuming task and may not be flexible enough to adapt to new domains. An automatic chatbot knowledge acquisition method from online forums is presented by Yu Wu et al. in [19]. They applied rough set and ensemble learning theory to data analysis and designed a reply classification model. To improve the recognition results, ensemble learning based on bagging is applied. The results are validated through an experiment on a child-care forum.

To overcome the difficulties of manual creation of linguistic knowledge base, Abu Shawar and Atwell have explored the corpus-based machine learning approach [21]. To automate the generation of the knowledge base, they developed a program to read the text in different languages from a corpus and convert this machine-readable text to the AIML knowledge base. With this, a range of different language speaking chatbots including English, Afrikaans [17], French and Arabic [46] languages have been generated. The implementation has also adopted the first word, most significant word and the first most significant word approaches of learning technique.

A smart answering web-based chatbot that can convert documents into knowledge base using OCR has been proposed by Ly Pichponreay et al. [20]. This conversion of documents into the knowledge base of the chatbot resolves the issues like typing and time conception which occurs in the manual generation of the knowledge base. This integrated

system can import documents from popular formats such as Portable Document Format (PDF) and digital photos and extracts texts using Optical Character Recognition (OCR) from files. Questions and answers from these documents are generated by the ranking algorithm and over generating transformations. Finally, the AIML knowledge base is updated by the generated question and answer pairs.

An approach to enhance semantic power and maintainability of a domain-oriented pattern-matching chatbot is proposed and illustrated within an industrial project, named FRASI [47]. To avoid the difficulty of manually writing the question-answer pairs of KB, this work proposes two methodologies, through ontology-based knowledge base preparation and symbolic reduction of user questions. The first approach exploits the possibilities of ontology to construct dynamic answers due to the inference process. The second method pre-processes the input queries to make simple structures that will be referred to the existing chatbot knowledge base. This symbolic reduction methodology can optimize the rule combination required to store the data in KB.

2.3.3 NLP Integration

NLP enables a machine to process and analyze the input provided to it in natural language, to understand the meaning and initiate appropriate action or response. Artificial intelligence and natural language processing technologies help chatbots to communicate with the users in their language. NLP powered chatbots help in creating a positive user experience by imparting human-like analysis traits to the chatbot. The following sections reviews some of the NLP based chatbot reported in the open literature.

To analyse natural language inputs, chatbots use certain keywords/pattern matching techniques. Most of these techniques face the issue of keyword arrangement for matching precedence and keyword variety for matching flexibility. Hence authors in [1], proposed One-Match and All-Match Categories (OMAMC) technique for keywords matching process. The possible keywords generated from the input by OMAMC is compared with the possible keywords created by previous matching techniques. Results of the study show that OMAMC technique is much more effective and it found to reduce the matching time and to widen the matching flexibility within the chatbot's keywords matching process.

The greatest deficiency of most chatbots is efficient response generation and semantic analysis. A conceptual framework for high-quality chatbots which points out for further improvements in this directions has been proposed in [48]. The author, Zdravkova proposes a solution by creating a lexicon extended with thesaurus intended for the simulation of semantic analysis. The generation of responses is based on many predefined templates for generic answers which depend on several keywords extracted from previous conversations.

The results of research towards the improvements of the chatbot technology, based on the implementation of a more proactive dialogue behavior have been described in [49]. The work proposes a mixed-initiative interaction methodology for enhancing chatbot proactivity. Additional to normal dialogue delivery, a proactive state driven by NLP algorithms initiates sub-dialogues which are managed by standard rules.

A Fairy tale Child chatbot has been implemented in [50], which

acts like a child that wants to hear a fairy tale. It analyzes the user utterances and tries to react with a question or remark if possible, simulating a curious child who can communicate in Czech or English. This chatbot is built using Treex natural language processing framework for both analyses of user queries and response generation.

To improve the conversational power of chatbots, Chakrabarti and Luger [51] designed an architecture that can leverage content and context. Instead of pairwise utterance exchanges, this approach makes chatbot to go beyond the question-answer exchange to hold a longer and more meaningful conversation with a human. An architecture for the chatbot containing the Chat Interface, which pre-processes the raw chat text, the Knowledge Engine, which provides the content of the conversation and the conversation engine, which manages the semantic context of the conversation, etc., has been described in detail.

2.3.4 Machine Learning Approaches

To improve the answering performance of conversational agents, many attempts using machine learning techniques have been carried out by the research community. Kim et al. in [52], proposed Semantic Bayesian Networks (SeBN) based agents. To infer the intentions of the user, this agent uses semantic Bayesian networks which are based on probabilistic models and respective semantic information.

In another attempt, a neural network based chatbot with Python is implemented in [53]. This approach uses Recurrent Neural Networks (RNN) to processes the data using its different network layers. The RNN model reads the input sequence one token at a time and predicts the output sequence. During the training time, the real output sequences

are provided to the model and trained to maximize the cross-entropy of the correct sequence given its context.

For generating informative and interesting chatbot responses, Xing et al. [54] integrated topic information into a sequence-to-sequence model. They proposed a topic aware sequence-to-sequence (TA-Seq2Seq) framework which incorporates the topic information to an encoder-decoder structure through Recurrent Neural Network model. After the topic keyword acquisition, this model leverages topic information by using biased generation probability and joint attention mechanism.

Even though neural models generate responses promisingly, they predict utterances one at a time, ignoring their impact on future out comes. The future direction of dialogue is important to generate coherent and interesting dialogues. In [55], authors modeled future direction in chatbot dialogues by applying deep reinforcement learning. This reinforcement learning framework generates responses by integrating the strengths of neural Seq-to-Seq systems with reinforcement learning for carrying out effective communication.

To improve the diversity of responses, the authors proposed a new matching model using an evolutionary algorithm in [3]. This model generates new sentences from existing ones using a modified form of Genetic Algorithm (GA). The initial population comprises of the sentences retrieved by the system and the fitness function used is computed by the number of common terms in the user query. By using the crossover operator, the algorithm breeds new sentences as new individuals of the GA.

2.3.5 Ontology Implementations

Ontologies, commonly referred to as semantic networks, are a set of relationally and hierarchically interconnected concepts. These concepts are interconnected into a graph, which allows to search through and infer new statements by applying reasoning rules [56]. Ontologies can be used in chatbots to find out synonyms, hyponyms and other relations between the linguistics constructs.

A new approach to develop an ontology-based chatbot, OntBot, is proposed in [57]. The ontology is mapped into relational databases and is used as the chatbot knowledge base. For this transformation, this framework applies a set of mapping rules that describe the generation of the relational model from the constructs of the ontological model. The main component of the OntBot is inference engine, which feeds with preprocessed user query and formulates the target response with the help of proposed Scope Specifier, Rule Matcher and Query Processor submodules.

Another implementation, named SynchroBot [58], which is powered on semantic web and NLP technologies, supports human-machine interaction in the domain of e-commerce. The knowledge base of this system is created in the Resource Description Framework (RDF) format, which represents the data as a triplet of subject, property and value. The user query is interpreted by natural language processing and the Expected Answer Type (EAT) is identified by considering relevant property and subject of the given query.

Most of the existing query processing systems disregard many important characteristics of a distributed ontology environment. Usage

of distributed ontology mapping methods to rewrite a query to generate an integrated answer for the query is proposed in [59]. The authors propose an implementation of Deeper Plan First Order (DPFO) scheduling method to increase the parallelism of executing sub plans included in a query. Previously scheduled queries are considered to eliminate repeated execution of same queries and help in generating faster responses.

Another ontology-based chatbot which can handle queries from e-commerce users is implemented in [60]. The proposed system can provide accurate details of products even without logging into the website. Ontology template is created through protégé platform that stores the data retrieved from the website API data source using jape rules. The dialog manager which is handled by Wit.ai framework and NLP. Wit.ai arranges the intent and other properties based on user query. A function call made to ontology template makes the chatbot answer the query and is returned to the user through the dialog manager.

2.3.6 Enhancing with Speech and Emotion

Emotion and voice are two important aspects of virtual personality for creating believable chatbots. Some of the attempts for providing emotion and speech ability to chatbot is being discussed in this section.

The implementation of a real-time conversational agent with an incorporated talking head is described in [61]. The presentation layer of the conversational agent is integrated with the talking head and the textual response is converted into facial movements of talking head with the synthetic voice.

Chatbot using natural interaction have the potential to act as an excellent and highly capable virtual-guide which can overcome the disadvantages of the traditional way of gathering information and eliminates dependency on prerecorded audio/visual guides. Santangelo et al. [62] presented a multimodal chatbot based virtual guide for cultural heritage tours. This multimodal interface integrated ALICE chatbot and an Automatic Speech Recognition engine through client-server paradigm developed with XHTML and voice (X+V) language.

In [63], the authors proposed a mobile Embodied Conversational Agent (ECA) platform for developing task-specific applications on hand-held devices. Automatic Speech Recognition (ASR) module converts the utterance that comes from the Voice Activity Detector to text and sends the resultant text to the Conversational Engine (CE). The CE module manages the dialog flow and produces the actions appropriate for the target domain. A Text-To-Speech (TTS) subsystem generates the voice from the CE response and visual head makes visemes from this voice and the corresponding facial expression.

In [64] authors showcase computer directed persuasive communication in the field of health care advise and have proposed an independent layered approach for emotional and social cues in human-computer dialogues system. The proposed system makes the user/patient feel that they are engaged in a human-to-human interaction. Vrajitoru in [65] has proposed a methodology to enhance the realism of the conversation by introducing two components for creating a personality to a chatbot, an emotional component and a personality-specific database.

2.3.7 Regional Language Chatbots

Majority of the present day chatbots are designed to communicate with the users in English. The number of chatbots that can converse in regional language is limited. It is quite evident that people who are not fluent or comfortable with English tend to sway away from a system that is in capable of processing queries in their native language.

Hettige and Karunananda developed the first Sinhala chatbot system [5], which can communicate between computer and user in Sinhala language. The prototype system is designed to work on a client-server model. User queries reads by server sockets and direct it into Sinhala language parsing system. This Sinhala parser system comprises of Morphological analyzer, parser, composer and three Lexical dictionaries. The Morphological analyser identifies grammatical information for each word in the user query and send to Sinhala parser. The Parser analyses the received tokenized word syntactically and identify syntax categories. Knowledge identification engine reads this information from parser and find matching pattern with the help of knowledge base.

A Chinese chatbot which is based on the AIML language, Xiao Huihui, was introduced in [22]. Chinese intelligent chatbot face the challenges of not having a perfect corpus and Chinese segmentation system. To overcome these issues, the authors proposed solutions by providing one answer to multiple questions and then merging similar questions. Prior to pattern matching the semantic analysis of the user's input was conducted using a Chinese word segmentation system which divided the input statement into individual words to understand their

semantics using machine language.

Fabian and Alexandru in [66], describe an application framework to facilitate natural language processing in the Romanian language. This chatbot is based on concepts of formal language theory which is implemented in Python and Django for data models and web interaction. Functionally the Romanian chatbot supports training and conversation operation modes. The authors also have suggested two approaches for further extending the framework based on fuzzy reasoning and stochastic approach.

JS and Araki developed humor enabled chatbot [67] for the Japanese language which outperforms two other chatbots in Japanese named Modalin [68] and Pundalin [69]. A very modular approach is followed in this framework to easily integrate the functionalities. Web N-Gram Model module handle the language related processing for Japanese. This module extracts the grammatical content words of the user query using the MeCab2 morphological analyser.

An Arabic chatbot named Qur'an28-30, which accepts user input in Arabic language and generates replies with appropriate sections from the Qur'an. Generate AIML based training corpus of this chatbot from Qur'an regarding Surah (chapter) and Ayat (sentence) using machine learning techniques [46]. Each Surah split into Ayat and tokenise, then generate the AIML categories which contain pattern and templates.

Even though there are many Indian languages prevailing in India, only a few research works have been reported in the language localisation of chatbots. One of such attempts is a bot named Deepti which is an interactive computer chat program that can speak in Hindi [70]. Deepti used Romanized Hindi for textual inputs and responses. The AIML

engine generates the responses to the user's stimulus both in the form of textual messages which is is passed through a text to speech system to generate a multi-modal response for the user. The proposed system can also initiate system-level actions to control the machines if required.

Another Indian language chatbot named Poongkuzhali [71], that communicates in Tamil on technical topics. The system accepts input in Tamil, picks out the keywords in the sentence, identifies the context of discussion using applying decomposition rules and responds to the user in Tamil by using a set of reassembly rules. The user is also free to provide input in normal form or transliterated form, after selecting a specific topic for discussion.

2.4 Task Specific Chatbots

In contrast to general-purpose conversational chatbots, many task-specific applications are developed for accomplishing specific tasks [72]. Such chatbots operate on specific domains like education, health care, e-commerce, etc.

2.4.1 Chatbots for Education

In a virtual environment, intelligent agents can stimulate learning by providing intelligence and enhanced believability. Implementation of chatbot as a pedagogical agent helps in eradicating barriers like time and distance and also enhances the efficiency of virtual communication. The pedagogical application of an AIML chatbot with text and voice interface is discussed in [73] with special reference to Intelligent Pedagogical Agents (IPA). The Implemented IPA can analyse user

actions against the expected ones, can provide timely feedback, find and rectify mistakes and updates student assessment data.

An English Dialogue Companion (EDC), to help elementary school students to learn English language has been proposed by Huang et al. [74]. EDC system design includes three learning activities, namely learning companion phase, the conversation phase and the teaching phase, to enhance learner's language learning. Fryer and Carpenter [75] describe a Jabberwacky chatbot for Foreign Language Learning (FLL). According to the authors, students tend to feel more relaxed talking to a chatbot as they do not get bored or lose their patience and can also provide quick and effective feedback.

Another notable chatbot in this field is Lucy proposed by Fei and Stephen in [76]. Lucy is an online intelligent language tutor that can carry out extensive conversations with learners as they speak into their computers through a microphone. Rodriguez et al. [77], have introduced two chatbots, T-Bot and Q-Bot, which are capable of teaching and evaluating students through open source platforms like Moodle or Claroline. T-Bots can effectively communicate with students in their natural language and provides a platform to look for required information. Q-Bot acts as an evaluation tool and supports tutors in evaluating students or self-evaluation by students.

Another chatbot to help school students to learn basic concepts of computer science like variables and conditionals has been described in [9], with a gamified, structured inquiry-based model. In this study, it was observed that there is considerable improvement in student participation, the intensity of concentration, enthusiasm and expressed interest when taught using chatbots.

Construction of a conversational agent to support online collaborative learning discussions through an approach called Academically Productive Talk (APT) is carried out in [78]. A new system for APT based dynamic collaborative learning support was developed using an agent-based architecture called Bazaar, which is an extension of earlier work called Basilica architecture. The classroom evaluation of the system provides evidence of significant positive effect. A framework to develop pedagogical chatbots has been proposed in [79] along with a practical implementation to help the children to understand the diversity of the urban ecosystem. The evaluation of the same suggests a high level of acceptability among students and teachers.

2.4.2 Health care Domain Chatbots

Apart from the educational domain, attempts have also been reported in the health care sector for te use of chatbots to cater the needs of the health seekers and health care provider. Manoj Kumar et al., demonstrate a sanative chatbot system in [80], which can provide instant replies to the questions posted by patients. This system focuses on giving appropriate answers for health seekers if the user query contains medical terms which match the database through a scheme to search the medical records using local mining and global approaches.

How to develop a chatbot which can help medical students in their studies has been presented in [81]. Medchatbot is developed using the AIML chatbot framework with the Unified Medical Language System as the domain knowledge base. MedChatBot could be incorporated in a tutoring system to support natural interaction with the students.

The authors in [82], developed a medicine consultant chatbot, known as Pharmabot: A Pediatric Generic Medicine Consultant chatbot. This chatbot can act as consultant pharmacist that will give the rational, appropriate and safe medication of generic drugs for children based on the information collected from the user by chatting.

2.4.3 E-commerce Applications

Usually, e-commerce websites have a variety of products with different features and users have to navigate through web pages to find relevant results, which could be time-consuming and annoying. Implementation of a chatbot which can improve user interaction with the e-commerce website is described in [10]. The bot understands and converses with the user in simple language and helps to make buying decisions and makes the ordering process easy and convenient for the user. The chatbot uses Rivescript to fetch responses based on user input.

Another customer service chatbot for an e-commerce website, named SuperAgent, is discussed in [83]. SuperAgent collects large-scale e-commerce data by crawling the HTML product page descriptions from the website. This agent consists of four different parallel engines, FAQ search for customer queries, Fact QA module to get product information, opinion-oriented text QA engine for handling customer reviews about the products and a chit-chat engine for replying to greeting queries.

Amir et al. introduced a chatbot that accept orders with minimum input from the customers for selling online goods [84]. The chatbot captures the information provided by the customer and in turn uses the

same for identifying the target audience and sends personalised deals and promotions. The system works with existing data of the WooCommerce system and the Telegram API.

In yet another attempt, Hemin Joshi et al. proposed an e-commerce engine based Cartbot which helps customers to place and manage orders on the e-commerce site in [85]. Three tier architecture of the Cartbot contains, an Android App, an intelligent agent and e-commerce engine. Cartbot helps users to interact with the e-commerce engine through the intelligent agent.

2.4.4 Customer Support

Implementation of chatbots in the field of customer relationship management has paved the way to revolutionize the traditional way of customer service. Authors in [86] describe how well chatbot system behaves when used in automated online customer care service which ensures better customer satisfaction with less cost and high efficiency with a pool of AIML based intelligent agents. While in [87] the authors, look into the possibility of intelligent agents in the domain of Customer Relationship Management over the Internet (e-CRM). An integrated model is proposed to show how intelligent agents can be used as powerful tools to achieve e-CRM. This study believes that in future intelligent agents will have a key role to play in CRM.

How a chatbot can work as a domain-specific information system and experiments on how to improve accuracy based on a specific domain is presented in [88]. ALICE chatbot converted as a domain-specific chatbot named University FAQbot is specifically tailored for providing query answering system for university students

with the objective of an undergraduate advisor in student information desk. The chatbot accepts natural language input from users, navigates through the information repository and responds with student information in natural language.

Anbang et al. [89] have presented a conversational agent on social media based on deep learning techniques like Long Short-Term Memory (LSTM) along with sequence to sequence learning and word embedding techniques. Customer's query is taken as input and the vector representation is calculated and is fed to LSTM. The system is trained with 1 million twitter conversations from over 60 brands.

The work in [90] depicts a technique to utilise the power of chatbots to serve as interactive support systems to large business enterprise applications like e-governance systems. The system uses an Artificial Support Entity (ASE) along with an already implemented prototype e-governance System, named IEGS. An AIML based chatbot engine has been incorporated to address the issues raised by computer illiterate audience to utilize the services offered by an enterprise application.

2.4.5 Other Works

Hoshino et al., describe a chat information service through spoken dialogues using a humanoid robot, ASIMO [91]. During conversation, this robot provides topical news information whether users explicitly ask for it or not. Priorities for possible answers in response to user inputs are determined according to the proposed dialogue strategy and also by searching items from news article database.

In yet another attempt, Rinkal and Suthar [92], have presented the

integration of an AIML chatbot for news on WhatsApp which can can run even on single board computers (SBC) like Raspberry Pi. Integration of AIML and news server connection to the WhatsApp server together makes the system work as a chatbot through which user and bot can communicate in natural language.

The work in [93], depicts how chatbots can be used as a search engine that can process next search by correlating responses of the previous searches incorporating the algorithm of Extension and Prerequisite. Such functionality strengthens the capability of chatbots and will produce a response which is very much in line with the context of the current conversation. Since such chatbots can evaluate and establish the correlation between the different set of responses, they can very well be used as a diagnostic system.

The way by which a chatbot system can be used as a tool to explore different types of English language used in British National Corpus (BNC) in a qualitative manner opposed to quantitative view is illustrated in [15]. By retraining ALICE with BNC spoken transcripts, the authors have achieved the ability to generate the largest language processing model, which contains more than one million dialogue-rules.

Rohan and Rishin presented [94] integration of chatbots with the Internet of Things (IoT), which is intended to resolve the shortcomings of existing IoT systems. This work present a conceptual system design for building chatbot incorporated to IoT frameworks. Chat engine identify the intent and map to IoT cloud platform through API and the IoT cloud platform is connected to IoT devices.

2.5 ALICE Chatbot Framework

ALICE is a natural language processing artificial intelligence chatbot that responds to queries. ALICE is an open source application that is available to the public for free under the GNU license. The framework uses Artificial Intelligence Mark-up Language (AIML) as the knowledge base. The main advantage of ALICE is the clear separation between the chat engine and the knowledge base. This separation helps in easily injecting the newly developed knowledge model as different AIML files without affecting the existing chatbot configuration.

The main components of ALICE chatbot are the AIML brain and an AIML interpreter. The AIML brain is the most important component of the architecture as it stores all AIML files and holds the entire knowledge base of the chatbot. The AIML interpreter is a software program that provides a user interface to accept input from the user. After accepting the input, it matches the query with the data available in different AIML files present in AIML brain and provides an output through the interface. The interpreter is highly customizable and it can load features that enable the user to select a specific bot personality or train the bot with new answers. In the following section, we present an overview of ALICE architecture and AIML.

2.5.1 Artificial Intelligence Mark-up Language (AIML)

AIML is an XML dialect with its specification developed by Dr. Richard S. Wallace during 1995–2000 [34]. This XML compliant dialect enables to encode the behaviour of chatbots in a standardized form that can be exchanged between different chatbot interpreters and

implementations [7]. When we consider a chatbot based on input-response model, AIML is one of the most commonly used building tools due to its simplicity and minimalistic yet extensible design features.

AIML describes a class of data objects named as AIML objects and partially describes the behaviour of computer programs that process them. AIML objects are made up of units called topics and categories, which contain parsed or unparsed data. The category is the basic unit of knowledge in AIML. Each category contains input query, the corresponding response(s) and an optional context [95]. The query is represented by the <pattern> element and the response(s) are represented by the <template> element. The optional contexts can be "that"and "topic", represented respectively by <that> and <topic> tags. <that> refers to chatbot's previous reply and <topic> groups set of categories. The functionality of the template element can be enhanced using constructs like srai, star etc.

The <srai> tag can target a multiple <pattern> for a single <template>. Hence AIML interpreter can efficiently answer for different user input having similar meaning. This tag can be used in multiple ways like symbolic reduction, synonym resolution etc. [34]. Another tag, <star>, is used to match wild card '$*$' character in <pattern> Tag. AIML can also utilises wild card symbols like '$*$' and '_'. Fig. 2.1 illustrates a basic unit of the AIML format.

2.5.2 Pre-processing Techniques for Pattern-matching

As discussed, AIML files require the use of an interpreter, for searching appropriate responses to the user queries. The AIML

```
<aiml version="1.0.1">
<topic name="The topic name"
        <category>
        <pattern>User Input</pattern>
        <that>Last response</that>
        <template>Chatbot answer</template>
        </category>
        <category>
        <pattern>User input * </pattern>
        <template><srai>User Input</srai></template>
        </category>
        ....
</topic>
</aiml>
```

Figure 2.1: Basic unit of the AIML format

interpreter read AIML objects and provide application-level functionality based on their structure with the help of the services of an XML processor [7]. Two pre-processing tasks, namely Input Normalisation and Load-time match path construction, are conducted for each input to the AIML interpreter before it starts the pattern matching procedure.

2.5.2.1 Input Normalisation

The AIML interpreter performs three different normalisation functions on all inputs before it attempts to find the match. The

minimal normalisation which is always carried out is the pattern-fitting normalisation. Other two additional normalisations, namely, sentence-splitting and substitution is perform based on the complexity to of input query.

If the AIML interpreter carries out substitution normalisations on input, then it needs to be perform before other normalisations techniques. However, the pattern-fitting normalisation process receives the output of the sentence-splitting normalisation process or substitution normalisation process or the direct input.

i. Substitution Normalisation

Substitution normalisations attempt to retain the information in the input which is susceptible to loss in next steps of normalisations. As a illustration, this process can differentiate the dot notation used in different context, used as an abbreviation, end of sentence or a prefix of file extension name. Other than the end of the sentence context, all other cases are replaced by a notation with appropriate meaning.

ii. Sentence-splitting Normalisation

Sentence-splitting normalisations are heuristics applied to an input that splits it into sentences using simple rules like breaking sentences at periods, question marks or exclamation symbols. This performed after the substitutions done during the substitution normalisation phase, for instance, substituting full words for their abbreviations.

iii. Pattern-fitting Normalisation

Pattern-fitting normalisations on input must remove all characters that are not normal. It removes punctuation from input sentences and

Table 2.2: Examples of Input normalisation

Input	Substitution	Sentence-splitting	Pattern-fitting
"When is the last train?"	"When is the last train?"	"When is the last train"	"WHEN IS THE LAST TRAIN"
"Please, go to http://cusat.in!"	"Please, go to http://cusat dot in!"	"Please, go to http://cusat dot in"	"PLEASE GO TO HTTP CUSAT DOT IN"
":-) That's amazing."	"That is amazing."	"That is amazing"	"THAT IS AMAZING"
"I don't know. Do you, or will you, have a main.txt file?"	"I do not know. Do you, or will you, have a main dot txt file?"	"I do not know" "Do you, or will you, have a main dot txt file"	"I DO NOT KNOW" "DO YOU OR WILL YOU HAVE A MAIN DOT TXT FILE"

converts the input letters to upper case to facilitate matching with the knowledge base pattern. Table 2.2 [7] shows example of input normalisations and the corresponding outcomes.

2.5.2.2 Load-time match and Input path construction

i. Match path construction

The AIML interpreter constructs a match path for each category from the AIML file at load time itself. The match path corresponding to each category has three components:

- pattern: the contents of the category element's *pattern* element
- that: the contents of the category element's pattern side *that* element. The default value is *.
- topic: the contents of the name attribute of the *topic* element,

Table 2.3: Input paths construction

Normalised input	**Previous output**	**Value of topic predicate**	**Input path**
"YES"	"DID YOU GO THERE"	""	"YES, <that> DID YOU GO THERE <topic> * "
"MY NAME IS JINI"	"I GUESS SO"	"FOOD"	"MY NAME IS JINI <that> I GUESS SO <topic> FOOD"

parent of the category element. The default value is *.

ii. Input path construction

The AIML interpreter also forms an input path from the normalised form of the user query. The input path thus generated contains three components with mandatory order and do correspond to the three components of load time match path.

- pattern: the normalised input query
- that: the previous bot output, normalised according to the same rules as in input normalisation
- topic: the current value of the topic predicate

Both match path and input path are represented in simple string format separated by the markers like <that> and <topic>. Table 2.3, shows some examples of an input paths constructed from different user query [7]. At run-time, an AIML interpreter must matches the constructed input path against each of the match path constructed during the load time using pattern matching algorithm.

2.5.3 The ALICE Pattern-Matching Algorithm

ALICE applies a heuristic pattern-matching algorithm to the path generated from the normalised form of input query to obtain the matching pattern from the match path generated from AIML files. Each input path from the user query is matched pattern by pattern, against the total set of loaded match path. As soon as the first complete match is found, the process stops and the category whose pattern was matched is selected. The AIML interpreter then constructs the output response from the template of the selected category.

To reduce the pattern matching time as well as the memory required, all the categories in the AIML files are stored as a tree managed structure called Graphmaster [95]. A Graphmaster consists of Nodemappers, which is a collection of nodes and it maps the branches from each node. These branches are either single words or wildcards. The Graphmaster's graph is a rooted directed tree with root r and node n. The Graphmaster stores patterns along a path from the root to a terminal node t, where the AIML template is stored. Let $w_1,\ldots,w_k$ is the sequence of k words in an AIML pattern and $G(n,w)$ is a function which is either undefined or returns the value of a successor node m in the graph. To insert the pattern into the graph, the Graphmaster initially checks whether $m = G(r, w_1)$ exists or not. If it does, then the program continues the insertion of $w_2,\ldots,w_k$ in the subtree rooted at m. When a first index i is encountered where $\exists\ n \mid G(n, w_i)$ is undefined, then the program creates a new node $m = G(n, w_i)$. After that the Graphmaster adds a set of new nodes for each of the remaining words $w_i, \ldots, w_k$. In this way the Graphmaster stores common pattern prefixes along pathways from the root to

achieve a significant compression compared to a linear array or database of all the patterns.

Fig. 2.2 shows the Graphmaster that represents ALICE brain loaded with 24,637 categories [34]. The spine is a log spiral with an exponent close to unity making it very similar to a linear spiral. Gray lines indicate nodes with one branch and black lines are nodes with two or more branches. The leaf nodes have two branches as it stores both <template> and <filename>.

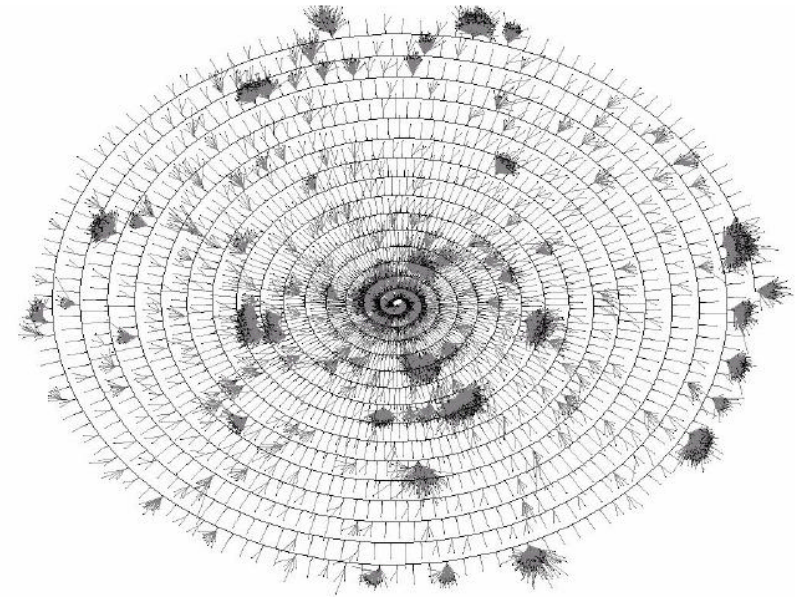


Figure 2.2: Graphmaster that represents ALICE brain

Algorithm for matching input to a pattern is described in Algorithm. 2.1 and flowchart of the same is represented in Fig. 2.3 [7]. Graphmaster matching process employs a depth-first search technique with backtracking. Assume that the input is a sequence of words starting with word "X". The algorithm to match an input to a pattern consists of mainly three cases based on whether the Nodemapper

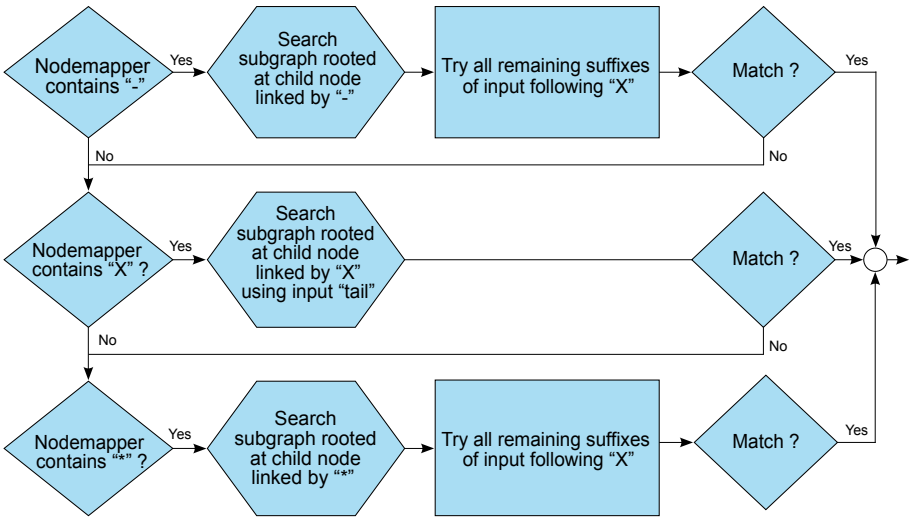


Figure 2.3: Flowchart of matching an input to a pattern

contains the key "_", "X" or "∗". If any of the keys are found, then search the subgraph rooted at the child node linked by the corresponding key. If no match is found, go back to the parent of this node and put "X" back on the head of the input.

It may be noted that, at every node, the "_" wildcard has the highest priority, an atomic word carries second priority and the "∗" wildcard has the lowest priority. The patterns need not be ordered alphabetically. They are partially ordered so that "_" comes before any word and "∗" comes after any word. The matching is performed on a word-by-word basis and not on a category-by-category basis.

2.5.4 AIML Platform for Development

A chatbot development platform provides necessary tools and technologies to reduce the cost and the complexity of development, deployment and management of chatbot systems [96]. Some of the

technologies that can be pointed out are the database integration, Web services for an online chatbot and the log recording of the conversations with the users. There is a wide range of development platform available, written in different programming languages like Java, .NET, Python, etc.

Algorithm 2.1 Matching an input to a pattern

1: **procedure** PATTERNMATCHING
2: Assume the user input starts with word X
3: Root of this tree structure is a folder of the file system that contains all patterns and templates, ie nodemapper
4: If the nodemapper contains the key _, then, search the subgraph rooted at the child node started with _. Scan and try to match all words suffixed X
5: If no match found, then go back to the folder, find another subgraph starting with word X, start search for matching the tail of X. If no match, then goto next step
6: Go back to the folder, find the subgraph rooted at the child node linked by *. Try all suffixes of input following "X" to see one match. If no match was found, go back up the graph to the parent of this node, and put X back on the head of the input
7: Once the match found, then halt the the process of search. If the input is null and the Nodemapper contains the <template> key, then a match was found. The template that belongs to that category is processed by the interpreter to construct the output.
8: If match is not found, interpreter generates a random answer like "sorry, not able to understand"

Program D is the most broadly used open source AIML bot

development platform. It is one of the most feature complete and best-tested implementation of present AIML specification [97]. In a single server instance, it supports unlimited multiple bots. It consists of an open-ended architecture that supports interaction through any interface imaginable. It also comes with an automated testing framework and has got an AIML Test Suite that verifies whether the program is complying to AIML specification. Program D works with different languages and character sets. It can be integrated into different application frameworks due to its component-oriented architecture [98, 99] and is used for both academic research and commercial applications. In this work, we have made use of this framework for the prototype development written in Java.

2.6 Chatbot Evaluation Methodologies

One of the important questions raised in chatbot research is how quality assurance can be performed on chatbots. Quantitatively describing the performance of a conversational agent and comparing the same with the performance of other chatbots is a challenging task. Since the definition of effectiveness and naturalness of a conversation highly depends on perceptiveness of the user and also on the conversational flow, the evaluation of a conversational dialog system has always remained as a controversial topic [100]. When dealing with different kinds of natural language systems in different domains, evaluation becomes very subjective because of the increased complexity.

One of the biggest challenges faced by researches in evaluating a dialogue system is the absence of a widely accepted metric. It can be

noted that a significant amount of work has been done on evaluating goal-oriented dialogue systems. Goal-oriented dialogue systems can be easily evaluated when compared to open-ended systems as it can be measured based on the successful completion of various tasks [101].

To evaluate the overall performance of a chatbot, different metrics need to be measured by considering various perspectives of chatbot development like information retrieval, user experience, linguistic, AI perspective. The performance of a question answering service based chatbot can be measured regarding accuracy metrics like precision, recall, accuracy and F-score metrics [102]. Measuring the level of user experience is another popular approach which can evaluate the degree of usability, task completion rate and user satisfaction. However, it is an expensive and time-consuming way and hence calls for AI-oriented measure such as the Turing Test for evaluation. Loebner prize competition has been used to evaluate chatbots.

2.6.1 Turing Test

In the area of AI, it was Turing who started exploring the possibility of making a machine think and raised the question – "Can a machine think?" [26] where thinking is considered as the ability of humans. Turing suggested an "imitation game", that acts as a tool for measuring the level of success of researchers in the field of artificial intelligence [103]. Turing Test is otherwise known as imitation game and the chatbots participating in this game aims at carrying out conversation an indistinguishable from humans. During the test, a human observer commonly known as a judge carries out a conversation over a computer link with someone [56]. This someone can either be a

real person or a chatbot. After the stipulated period, if the judge believes that he/she was talking to a real person, then the chatbot will pass the test.

2.6.2 Loebner Prize

Hugh Loebner and The Cambridge Centre for Behavioural Studies agreed in the year 1990 to set up a competition based on implementing the Turing Test. Hugh Loebner announced a gold medal and a grand prize for the computer that can disguise as a human and can provide responses to the users making them believe that they are communicating with a real human. They have decided to conduct this contest on an annual basis and they offered medal and an annual prize for the computers that are more human when compared to other competitors [103]. This test is regarded as the first competition of its kind that represented a formal instantiation of Turing Test [56].

Since 1991, the competition is held every year with slight changes when compared to the original version of the competition. The foremost requirement in this competition is to have a chatbot that can converse with a judge. After 10 minutes of conversation, the judge has to decide whether the conversation that happened with a real human or not and scale of 1 to 4 that rates the level of being non-human to human [104]. The chatbot that best emulates the human is considered as the winner. So far, no chatbot has won the gold medal. Only a few chatbots could achieve a rating of 3 out of 12 judges believing that they are chatting to a real human.

2.6.3 Chatbot Evaluation - Past Works

Walker et al. have presented PARADISE (PARAdigm for Dialogue System Evaluation) framework for evaluating spoken dialogue agents [105]. The framework is capable enough to compare agents performing different tasks by normalising for task complexity. This performance model considers different factors and their relative contribution to evaluate the overall performance of dialogue agents. PARADISE uses decision theory methods to derive the performance function by combining the set of performance measures based on task success, maximizing user satisfaction and minimizing cost measures. In addition to the use of decision theory methods paradise also calculates the Kappa coefficient from the confusion matrix and quantify relative contribution by applying linear regression.

Goh et al. [102] pointed out the drawbacks of existing measures to evaluate response quality and presented a solution for evaluating the quality of response of dialogue agents. This model evaluates and ranks chat agents using a black box approach which involves observation, classification and a scoring mechanism. The authors have proposed a classification scheme with three category codes that can handle all possible responses from the systems being evaluated. After observation, evaluators classify the response from each chat agent to any of the proposed categories. Once all responses are categorised, it is then subjected to a scoring mechanism which helps to identify the best dialogue agent regarding the quality of responses.

Usage of different measurement metrics for evaluating the different aspects of the chatbot system is explained in [106]. The authors advocate

the usage of openly ended trials by the real users and the system is evaluated using glass box dialogue efficiency metrics, black box dialogue quality metrics and user satisfaction feedback. The authors suggest that the evaluation method as well as the way by which it is conducted has to be based upon the application and user needs. According to the authors, it is better to adopt an evaluation strategy based on application and user needs rather than following standard evaluation methodologies.

David Coniam [107] evaluates the linguistic accuracy of chatbots by examining their responses based on accuracy and aptness on a grammatical level and also on a word level. While Word level analysis was based on vocabulary range, upper/lower case and spellings, the grammar level checks were based on nouns, pronouns, verbs, articles, question words and their order. According to the author, the chatbots should meet the basic requirements of linguistic accuracy regarding the correctness of grammar, sensibility and meaningfulness.

Authors proposed an approach for assessing emotional agents architectures by evaluating the quality metrics of the design of architecture in [108]. A three-level measurement framework that includes conceptual, quantitative and operational level has been proposed to derive the metrics to ascertain the quality. Extensibility, complexity and modularity attributes are considered in determining the quality of architecture and twelve metrics have been captured using the Goal Question Metric (GQM) approach to evaluate the architectures of agents with emotions. This framework is useful for those who are involved in developing emotional agents as they can have an objective criteria to analyse and determine the most suitable architecture, satisfying desired qualities.

Kaleem et al. emphasis on a software quality model that can be used to evaluate conversational agents in [109]. A new framework is proposed in this work that evaluates two conversational agents deployed in two different contexts. The proposed framework applies the Goal Question Metric (GQM) approach [110] formulated by Fenton and Pfleeger. GQM approach is based upon the assumption that the goals of the system must be ascertained first and the same has to be matched with the questions that are required to answer the goals operationally. The next step is to find measurable metrics by interpreting the questions corresponding to each goal. This proposed framework can evaluate from an objective and subjective perspective to provide an overall performance measure.

An evaluation framework which is independent of natural language has been proposed for text-based conversational agents in [111]. This framework is based on the exchange of specific information between chatbot and user, this "Information Requirements" should be preset in the user's query and tagged rules in the applied domain. Based on the generated quality of response and pattern scripting, this system evaluates aspects like Response and Scripting for each conversation. Each aspect has four potential results and provides a general overview of the chatbot performance.

An evaluation strategy that is comprehensive enough to take care of non-goal-oriented dialogue systems is proposed in [101]. The backbone of the strategy is the presence of multiple metrics that do a granular analysis of dialogue agents and hence reduces the subjectivity of evaluation by selecting those metrics that go in hand with human judgment. The authors aim at creating an automatic evaluation process for dialogue agents as it will help to address the shortcomings of existing evaluation methods like incomparable elements between

humans and conversational agents, generating interesting content and plausible responses, the difference in objectives when it comes to evaluating the response and the AI itself. They have created an evaluation model based on metrics such as Conversational User Experience, Engagement, Coherence, Conversational Depth, Topical Diversity and Domain coverage and unified these metrics to gain a single metric for comparison and evaluation.

2.7 Named Entity Recognition

Named Entity Recognition (NER) is a subtask of information extraction which extracts factual information from text expressed in natural language. It identifies and classifies proper nouns into its predefined categories such as person, location, organization, time, date, quantities, monetary values, etc. [112]. NER models a sequence classification problem and is solved by methods such as Hidden Markov Models (HMM), Conditional Random Fields (CRF) etc.

The main aim of NER is to distinguish entities that represents noun phrases from normal texts. Entities are categorized into three types of label, each of which uses specific attributes for a particular entity type [113]. The three entity groups are named entity (PERSON, LOCATION, ORGANIZATION, MISC), numerical entity (MONEY, NUMBER, ORDINAL, PERCENT) and temporal entity (DATE, TIME, DURATION, SET). The different categories of entities are depicted in Fig. 2.4.

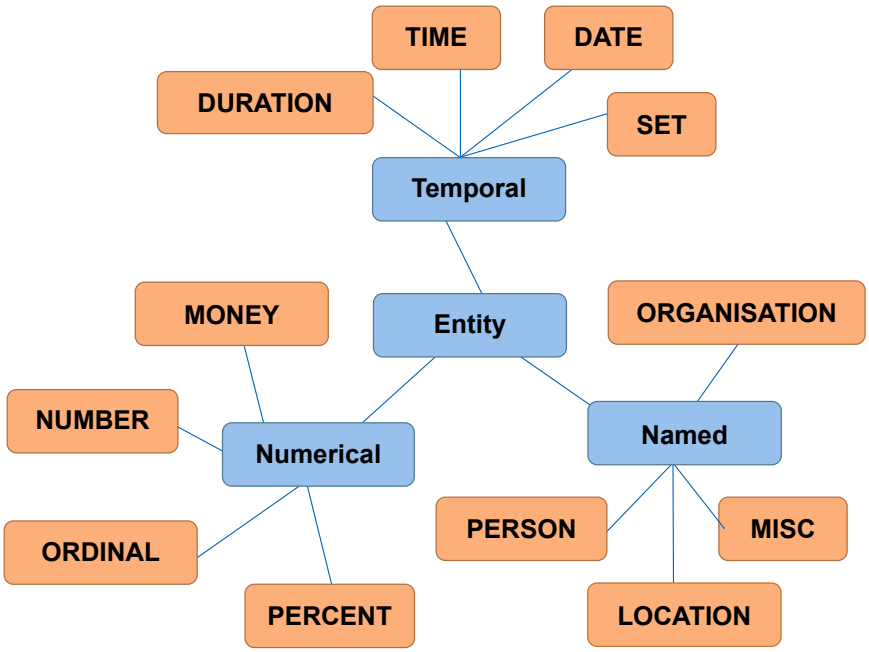


Figure 2.4: Different Named Entity categories

2.7.1 Technical Approaches of NER

Named Entity Recognition is gaining more importance and popularity over the past few years. It can be noted that many studies are conducted and reported on automatic detection of named entities. The technical approaches to build NER systems are broadly divided into three categories *viz.*, Rule-based, Machine Learning and Hybrid approach [114] and is depicted in Fig. 2.5.

2.7.1.1 Rule Based Approach

Rule-based approaches create a set of manually crafted rules based on syntactic, linguistic and domain knowledge to recognize particular

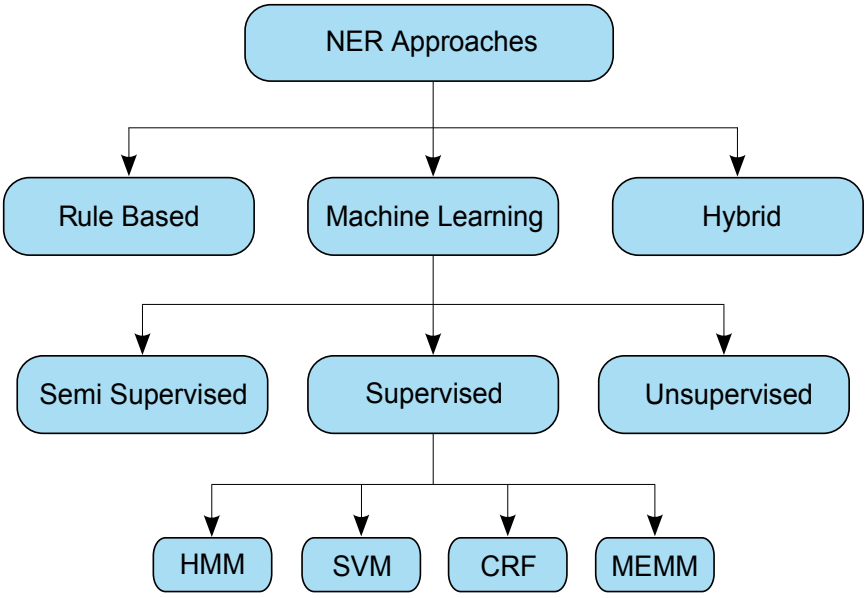


Figure 2.5: Technical approaches of NER

named entity type [115]. This approach is also known as hand-crafted rules or linguistic approach as the rules are manually designed for writing for a specific domain or language; and as such the rules may not always be portable and robust. Another limitation of this approach is that it requires a lot of rules for identifying the entities and this calls for alternative approaches.

2.7.1.2 Machine Learning Approach

Compared to the rule-based approach where the rules are composed manually, the machine learning approach employs a statistical model for classifying text to exact entities by applying machine learning algorithms. Machine learning based approaches can be easily applied to different domains and are more efficient and frequently used compared to a rule-based approach. Three types of learning models

that can be used for recognition are supervised, semi-supervised and unsupervised [114].

1. Supervised learning

Supervised learning algorithms can be used to discover NER rules from the labeled training dataset. In this approach, human experts identify the instances of named entities from a set of labeled training dataset input and utilise tagged corpus to develop the model. Even though supervised methods provides the best performance, it needs huge annotated data for training purpose and the same has to be done manually by experts in the domain, which is very time-consuming. Following are the different supervised algorithms for NER.

a. Hidden Markov Model (HMM)

HMM is a statistically based approach where the states are either hidden or unobserved [116]. It applies Markov Chain Property, where they consider successive states and evaluates the dependency between the two stages. It checks how the occurrence of one state is dependent on the previous state. HMM can be applied for named entity identification [117] by representing words by states and computing the probability of a sequence word by applying Markov chain. The main drawback of this approach is the requirement of a large amount of training to obtain satisfactory results. Hence this approach may not be suitable for limited data sets [116].

b. Maximum Entropy Markov Model (MEMM)

Maximum Entropy Markov Model is introduced by combining Hidden Markov Model and Maximum Entropy Model for NER [118]. At the time of training, this model connects the unknown values in the

Markov chain which is not conditionally independent. This model addresses the problem of large dependency faced in the HMM model. When compared to HMM, the recall and precision rate is more for MEMM. One of the disadvantages of this approach is the label biasing, the probabilities of transition from a particular state must be sum to one [115].

c. Support Vector Machines (SVM)

SVM was introduced by Vapnik in 1995 [119] and provides high accuracy results when it comes to text categorisation. The main objective of SVM classifier technique is to identify whether the entity or vector under observation belongs to the target class or not. During the training phase, a hyperplane is generated to classify the members into two classes which lie on the opposite sides of the hyperplane. SVM can model nonlinear relations and provide high accuracy for the text categorization. However, SVM does not consider state-to-state dependencies and feature-to-state dependencies. SVM based NER system was proposed by Yamada et al. and was first applied for Japanese language [120].

d. Conditional Random Field (CRF) Model

Lafferty et al. in 2001 [121] introduced CRF as a statistical modeling tool which can be applied in machine learning and pattern recognition for structured prediction. It is a probabilistic graphical model that can be used for sequence labeling. This model is different from other models as it takes into consideration the neighbouring samples and computes the conditional probability on the following nodes given the value of present nodes. This becomes very handy when compared to an ordinary classifier which disregards neighbouring samples and predict a label for a

single sample. The CRF model also assumes that features are dependent and there is no label biasing problem as experienced in MEMM [122]. CRFs perform better than HMMs and MEMMs when the true data distribution has higher-order dependencies than the model [121] .

CRFs are essentially a way of combining the advantages of classification and graphical modeling [123]. CRF can use linear chain or more general graphical structures to compactly model multivariate data for prediction. While general CRF models are employed for predicting complex structures, like graphs and trees, linear chain CRF are commonly used for sequence labeling.

The linear chain CRF is a discriminative undirected probabilistic graphical model which predicts the sequence of labels for the sequence of input text. In the graphical model representation, random variables are represented using vertices and the associations between random variables are represented by edges. Fig. 2.6, illustrates a simple linear-chain CRF model, that depends on the current as well as previous labels. Given a word sequence, CRF works in a way to identify the best entity tag sequence using conditional probability estimation. It can be noted that for a particular word sequence, CRF maximizes the conditional probability distribution of tag sequences.

Let's consider $X = x_1, x_2, x_3 \ldots x_n$ to be the input sequence and $Y = y_1, y_2, y_3, \ldots . y_n$ be the corresponding label sequence. The conditional probability distribution $P(y|x)$ given the input sequence is maximised by CRF [123] and is given by.

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp \left\{ \sum_{k=1}^{K} \Theta_k f_k \left(y_t, y_{t-1}, x_t \right) \right\} \tag{2.1}$$

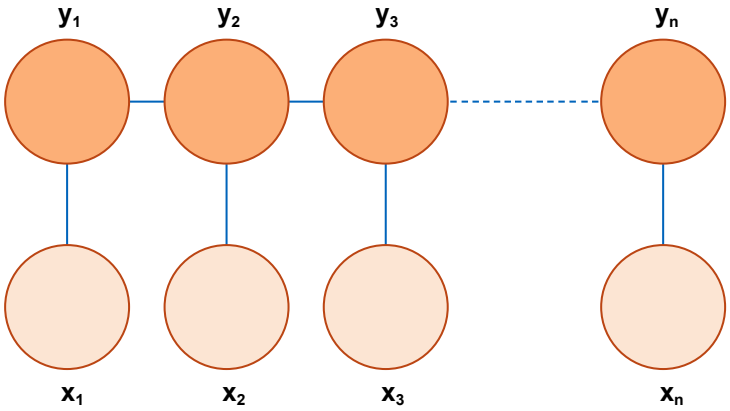


Figure 2.6: NER model

where $\Theta = \{\Theta_k\} \in \Re^k$ be a parameter vector and $\left\{f_k(y, y^{'}, x_t)\right\}_{k=1}^{K}$ be a set of real-valued feature functions. $Z(x)$ is an instance-specific normalisation function.

$$Z(x) = \sum_{y} \prod_{t=1}^{T} exp \left\{ \sum_{k=1}^{K} \Theta_k f_k \left(y_t, y_{t-1}, x_t\right) \right\} \tag{2.2}$$

2. Unsupervised Learning

NER models of based on supervised learning demand a robust set of features and large annotated corpus for learning a good model and the same may not be available easily for most of the languages. Unsupervised techniques for NER takes care of the lack of annotated text across domains and languages [114]. Unsupervised methods can be applied even if there is no labeled data and there is no need for a supervisor as it learns representations from the data.

3. Semi Supervised Learning (SSL)

This approach uses both labeled as well as an unlabeled corpus to

create the hypothesis. Algorithms mainly start with a small amount of seed data set and create more hypothesis using a large amount of unlabeled corpus. Lack of annotated corpus and data scarcity problem can be addressed by semi-supervised algorithm [124]. The main technique for SSL is called bootstrapping and requires a minimum level of supervision in the form of set of seeds for starting the learning process.

2.7.1.3 Hybrid Approach

Rule-based and learning based approaches have its disadvantages. By combining both approaches, can address individual drawbacks and can improve the performance. In the case of the hybrid approach, the rule-based approach is integrated with the machine learning approach, resulting in optimisation of overall performance [114]. When it comes to hybridisation of NER, the efficiency can be improved by combining different machine learning technique with rules.

2.7.2 NER Tools

Many tools for recognition of named entities have been developed and are available as open source as mentioned in Table 2.4 [125]. One of the widely used NER tool, Stanford Named Entity recogniser comprises of statistical algorithms based on CRF and maximum entropy and is developed in JAVA. Lingpipe is a toolkit that is used for computational linguistics based on dictionary lookup and Hidden Markov model. Another NER tool, Yamcha is an open source text chunker based on Support Vector Machine and is customisable and generic. Sanchay is yet another tool kit based on object-oriented

Table 2.4: NER tools

Tool	URL
Stanford NER	http://nlp.stanford.edu/software/crf-ner.shtml
Lingpipe	http://alias-i.com/lingpipe/
Yamcha	http://chasen.org/∼taku/software/yamcha/
Sanchay	http://sanchay.co.in/
CRF++	http://crfpp.googlecode.com/svm/trunk/doc/
Mallet	http://mallet.cs.umass.edu

concepts which emphasis on modularity, reusability, extensibility and maintainability. Another implementation is CRF++, which is very simple and customisable and employs Conditional Random Fields using C++ with standard template library. Mallet is another tools for NER extraction based on linear-chain CRF.

2.7.2.1 Stanford CoreNLP Tool

Stanford CoreNLP is an integrated NLP toolkit that contains a set of natural language analysis tools [126]. This Java-based annotation pipeline framework provides most of the common NLP techniques, from tokenisation to co-reference resolution [127]. It not only supports the English language but also some other major languages like German, French and Chinese. Stanford CoreNLP integrates many NLP tools, including part-of-speech (POS) tagger, NER, parser, co-reference resolution system, sentiment analysis, bootstrapped pattern learning and the open information extraction tools [126]. This framework is used by various field of academia and industry for applications involving rule-based, probabilistic machine learning and deep learning techniques.

The Stanford NER, an information extraction tool in the CoreNLP

framework, can be used to detect named entities in the text. By default, for English language, Stanford NER annotator recognizes named entities like a person, location etc., numerical values like money, number, ordinal etc. and temporal like date and time. The functionality can be extended by adding regular expression annotator for additional entity classes like email, city, country, URL etc. [128]. This tool provides a general implementation of a linear chain based Conditional Random Field (CRF) sequence model. Named entities are recognized using a combination of three CRF sequence taggers trained on various corpora including CoNLL 2003 (Computational Natural Language Learning), MUC (Message Understanding Conference) 6 and MUC 7 and numerical entities are recognized using a rule-based system [129]. The architecture of the Stanford NER system is depicted in Fig. 2.7.

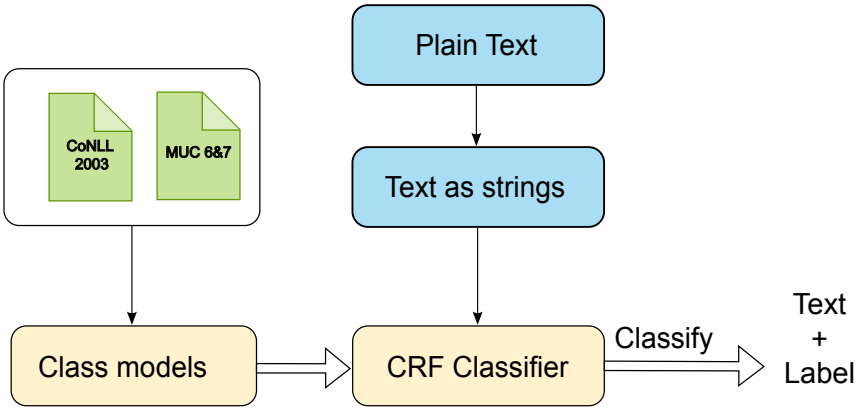


Figure 2.7: NER architecture

When a plain text sequence is given to the trained NER tool, the system identifies the entities present and outputs the text along with the proper label. As an illustration, the outputs of two sentences after

labeling the entities present are shown in Fig. 2.8. In the first sentence, NER system has identified tomorrow as Time entity and Bangalore as City. In the next sentence, the system has identified John (Person) and 1000 dollars (Money).

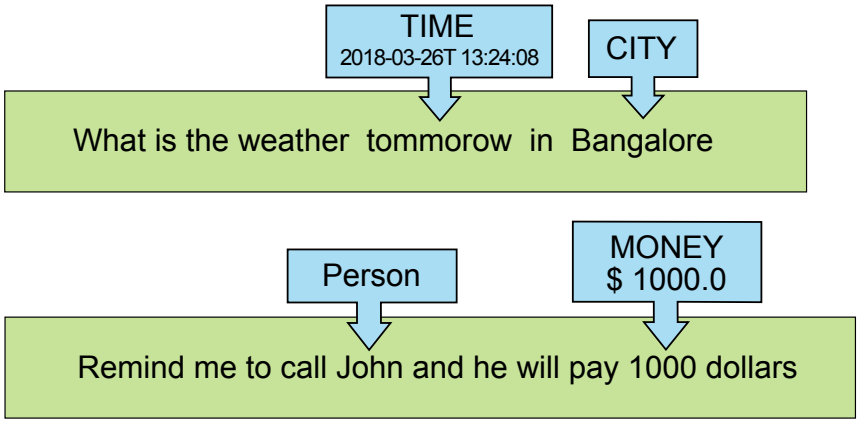


Figure 2.8: Entity extracted by NER

2.8 Summary

This chapter presents a review of pertinent works reported in open literature on the topic covered by the thesis. A detailed survey on the chatbots and their applicability in various domains have been included. Detailed analysis of the architecture, knowledge base and pattern matching algorithm of ALICE chatbot has been discussed in this chapter. Different evaluation methodologies of the chatbot are studied in detail and consolidated. The chapter also gives a comprehensive study about different Named Entity Recognition technologies along with an elaborated study of Stanford NER tool.

Chapter 3

Inquisitive Chatbot Framework

3.1 Introduction

The successful implementation of a chatbot calls for an efficient analysis of the user's query by the chatbot and the formation of the appropriate response that should be given to the user. Different kinds of technologies like AIML, deep learning, NLP etc. [2] are being employed for proper response generation. Rather being scripted as linear conversational agents, chatbots are expected to be advanced intelligent conversational agents and the same is made possible with the introduction of Natural Language Processing (NLP) techniques.

Natural language processing is an artificial intelligence technology that supports in interpreting, recognizing and understanding inputs in the form of human language. NLP tools have helped chatbots to

exhibit their real potential by adding human nature to it. However in many scenarios, the information available from the original query of the user may be inadequate to provide a suitable response. In such contexts, an intelligent system should be capable enough to identify the gap and should be able to probe the user further to collect the missing information, to provide an adequate response to the user query.

This chapter details the design of an inquisitive chatbot that identifies the missing information in query and probes the user to collect the information that is required to answer the query. Implementation of first level inquisitiveness of the chatbot that results in proactive interaction with the user is explained in this chapter. The same is demonstrated using a case study about a college information system. An approach for enhancing the inquisitive capabilities of chatbot using Named Entity Recognition (NER) has also been devised and the same has been presented in section 3.5.

3.2 Inquisitiveness

Generally, AIML chatbots can respond to a set of predefined queries and their variants which are stored in a knowledge base. The agents could answer only the queries, and cannot probe the user meaningfully for further information collection. This is a very important aspect of a chatbot implementation, for two main reasons; one being that, it could make the conversations more natural and user-friendly. The second reason is attributed to the fact that the user's query may not always contain sufficient information to provide an answer. In other words, the query is ambiguous and the bot needs more information to provide a proper answer to the query. One main

challenge involved here is the identification of the information which is missing. After identifying the missing information, the system should have the capability to probe the user, evaluate the answer given by the user and provide a response for the original query.

3.2.1 Level of Inquisitiveness

Probably the user's query may contain one or more missing information and in such a situation, the bot needs to come back with one or more questions for collecting the same. Depending on the number of insufficient information in the user's query, the level of complexity for probing the user increases. First level inquisitive query mechanism will return only one inquisitive question to the user. For the second level, the number of missing information is two, and thus, the chatbot is required to pose a maximum of two questions to the user. In complex scenarios, may be more than two proactive queries needs to be asked by the agent to reach the appropriate answer and can be referred to as multilevel inquisitiveness.

The number of inquisitive queries and missing information is directly proportional.

$$Q_n \propto D_n \tag{3.1}$$

where Q_n is the number of questions needs to be asked by the chatbot to the user and D_n is the number of missing information which is required to answer the user's query accurately.

As an illustration of the level of inquisitiveness, consider the case of a college information desk. Typically, in a college, the receptionist handles the queries raised by the user and provides the relevant information

related to the college. Instead of the receptionist, we may use a chatbot application for providing information to the users in real time. Such an interactive chat agent is expected to assist the parents or students for providing information regarding admission, academic transportation, hostel accommodation etc.

Consider a query asked by the student:

User: Who is the principal?
Bot: Dr. Srinivas Iyer

Here the system can directly look up in the knowledge base and can directly fetch the answer as there is no ambiguity in the question. However, the situation changes when there is ambiguity in the query.

User: Who is the HOD?

Here, the user wants to know who is the head of a particular department. When the system looks up for an answer, it identifies the ambiguity in the query as the system returns the name of different Head of the Department (HOD) in the college. In this case, the missing information is the name of the department. Since the user has not specified the department name, the chatbot needs to be inquisitive and should ask the user for more information. Here the maximum number of inquisitive questions required is one, and thus this is a case of first level inquisitive query mechanism for finding the missing data.

Let us consider another query.

User: Who is our course coordinator?

The query misses two pieces of the relevant information required for providing an appropriate answer: department name and course name.

Hence we can consider this as a second level query mechanism. Likewise, the chatbot should identify the missing information and collect it from the user through a set of inquisitive queries.

3.3 Inquisitive AIML Chatbot

In the existing AIML based chatbots, the chat engine identifies the suitable answer to the user's query, using pattern matching algorithms with the help of the AIML knowledge base which stores a set of predefined queries and its variants. However, when information changes, it is also required to change the AIML which will be very difficult especially if the changes are frequent. Moreover, the chatbot may require a restart on AIML knowledge base updation to reflect the changes as the AIML interpreter loads the file on program start itself. To avoid such difficulties we may seek the help of a hybrid knowledge base model, involving AIML and Relational Database Management System (RDBMS) to make the hard-coded answers dynamic.

3.3.1 Design of Knowledge Base Engine

The proposed system augments an additional Knowledge Base engine (KB engine) to the existing system and interfaces this with a database for fetching factual data for responding to certain queries. Fig. 3.1 depicts the architecture with the augmented KB engine with other elements involved in the proposed system. Since the query is first searched in the AIML, we need to have a mechanism in the AIML to instruct the chat engine, to direct the query to the KB engine for searching the database. To facilitate this, a new AIML construct that can be included in the

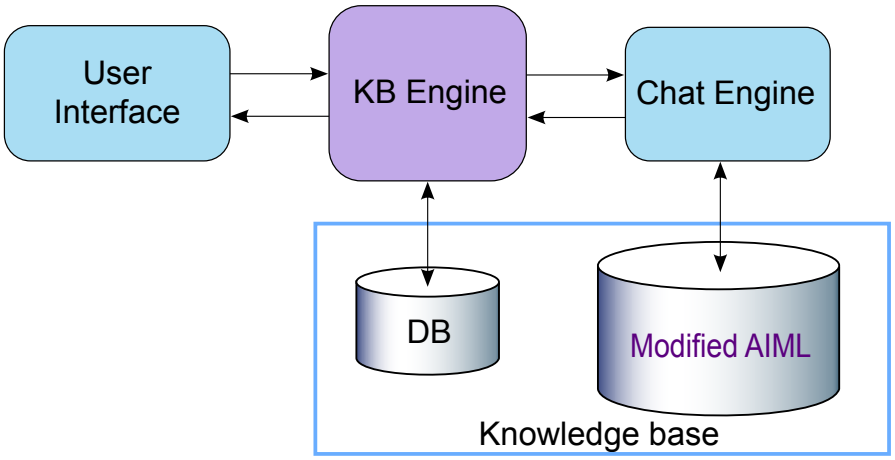


Figure 3.1: Inquisitive chatbot architecture

template tag has been proposed as shown in Fig. 3.2.

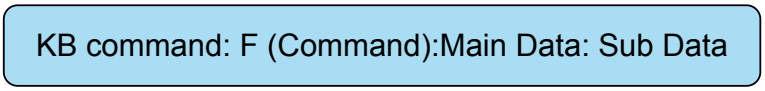


Figure 3.2: KB Engine command template

The KB engine processes this template as it starts with token "KB" which indicates that it is the KB engine's responsibility for further processing. The next token, F (command), contains the command that is mapped with action that has to be executed by the KB engine. The main and sub data required for processing the action is also stored in the template. The KB engine evaluates this template and subsequently execute the appropriate command action. Depending on the type of the command, the Main and Sub data fields varies.

Knowledge Base engine is designed to facilitate the database integration to analyse the missing information, on a first level, that is required to answer a query. KB engine can find the proper response from the database using the information contained in this newly

introduced construct. The KB engine works on a two-phase methodology. First phase identifies the missing data and generates the inquisitive query to probe the user. The second phase processes the collected information from the user to generate response to the user query. The following subsections details both the phases.

3.3.2 Primary Phase – Missing field Identification

The KB engine intercepts the user query and passes it to the chat engine for further processing. The chat engine parses the query and compares the same with AIML templates. If AIML has the matching response then it is passed directly to the user through the KB engine. In case if the answer has to be fetched from the RDBMS, it will be marked in AIML using a template with a specified format derived from the generalised format shown in the Fig. 3.2. On receiving this template, KB engine process it and find missing information in the query using the required information list provided in the AIML template. The specific command format is defined as below.

KB engine command: Function to be acted by KB engine: Table name which contains the field: (Field required to answering the query)

The required field retrieved from the template can have any value from the set of field values stored in the RDBMS table. Let V be the set of stored field values, represented by

$$V = \{v_1, v_2, \cdots, v_n\} \tag{3.2}$$

KB engine compares the user input with the set of available field values.

$$\{f_i\} \cap \{v_1, v_2, \cdots, v_n\} = \emptyset \tag{3.3}$$

where f_i is the user input and $\{v_1, v_2, \cdots, v_n\}$ is the set of available values for the identified field in the system. $\{f_i\}$ should be subset of $\{v_1, v_2, \cdots, v_n\}$. If the intersection of set of f_i and set of V is a null set, then the KB engine generates the query to collect the missing information and retort to user and set a flag to get the input directly to the KB engine. If the intersection of $\{f_i\}$ and V is not null, the KB engine is intelligent enough to identify that there is no missing data and directly provides the answer to the query from the RDBMS, without probing the user.

3.3.3 Secondary Phase – Validation and Response Generation

In case if the chatbot has probed the user in the first phase, the user response retrieved will be processed in the second phase. The response retrieved from the user is directly processed by the KB engine instead of the chat engine. To validate the response of the user, KB engine again compares the retrieved user input with the set of field values as described before. If any matching field value is found, then the KB engine sends the retrieved value and the user's query to the chat engine. The chat engine processes the query and generates an appropriate response to the user's query by means of applying the retrieved value. Flag is also reset to indicate that the upcoming queries needs to be processed by the chat engine. Algorithm and flowchart for the proposed inquisitive chatbot is furnished in Algorithm 3.1 and Fig. 3.3 respectively.

Algorithm 3.1 Inquisitive chatbot processing

Input: User query

Output: Response to query

1: **procedure** INQUISITIVENESS
2: Receive user query
3: Normalise the input with substitution, sentence splitting, word substitution and pattern fitting normalising techniques
4: Search for a matching pattern from AIML KB
5: **if** a direct response is found **then**
6: Redirect response to user through KB engine
7: **else**
8: Tokenise the command template from the AIML
9: Generate query for searching the presence of required field in user query
10: **if** required field is missing **then**
11: Set KB engine to process the queries instead of chat engine
12: Construct the question to probe the user
13: Receive response from the user
14: Validate the retrieved info by comparing with the required field values
15: **if** Matching field value is found **then**
16: Send the retrieved value to chat engine
17: Turn off the direct processing of query by KB engine
18: Regenerate query with the retrieved value
19: **else**
20: Goto Step 12.
21: Find the answer from the knowledge base and construct the response

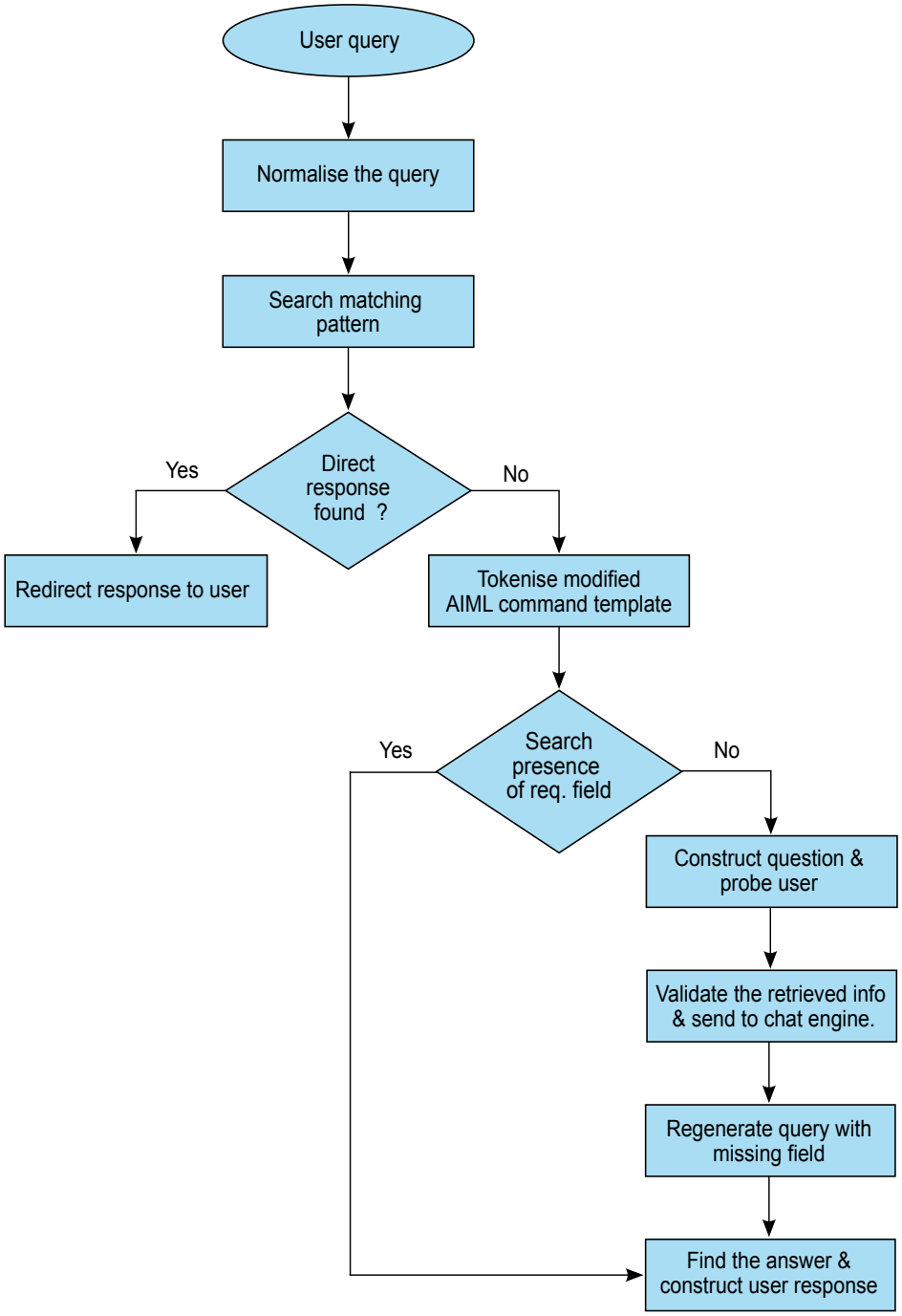


Figure 3.3: Flowchart of inquisitive chatbot

3.3.4 Preprocessing by Word Substitution

It is observed that some words can be entered by users in different forms. As an example, department may be referred to as 'department', 'dept', 'branche' or 'branches' in user query. In the example discussed above, any of the word like dept or branch will be substituted by the word 'DEPARTMENT' to reduce the number of entries in the AIML knowledge base. Like wise, head of the department might referred as HOD.

To address such a scenario, it is normally required to add all combinations of the pattern-template pair to the AIML knowledge base for the same query. This will lead to multiple knowledge base entries for a single query. In order to handle this, the word substitution method has been applied. A look up table is created which contains a list of similar words and the corresponding word to be substituted. Once the default normalisation techniques of AIML interpreter is carried out, as discussed in section 2.5.2, the words in the input will be checked against the look up table entries and will be replaced with the substitution word.

3.4 Case Study of Inquisitive Chatbot

The scenario of the college information desk as discussed in section 3.2.1 has been considered for presenting a case study for an inquisitive chatbot. The first level of inquisitive ability that makes the chatbot much more interactive is implemented by means of a modified AIML chat engine framework and AIML format. On implementation, there is a substantial improvement in the level of interactivity between the user

and the agent.

The user interface of a simple interaction session between a user and the first level inquisitive chatbot is presented in Fig. 3.4. The knowledge base is configured using AIML file with static data of college information and RDBMS is used for storing frequently changing details as the RDBMS can be updated more conveniently than AIML.

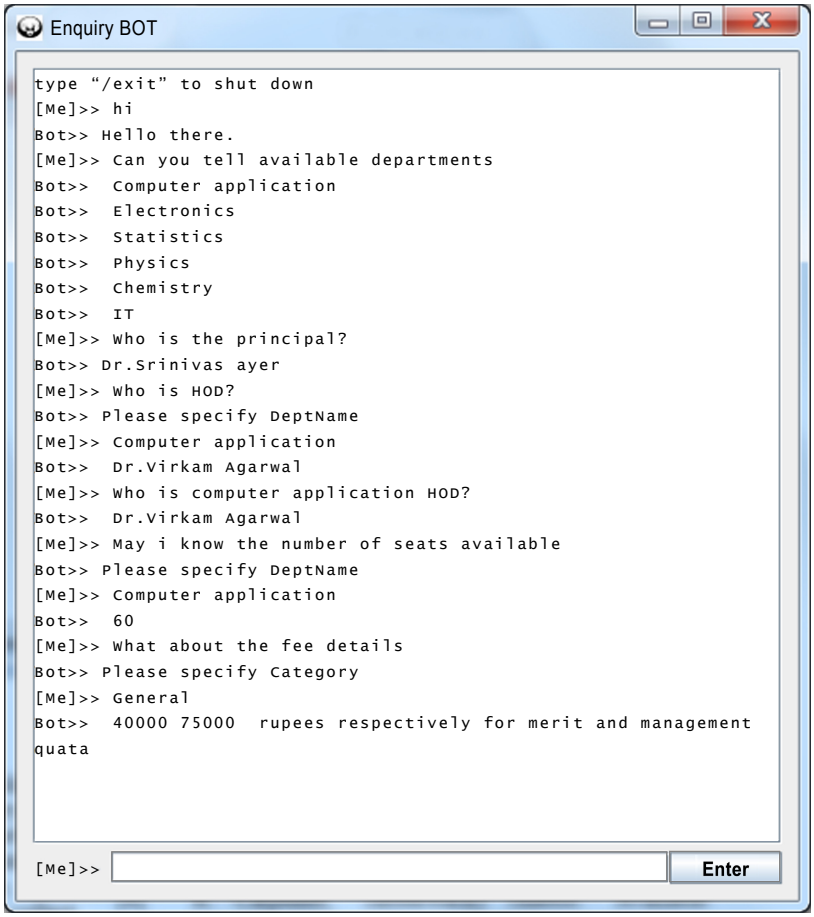


Figure 3.4: User interface of inquisitive chatbot of college information desk

For analysing, consider the below conversation.

User: Who is your principal?

Bot: Dr. Srinivas ayer

User: Who is the HOD?

Bot: Please specify the department

User: Computer application

Bot: Dr. Virkam Agarwal

Here the KB engine has identified that, the name of the department is missing in the query and the same is required to find the name of the HOD. It then probes the user to specify the department name. Upon receiving the department name, KB engine has verified the presence of "Computer application" department in the system. Once the verification process is completed, the chat engine regenerates query with this data and searches in the corresponding database table to obtain the name of the HOD and provides it as the response to the user. The AIML that is compiled for answering this question is provided in Fig. 3.5. The structure of the relevant tables has been furnished in Table 3.1 and Table 3.2.

Table 3.1: Structure of Department Database table

Field name	**Data type**	**Constraints**
DeptID	varchar(10)	Primary key
DeptName	varchar(50)	Not Null
DeptDesc	varchar(250)	Null

Consider another conversation where the department details are provided in the query.

User: Who is computer application HOD?

Bot: Dr. Virkam Agarwal

In this case, the chatbot directly provides the answer to the query

Table 3.2: Structure of Empdetails Database table

Field name	**Data type**	**Constraints**
EmpID	varchar(10)	Primary key
EmpName	varchar(50)	Not Null
EmpQualification	varchar(50)	Not Null
DeptID	varchar(10)	Foreign key
EmpEmail	varchar(25)	Not Null
Designation	varchar(25)	Not Null

without probing the user for providing the details of the department. KB engine is intelligent enough to identify the department details in the user's query.

3.5 Enhancing Inquisitiveness through NER Integration

Natural Language Processing (NLP) help machines to interpret and understand human language. They generally refer to a group of techniques like part-of-speech (POS) tagger, Named Entity Recognizer (NER), parser, coreference resolution system, sentiment analysis, bootstrapped pattern learning etc. NLP tools can be incorporated in different ways to the chatbot and the same depends upon how we intend to use the chatbot, and it is more about providing a human touch to the bot. Most of the chatbot systems use some form of natural language processing to analyse user input, match the same with words and phrases stored in the knowledge base and to provide a proper response based on conversation context. Even though the potential of NLP tools is immense, it is still considered to be in the early stages of development.

college.aiml — Edited

Find Done Replace

```
<!-- - - - - - HOD - - - - - - - -->

<category>
        <pattern>_ WHO * HOD *</pattern>
        <template> <srai>WHOISHOD</srai> </template>
</category>

<category>
        <pattern>WHO * HOD *</pattern>
        <template> <srai>WHOISHOD</srai> </template>
</category>

<category>
        <pattern>WHO * HOD </pattern>
        <template> <srai>WHOISHOD</srai> </template>
</category>

<category>
        <pattern>WHOISHOD</pattern>
        <template>KBHOD :CHECKDB:DEPARTMENT:(DeptName)</template>
</category>

<category>
        <pattern>KBOK *</pattern>
        <that>KBHOD *</that>
        <template>KBGETHOD :GETDETAILS:Select EmpName from empdetails where
designation='HOD' and empdetails.DeptID=(Select DeptID from department where
deptname='<star/>'):1 </template>
</category>

<category>
        <pattern>_ QUALIFICATION _</pattern>
        <that>KBGETHOD *</that>
        <template>KBGETHOD :GETDETAILS:Select EmpName from empdetails where
designation='HOD' and deptname='<star/>':1 </template>
</category>
```

Figure 3.5: AIML for handling HOD related queries

If the chatbot has an ability to extract the entities in a user query, will lead to better response generation. Named entities are names used for the referring persons, organizations, locations, etc. A detailed description of NER, its methodologies and tools are discussed in section 2.7. In the following sections, we are exploring the possibility of NER integration to the chatbot for enhancing the process of inquisitiveness.

3.5.1 Integration of NER with Chatbots

In the proposed model, an NER module has been introduced which can be integrate to KB engine. A detailed block diagram of the proposed system architecture with the main modules involved in it has

been presented in Fig. 3.6. Incorporating the relevant NER features to the chatbot has been achieved through Stanford CoreNLP toolkit integration. The KB engine communicates with the NER module via Stanford JAVANLP API for extracting the presence of the entity. When the user enters a query, the KB engine passes it to the chat engine for processing. The chat engine parses the query and compares the same with AIML templates. In case if the response needs to check the entity presence, then it will be marked in AIML using the specified format given below.

KBcommand: NLPProcess: EntityExtraction: List of required entity classifier

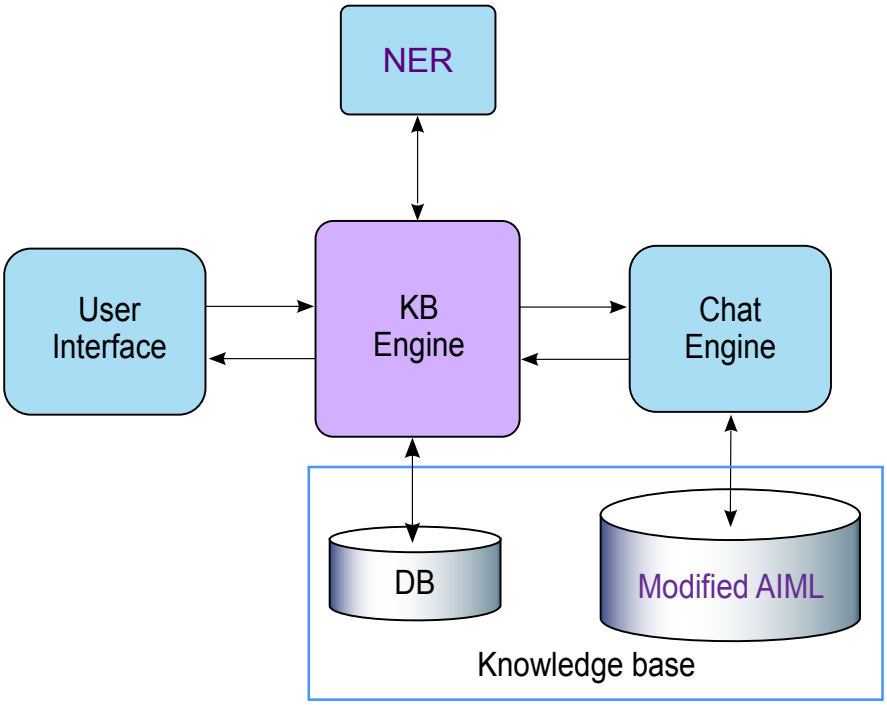


Figure 3.6: NER integrated chatbot architecture

Once the 'NLPProcess' command is received by the KB engine, it identifies that this query requires further NLP processing. The next token specifies which of the NLP operations needs to be carried out. In this case, the token 'EntityExtraction', indicates that the query requires the processing of NER functionality out of other NLP functionalities. The KB engine sends the user query to the NER module and the NER module provides a list of named entity tags that are identified for each token in the user's query. KB engine compares the required entity list extracted from AIML template with the named entity tags received from the NER module. Once the KB engine identifies the unavailability of any of the required entity tag, the engine probes the user for collecting the missing information.

3.5.2 Case Study of NER integrated Chatbot

NER technique makes sure that the given input is parsed properly, disseminated to smaller pieces and classified under proper categories. Thus, the KB engine can find the missing entity through entity extraction and can probe the user for additional data when it finds the input as ambiguous or insufficient. To illustrate its use in inquisitiveness, let us take the case of the college information desk described in section 3.2.1.

Consider a query asked by the student:

User: As I lost certificate, how to apply for duplicate certificates?
Bot: Could you provide the year of graduation and enrollment number
User: 51322065
Bot: Please specify the year

The user's query misses the required information, the roll number and the year of graduation, and the same is required for generating a valid response. Thus the bot requests to provide the required information. Once the user responds, the bot needs to ensure whether the user has entered a number and date entity corresponding to the roll number and year. A system without NER finds it difficult to identify whether the user has provided the roll no and year. However, with the integration of NER, the chatbot can identify whether the user has provided the required details and if not, probe the user again for obtaining correct details.

In the above scenario, a number and date entity are expected and when probed, the user has only entered the roll number and not the year. When the sentence is analysed using NER module no entity corresponding to date is found and the KB engine immediately understand that the year is missing and probes the user. If there are no missing details in the user query, then the KB engine is intelligent enough to identify that the entity is not missed in the user's query and directly attempts to provide the answer to the query, by searching in AIML. Thus, NER integration has empowered the chatbot to ask very specific and vital information that is required to provide correct responses. A chat session between a user and the NER enabled inquisitive chatbot is shown in Fig. 3.7.

3.6 Evaluation of the Inquisitive Chatbot

Evaluation is an important dimension which would help in the process of assessing, comparing and ranking the performance of conversational agents. As already discussed in section 2.6, literatures

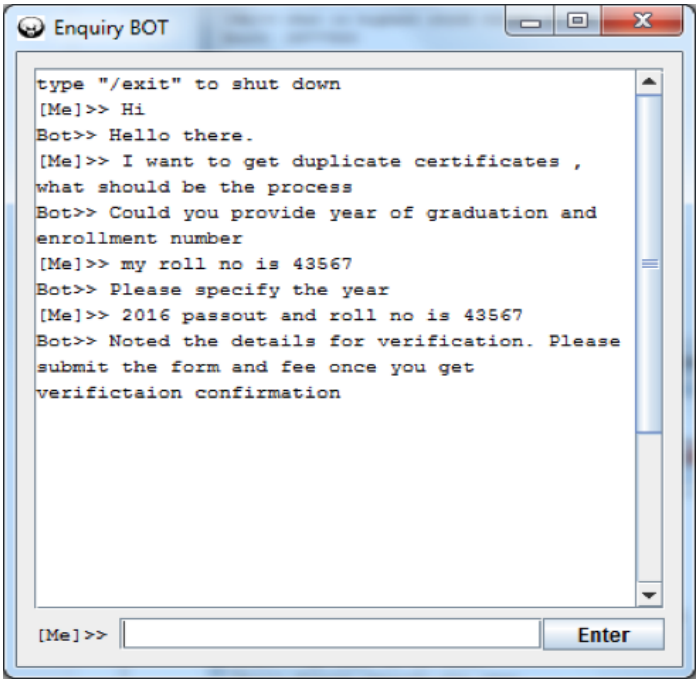


Figure 3.7: User interface of NER integrated chatbot

on evaluation of chatbots are relatively sparse given its state of importance and are mostly available in the form of evaluating general natural language systems.

As there is no single standard methodology to evaluate different aspects of conversational agents, we are evaluating the inquisitive chatbot using different evaluation metrics as followed in [102] and the following sections discuss the applied evaluation metrics and the results.

3.6.1 Measure of Success Response Rate

To evaluate the effectiveness of carrying out conversation with an inquisitive chatbot, a test script has been prepared. The test script

contains commonly asked 150 queries with and without missing information, based on case study described in section 3.2.1. This test script is configured with 75 direct queries which are not having any missing information and 75 queries with missing information (indirect queries). The system is then evaluated by executing the test script and logging the responses obtained. By analysing the response log generated, correctness of the responses are identified and the evaluation metrics are computed.

Test script was ran on an inquisitive chatbot without NER support and with an NER integrated inquisitive chatbot. Success and failure responses are counted and are tabulated in Table 3.3 and Table 3.4 for inquisitive chatbot without and with NER integration respectively. 60 direct queries were answered by both chatbots. In the case of indirect queries, while the inquisitive chatbot without NER could answer only 23 queries, the NER integrated chatbot performed better by responding to 37 indirect queries. The results are in plotted in Fig. 3.8. While the inquisitive chatbot without NER integration reported an overall success rate of 55%, the chatbot with NER integration resulted in 64% success rate.

Table 3.3: Success rate of Inquisitive chatbot (without NER support)

Query Type	**No. of Queries**	**Success Responses**	**Failure Responses**	**Success Rate**
Direct	75	60	15	80%
Missing Info	75	23	52	30.66%
Total	150	83	67	55.33%

Table 3.4: Success rate of NER integrated inquisitive chatbot

Query Type	No. of Queries	Success Responses	Failure Responses	Success Rate
Direct	75	60	15	80%
Missing Info	75	37	38	49.33%
Total	150	97	53	64.66%

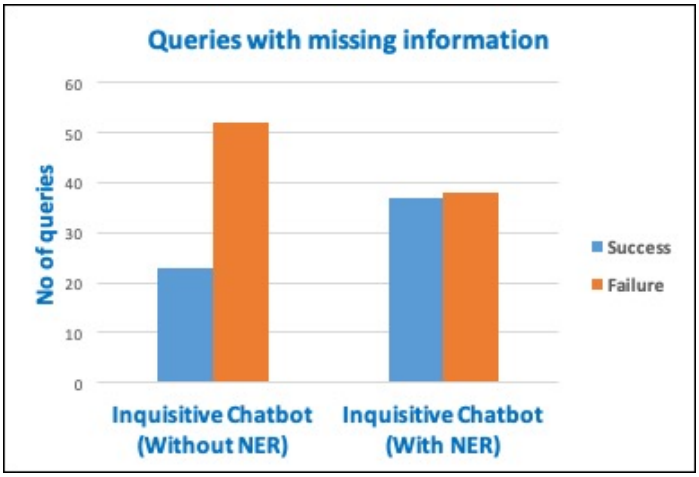


Figure 3.8: Success and failure responses of indirect queries

3.6.2 Evaluation using Performance Metrics

The performance of the chatbots has been evaluated with different accuracy metrics namely precision, recall, accuracy and F1-score [112]. Precision measures the accuracy of the system and recall finds its exhaustiveness. F1-score is computed by combining recall and precision. These metrics can be calculated by the following equations [102].

$$Precision(P) = PC/(PC + NC) \tag{3.4}$$

$$Recall(R) = PC/(PC + PI) \tag{3.5}$$

$$F1Score = (2 * P * R)/(P + R) \tag{3.6}$$

$$Accuracy = (PC + NI)/(PC + NC + PI + NI) \tag{3.7}$$

where

PC = Correct response produced

NC = Unexpected response

NI = Absence of response

PI = Incorrect response produced

From the analysis of the response log, after running the test script described in previous section, the performance metrics namely precision, recall, F1-score and accuracy for different chatbot configurations are tabulated in Table 3.5. For the basic chatbot configuration without inquisitiveness the precision and recall are 0.69 and 0.67 respectively. With inquisitiveness these measure have increased to 0.76 and 0.79 respectively. With incorporation of inquisitive algorithm with NER the results have further increased to 0.84 and 0.83 respectively. From the comparison results of these chatbot configurations, one can observe that inquisitive chatbot with NER outperforms others with an accuracy of 75%.

Table 3.5: Chatbot evaluation metrics

Algorithm	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Basic (without Inquisitiveness)	0.6976	0.6741	0.6857	0.6333
Inquisitive without NER	0.7614	0.7904	0.7757	0.6800
Inquisitive with NER	0.8434	0.8362	0.8398	0.7533

3.7 Summary

Proposed an inquisitive chatbot with an additional KB engine which is able to communicate more interactively with users. This chatbot is able to identify the missing information in a query and the inquisitive nature of the bot enables it to collect additional data that is required to answer the query. Since the chatbot can collect the required information from the user interactively, the users feel that they are communicating with a human and hence user satisfaction improves. To further improve the performance of the inquisitive chatbot, Named Entity Recognition was applied to identify the named entities. A case study has also been conducted along with its evaluation by a combination of different metrics.

Chapter 4

Ontology based Inquisitive Chatbot

4.1 Introduction

Inquisitiveness enables the chatbot to probe the user meaningfully for gaining further insights about the user's query so that it can be answered efficiently. Chapter 3 discussed design of the first level inquisitive chatbot where the chatbot can find out a single piece of missing information. There may be instances where the response generation requires multiple information, and the same may be absent in the query. This chapter discusses the design of a multilevel inquisitive chatbot that can handle multiple missing information by suitably probing the user. The design proposes an ontology based approach for multilevel inquisitiveness.

4.2 Ontology

Ontology in general denotes an artifact that is designed to enable the modeling of knowledge about specific domains. They enable a structured knowledge base of a domain which can be interpreted syntactically and semantically by machines. An ontology defines a set of representational primitives to model a domain of knowledge or discourse [130]. The representational primitives are typically classes (or sets), attributes (or properties) and relationships (or relations among class members). An ontology together with a set of individual instances of classes constitutes a knowledge base [131]. Classes are the main focus of the ontologies which describe concepts in the domain. Concepts which are more specific than the parent classes are represented by subclasses. Practically, developing an ontology includes defining classes for concepts, designing the class hierarchy, identifying the properties among the classes and binding the properties to appropriate classes [132]. The knowledge base is created by defining individual instances of these classes.

4.2.1 Description Logic

Description Logic (DL) provides a logical formalism for ontologies and the semantic web. It is one of the formal knowledge representation languages which describes domain in terms of concepts, individuals, roles and their relationships [133]. The description logic is a subset of the first-order predication logic [134]. The same is used in artificial intelligence to describe and reason the relevant concepts of an application domain.

A DL knowledge base K, is a pair $< T, A >$, where T is a set of "terminological" axioms (TBox) and A is a set of "assertional" axioms (ABox) [134]. TBox axioms define terminology (schema) of an application domain and ABox use the terminology (data). The terminology consists of concepts which denote set of individuals, roles and the relationships between individuals.

Other than storing terminologies and assertions, description logic also provides services that can reason about them. DL reasoners check satisfiability of descriptions and consistency of assertions which verifies the logical contradictions in the ontology axioms. DL reasoners, in addition to this, can derive inferences from the asserted information.

4.2.2 Ontology Reasoner

Reasoner is a program that provides automated support for different reasoning tasks like classification, debugging and querying [135]. This is made possible by deriving logical consequences from a set of asserted facts which is expressed explicitly. The reasoners can ensure quality of ontologies by carefully analysing the underlying inconsistency and uncertainties present in the ontologies [136]. The redundancy of information in the knowledge base can be reduced by means of ontology reasoning. It also helps in finding the conflict in knowledge content. Pellet, RACER, FACT++, Snorocket, CEL, ELK, SWRL-IQ, TrOWL HermiT etc. are some of the most popular reasoners developed and employed in the past few years [137].

HermiT is the first publicly-available OWL reasoner written using Web Ontology Language (OWL). HermiT checks the given OWL files and can determine the consistency of ontologies. It also identifies the

subsumption relationships between concepts and other features [138]. This reasoner is based upon hypertableau calculus which aids in classifying ontologies in a faster and efficient manner [139].

4.2.3 Ontology Editors

Ontology editors are tools that enable ontology development, visual manipulation, inspection and maintenance tasks [140]. Apollo, Protégé, NeOn Toolkit, OntoStudio, TopBraid Composer and Swoop are some of the most commonly used editors [140].

Protégé is an open-source ontology editor developed by Stanford University [141]. It provides a suite of tools to create domain models and knowledge based applications with ontologies. By acting upon knowledge modeling structures, it facilitates the creation, visualisation and manipulation of ontologies in different representation formats. It helps to define classes, hierarchy between classes, variables, variable value restrictions and the relationship between various classes and properties of these relationships [142]. Scalability and extensibility are the prominent advantages of protégé. It can be used to construct and process large ontologies efficiently [143] and can store them in various formats like UML, relational databases, Resource Description Framework (RDF) and XML [144].

4.3 Multilevel Inquisitive Chatbot

Inquisitiveness or the ability to identify the missing information that is required for answering a query is a very desirable characteristic of a conversational agent; and has been detailed in chapter 3.

Chapter 3 also discusses the design of an inquisitive chatbot; using RDBMS and AIML. The proposed chatbot was able to handle first level of inquisitiveness. However, it was unable to handle the queries with more than one missing information. Multilevel inquisitiveness (MLI) is the ability to find multiple information that is missing from the user's query and to probe the user further to collect the same.

This chapter proposes an ontology based chatbot which has the ability to carry out conversation with the user, with multilevel inquisitiveness. The features of the ontology like the conceptual structure of the domain knowledge and the ability to analyse the domain knowledge to derive assumptions have been utilised to accomplish this.

Let us consider a query from the scenario of college information desk which has been discussed in chapter 3.

User: Who is our course coordinator?

The above query misses two pieces of relevant information required for providing an appropriate answer: department name and course name as there can be many course coordinators for different courses in different departments. In complex scenarios, multiple proactive queries may be required to reach the appropriate answers. As an example consider the following query:

User: When will be my exam?

In this case, this multilevel query needs details of Department, Course, Subject, and Semester to provide an answer.

4.3.1 Architecture of Ontology based Inquisitive Chatbot

The design of an ontology integrated multilevel inquisitive chatbot by modifying the AIML engine is described in this section. In order to support MLI, an additional knowledge base, in the form of ontology has been incorporated. To accomplish this, the current system has been extended by introducing a knowledge base engine (KB engine) similar to the approach followed in Chapter 3. The KB engine manages the interaction with the ontology through OWL API. Fig. 4.1 depicts the proposed system architecture with the main components involved in it. The original AIML KB has been modified to provide support for the new functionalities.

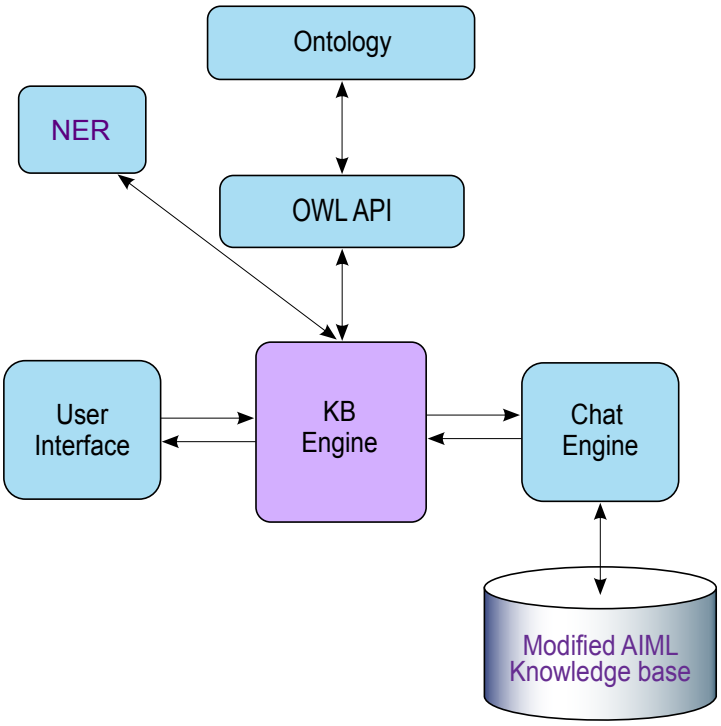


Figure 4.1: Proposed ontology based chatbot architecture

A new template format following the generic command template specified in Fig. 3.2 has been defined for specifically fetching the data from the ontology as given in Fig. 4.2. Whenever the "ReadOWL" command is received in KB engine, it tokenizes the command and finds the missing information in the user's query using the required information list ("Info list") provided in the AIML template.

Figure 4.2: KB Engine OWL command structure

Missing information is identified by analysing the items in the 'info list' with the corresponding instances of the particular class of that field in the ontology. If any of the required field is missing which is required to generate the response of the query, then the KB engine probes the user and collects the missing information. This process is repeated for all the items in the "info list". Once, all the required info list values are collected, KB engine regenerates the DL query with required information collected from the user. By executing the DL query, the data received from the ontology will be used for generating the response for the user's query. The block diagram depicting single missing information finding processes is given in Fig. 4.3.

4.4 Case Study of the Ontology based Inquisitive Chatbot

For the demonstration of the proposed system, an ontology which details the working of a University has been designed. This is

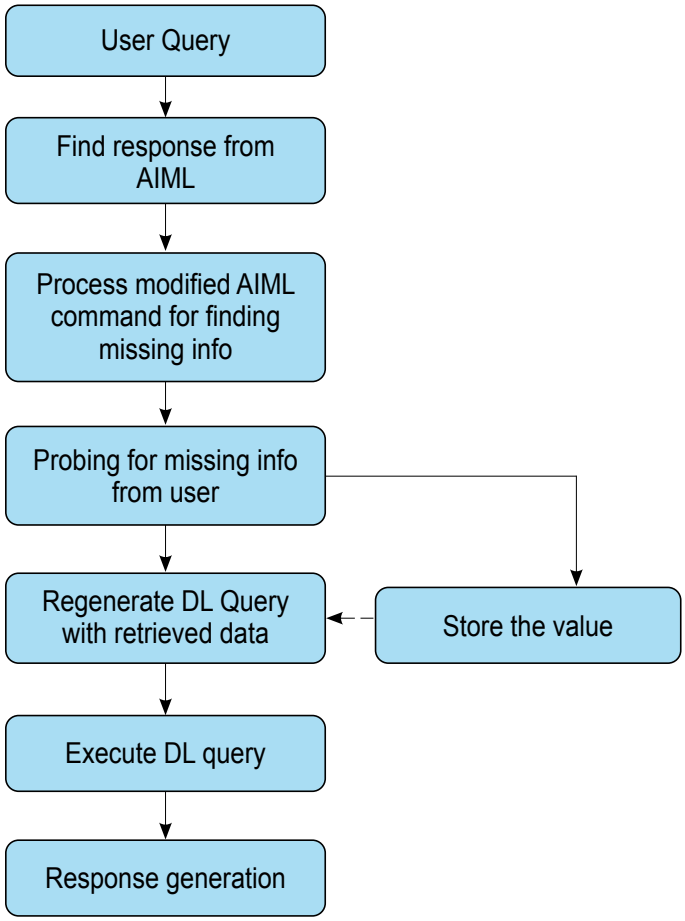


Figure 4.3: Block diagram of ontology based chatbot process

implemented through the Protégé editor [144] and the output has been saved in OWL format. Protégé was used to create the class hierarchy of university classes and also to define Objects, Inverse, Data and Annotation properties for the university classes and to map the properties to appropriate classes. The validity of the ontology design was confirmed by matching inferred and asserted ontology. The consistency of the ontology has been checked using HermiT reasoner [138], which can determine whether or not the ontology is

consistent and can identify relationships between the classes. The ontology graph with some classes and their object properties are shown in Fig. 4.4 and the corresponding ontology graph with instances is given in Fig. 4.5.

To illustrate the working of the MLI chatbot, consider the sequence of queries.

User: Who is our course coordinator?
Bot: Please specify department
User: Electronics
Bot: Please specify the course
User: M. Sc electronics
Bot: Your course coordinator is Madhav Ram

Here the initial query raised by the user does not contain the required information about the department and course which are necessary for providing the response. KB engine identified the missing fields and probed the user for collecting relevant information. If the query does not miss the department or course details, then the KB engine is intelligent enough to directly provide the answer to the query.

4.4.1 Evaluation of Multilevel Inquisitive Chatbot

4.4.1.1 Success Response Rate

Multilevel inquisitive chatbot is evaluated with a test script as discussed in section 3.6.1 by counting both success and failure responses. A total of 150 queries were used for the evaluation and the success and failure response counts are tabulated in Table 4.1. The

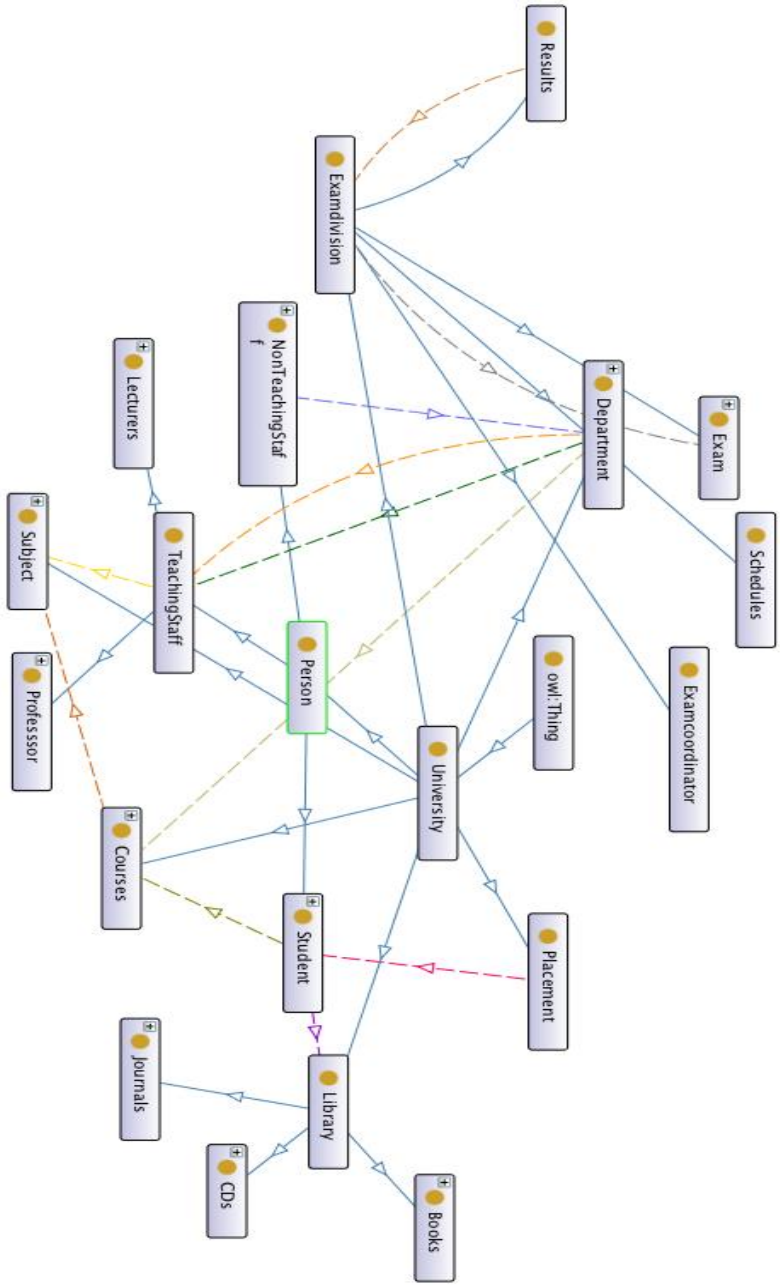


Figure 4.4: Sample of University ontology graph

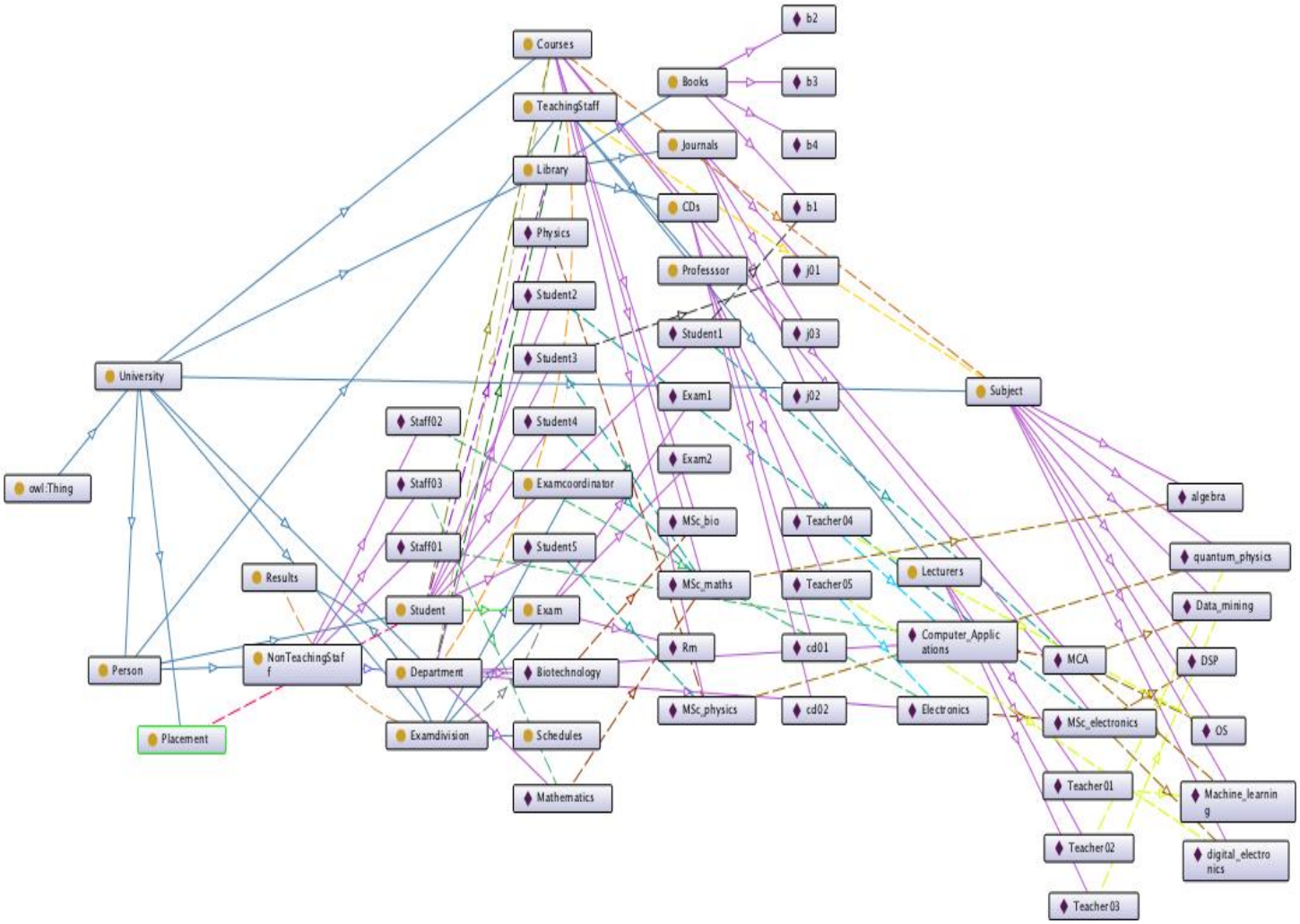


Figure 4.5: Sample of knowledge graph of University with instances

multilevel chatbot responded successfully for 60 direct queries and 56 indirect queries with missing information resulting in an overall success rate of 77%. Comparing this result with the NER integrated chatbot (Table 3.4) with 64% success rate, we can observe a considerable improvement in the success rate.

Table 4.1: Multilevel inquisitive chatbot

Query Type	**No. of Queries**	**Success Responses**	**Failure Responses**	**Success Rate**
Direct	75	60	15	80.00%
Indirect	75	56	19	74.66%
Total	150	116	34	77.33%

4.4.1.2 Performance Metrics Evaluation

The precision, recall, F1-score and accuracy for multilevel chatbot has also been computed and is tabulated in Table 4.2. A comparison of these metrics with that of NER inquisitive chatbot is illustrated in Fig 4.6. We can observe that the performance has improved with the incorporation of multilevel inquisitiveness using ontology.

Table 4.2: Performance metrics of multilevel inquisitive chatbot

Metrics	**Multilevel Inquisitive Chatbot**	**Inquisitive Chatbot with NER**
Precision	0.8787	0.8434
Recall	0.8992	0.8362
F1 Score	0.8888	0.8398
Accuracy	0.8066	0.7533

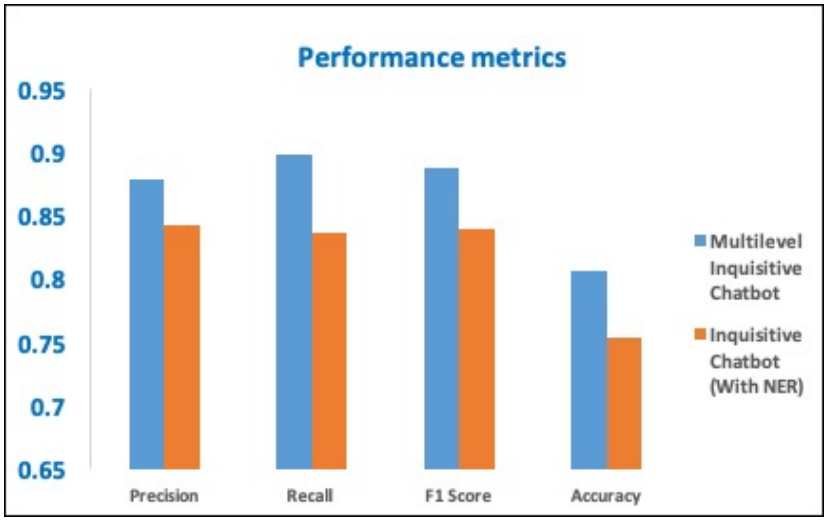


Figure 4.6: Comparison of performance metrics

4.5 Summary

Chatbots can be coupled with ontology to enhance multi level inquisitiveness and in turn enhance the way the human-computer interactions takes place. The methodology adopted for the design of an ontology based multilevel inquisitive chatbot has been discussed in this chapter. An overview of the concepts of ontology along with description logic, reasoners and editors have been presented. Since the chatbot can collect the required multiple missing information from the user, the interactive nature of the chatbot is increased substantially. A case study of the proposed chatbot along with the results of the evaluations carried out has also been presented.

Chapter 5

Hybrid Knowledge based Chatbot

5.1 Introduction

Chatbots can respond to a set of predefined queries which are generally stored in a knowledge base. The ability of the chatbot to respond to the variants of the same query makes it more efficient and attractive and this ability, to a large extend, is determined by the algorithms used for pattern matching. Another important factor is the amount of information made available to the chatbot, which helps to frame an accurate response to the queries and effectiveness of such systems is determined by the implementation of the knowledge base. It may also be noted that such information may be dynamic in nature and the knowledge base should be updated regularly to reflect the updated information. As the knowledge base available to the chatbot

become more comprehensive, the number of queries to which the chatbot can respond also increases, making it more efficient and usable.

Generally, the knowledge base is implemented using text files, AIML files or using relational databases [2]. Even though text files are one of the easiest methods to implement the knowledge base, it does not provide any inbuilt mechanism for easy pattern matching. With relational databases, one can store data in a structured and organised way. However, it may not be a suitable option for storing generic query-response patterns. Thus, the usage of AIML for implementing the knowledge base has gained focus. Even though AIML is very easy to use and learn, it is difficult to incorporate dynamic changes in data. Also, only a limited amount of data, using simple category-pattern format, can be stored in AIML files.

In this chapter, we propose the integration of big data to chatbots. Such an integration acts as an external knowledge base and it enables the chatbot to gain knowledge by analysing a large amount of data from a distributed environment.

The remaining part of this chapter is organized as follows. Section 5.2 summaries the big data analytics and Hadoop framework. Section 5.4 demonstrates the case study of big data integrated chatbot capable of fetching data from the distributed environment. An inquisitive chatbot with hybrid knowledge base is proposed in section 5.5 and we conclude this chapter in section 5.6.

5.2 Big Data Analytics

Big data, the vast amount of data, generated due to digitalization has a tremendous impact on business processes. According to business research studies, big data might help companies to make better strategic decisions, understand customer needs, efficiently control processes and reduce costs [145]. Big data analytics is the process of collecting, organizing and analysing large data sets to discover patterns and unknown correlations hidden in the data, such as usage statistics and customer preferences, which can serve as valuable business information. Big data analytics enables the analysis of a combination of structured, semi-structured and unstructured data.

The specific attributes that define big data are known as four V's: volume, variety, velocity and veracity [146]. Big data is generally a large volume of data that has to be analysed. Variety denotes disparate form and source of the data. Big data can be structured, unstructured or semistructured. Velocity refers to the frequency of new incoming data. Uncertainty and trustworthiness of the data are indicated by veracity.

Big data is a term for data sets that are so large or complex, where the traditional techniques of data processing may turn out to be inadequate. It is generated on a very large scale from social media websites, multimedia sources and other forms of network related data along with real-time data generated from sensors and devices. Analysis of such large chunks of data requires the use of specialised platforms like Hadoop [147], Spark [148] etc.

5.2.1 Hadoop Ecosystem

The analysis of big data sets in real-time requires a platform like Hadoop to store large data sets across a distributed cluster and process these data from multiple sources. Hadoop is an open source software framework for storing big data and comprises of a set of tools for processing very large data sets in a distributed computing environment [149]. It provides a massive storage facility for any kind of data, and it possesses enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs [147]. Hadoop is not a conventional type of database, rather a software ecosystem that allows for massive parallel computing.

The core modules of Hadoop comprises of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce, along with a resource management platform called YARN [147]. HDFS is a distributed file system that stores data across multiple machines without prior organization. MapReduce is a programming model that supports parallel processing of large scale data. Such a process normally takes intensive data processing and the computation spreads across a potentially endless number of servers. YARN (Yet Another Resource Negotiator) is the resource management platform for scheduling and managing resource requests from distributed applications.

There are also supplementary tools like Pig, Hive, Sqoop, Avro etc., which have been developed on top of the Hadoop framework and can act as data access frameworks. One of these tools, Hive, is a data warehouse software which facilitates easy data management of large

datasets residing in distributed storage and provides data summarization and ad hoc querying. Hive employs a declarative SQLish language called HQL (Hive Query Language) which is very similar to SQL, for querying the data. The queries in HQL are translated into MapReduce programs by Hive and are executed in runtime [150]. Fig. 5.1 illustrates the Hadoop framework and its associated tools [151].

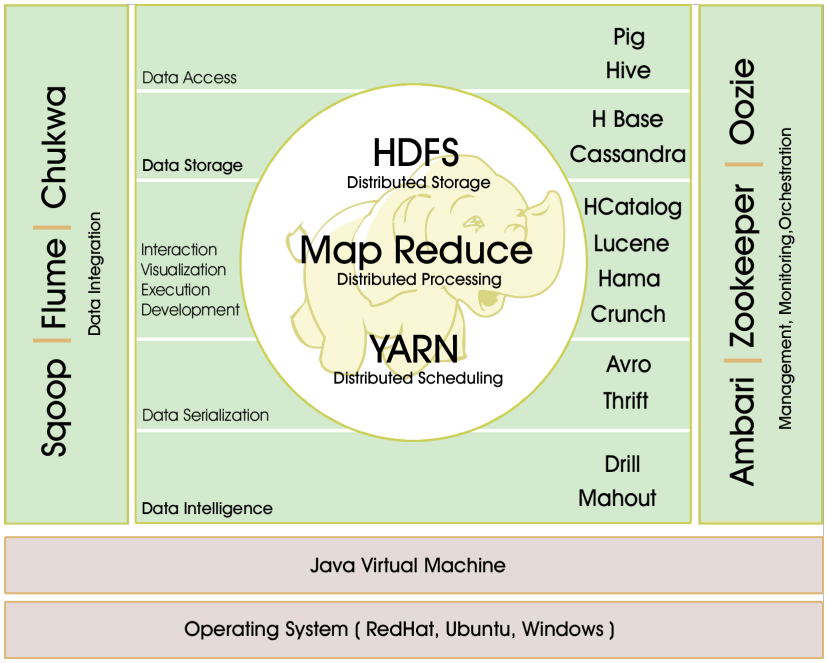


Figure 5.1: Hadoop framework and tools

Being a storage and table management layer for Hadoop, HCatalog holds the metadata and location of the data in a Hadoop cluster [152]. This allows scripts and MapReduce jobs to be decoupled from data location and metadata, enabling different data processing tools to read and write data effortlessly on a grid.

The Hadoop-MapReduce framework is gaining importance rapidly

and is becoming popular for big data processing. Scalability and fail over properties of Hadoop-MapReduce are the main reasons that have attributed to such a popularity [153]. The framework is also known for its ease of use, which allows even non-expert users to easily run analytical tasks over big data.

5.2.2 Hortonworks Data Platform

Various IT vendors such as MapR, Cloudera, IBM and Hortonworks have developed their Hadoop distribution frameworks, that provides various services for managing and processing big data. The Hortonworks Data Platform (HDP) is a single node implementation of enterprise-ready open source Apache Hadoop distribution based on a centralized architecture (YARN) along with a virtual environment that can run in the cloud or on personal machines [154]. The HDP can serve as a straightforward, pre-configured, learning environment to make evaluation and experimentation fast and easy.

HDP can handle big data storage, querying and processing, and also provides services for monitoring, management and data integration [155]. Its one of the key platforms as it provides open source tools and supports for connecting to some Business Intelligence platforms. HDP allows MapReduce based distributed data processing, data querying through Hue, the script running using Pig or Hive and metadata services by Hcatalog.

5.3 Big Data Integrated Chatbot

In the existing AIML framework, the chat engine identifies a suitable response for the user's query using pattern matching algorithms with the help of the knowledge base, which stores set of predefined queries and its variants. In the proposed hybrid knowledge base model, the response for the user's query can come either from the AIML or from the big data knowledge base. While the responses which are more permanent in nature can be stored in the AIML, those responses which need some sort of analysis or which are dynamic in nature can be fetched from the big data.

To achieve this, one knowledge base engine (KB engine) has been configured to interface with the big data framework for fetching factual data for responding to certain queries. Since the incoming query is first searched in the AIML, a suitable mechanism is needed in the AIML to instruct the chat engine, to redirect the query to the KB engine. Hence a modified AIML construct have been implemented to incorporate such KB engine processing. The KB engine communicates with the big data knowledge base through hive interface and finds a proper response from the big data using the modified AIML response redirected by the chat engine. Fig. 5.2 illustrates the architecture of the proposed system.

In order to setup a prototype environment, a single node Hadoop data cluster is configured using Hortonworks Data Platform. For sending queries to this big data environment and to retrieve data, the chatbot framework has to establish a connection with the environment. HiveServer2 (HS2) is a server interface that enables remote clients to execute queries against Hive and retrieve the results. HS2 supports

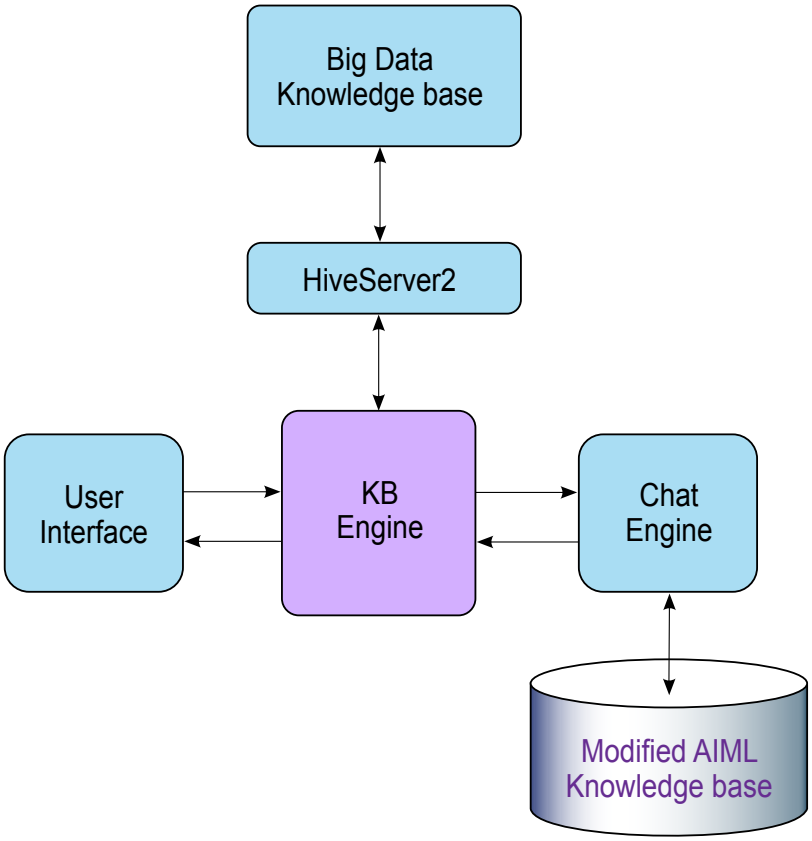


Figure 5.2: Big data integrated chatbot architecture

multi-client concurrency and is designed to provide better support for open API clients like JDBC and ODBC [156]. The chatbot communicates with HS2, over a driver connection using JDBC, which in turn connects with Hive. Hive facilitates easy data management of large datasets residing in the distributed storage. It provides a SQL-like interface to create a hive query to retrieve data from Hadoop [157].

In the proposed chatbot framework, the connectivity to the Hive to access Hadoop data has been implemented through the KB engine.

Thus, the KB engine is designed to integrate the big data as a knowledge base and act as an interpreter between big data and the chat engine. The Hive connectivity API driver enables the access to Hadoop data through KB engine and sends the HQL statements to Hive to fetch information from the big data knowledge base.

As has been previously described, the chat engine processes the user's query by parsing it and comparing it against the AIML templates. If the AIML, readily contains a response, it is passed directly to the user through the KB engine. In case if the response is dynamic in nature and needs to be fetched from big data knowledge base, then it is represented using a modified AIML template.

KB engine command: ReadBigdataKB: HQL query: Number of Fields to process.

The template, following the generic format represented in Fig. 3.2, starts with a token "KB" which indicates that it should be processed by the KB engine.

The process flow of the proposed system is depicted as a sequence diagram in Fig. 5.3. Two scenarios are depicted in the sequence diagram, one for the direct response (Sequence 1 to 3) and the other for the data fetched from the big data knowledge base (Sequence 4 to 11). To fetch information from the big data knowledge base for responding to a user query, the chat engine send the AIML template to the KB engine. The KB engine process the received template and extracts the HQL query and and uses it to fetch the information from the big data; generates a response and sends it to the user .

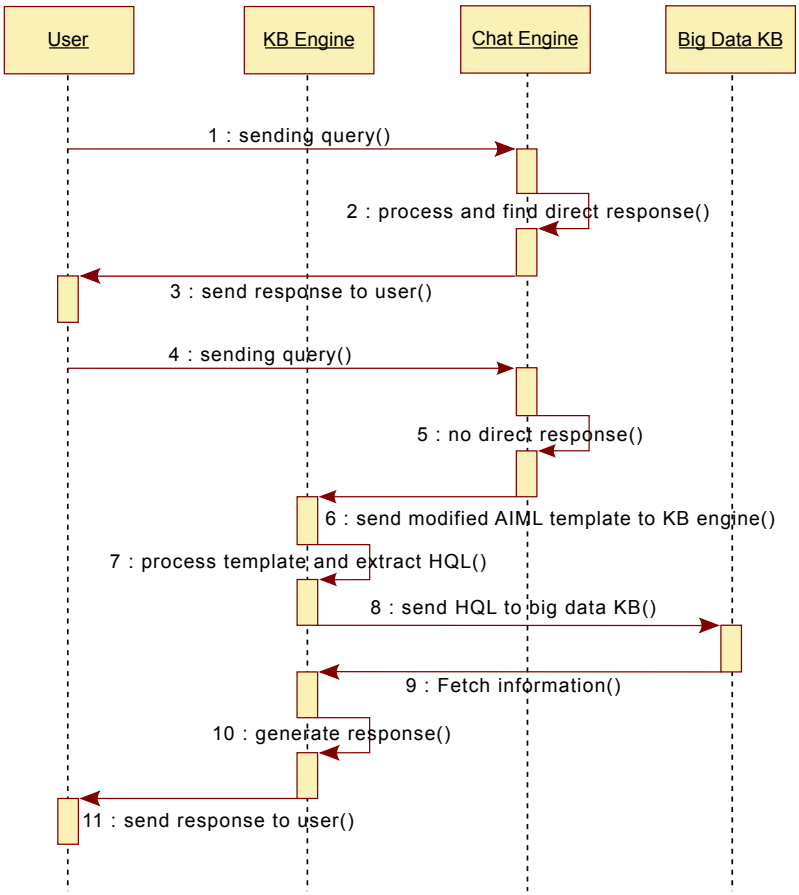


Figure 5.3: Sequence diagram of process flow of big data integrated chatbot

5.4 Case Study of Big Data Integrated Chatbot

It is practically impossible for any layman to disseminate information from the huge chunk of data stored by the big data framework, without knowing big data processing tools and techniques. Gathering of these details from a big data environment requires some software tool for analysing the data and also requires knowledge of

Hive queries or Pig script. Here, as we are using a chatbot to fetch details from big data environment, the user can input their query in natural language and they are not forced to learn the underlying architecture or technologies. The chatbot will communicate to big data knowledge base through Hive servers and obtain details required for response generation and responds to the user.

Assume that a stock market user needs to know the particular details like the average stock volume of a particular company in a specific day or over a period of time. Existing architectures fail in this regard as the AIML framework supports only static query. The proposed architecture helps in addressing this limitation too, by generating dynamic responses with the help of the KB engine by analysing big data.

This case study demonstrates how data analytics tools like Hadoop are applied to make a chatbot respond to a user query that is given in the form of natural language. For demonstration, we took the historical data of the New York Stock Exchange (NYSE) from January 2000 to December 2001 [158]. NYSE is an American stock exchange and it is the world's largest stock exchange. It provides a platform for buyers and sellers to trade shares of stock in companies registered for public trading. NYSE data set is an open data set that can be downloaded and the same data is taken for this case study. The downloaded dataset is comma separated values (CSV) file format and the size of the CSV file in local machine is 44 MB. The dataset contains more than 81 lakh rows of stock details of different companies. Each row in the dataset is composed of following data fields: Exchange name, Company symbol, Date, Stock price, High of the day stock price, Low of the day stock price, Close of the day stock price, Stock volume and Stock price adjusted for dividend payment. Few rows are displayed in Fig. 5.4.

exchange	stock_symbol	date	stock_price_	stock_price_high	stock_price_low	stock_price_close	stock_volume	stock_price_adj_close
NYSE	MED	31/12/01	0.23	0.28	0.21	0.22	72500	0.22
NYSE	EGY	31/12/01	0.51	0.6	0.5	0.55	64100	0.55
NYSE	CQB	31/12/01	0.62	0.65	0.6	0.64	202700	8.71
NYSE	IHR	31/12/01	0.67	0.7	0.67	0.69	384300	0.69
NYSE	DAR	31/12/01	0.66	0.7	0.66	0.7	1500	0.7
NYSE	KGC	31/12/01	0.69	0.76	0.69	0.76	255100	2.26
NYSE	CDE	31/12/01	0.76	0.8	0.75	0.8	66100	8
NYSE	NEU	31/12/01	0.88	0.94	0.87	0.92	166900	0.87
NYSE	HL	31/12/01	0.87	0.96	0.87	0.94	185300	0.94
NYSE	CYD	31/12/01	0.9	0.98	0.88	0.96	31700	0.61
NYSE	HNR	31/12/01	1.37	1.44	1.31	1.44	267300	1.44
NYSE	GRA	31/12/01	1.51	1.55	1.5	1.55	748600	1.55
NYSE	CIB	31/12/01	1.61	1.62	1.55	1.56	4800	1.1
NYSE	WNI	31/12/01	1.75	1.76	1.66	1.68	21800	1.22

Figure 5.4: NYSE dataset sample

To execute the query, the data should be stored in a table and the first step is to create a table to store data in a meaningful manner. The data has been uploaded to HDFS and registered with HCatalog utility to generate table structure so that the location and metadata can be accessed in Hive. We have used Hive query to retrieve data from HDFS and is fed to the chatbot so that they can respond to the user query with the appropriate answer. Fig. 5.5 shows the metadata of the uploaded NYSE data through HCatalog. The column name of the table and data type are displayed in the figure.

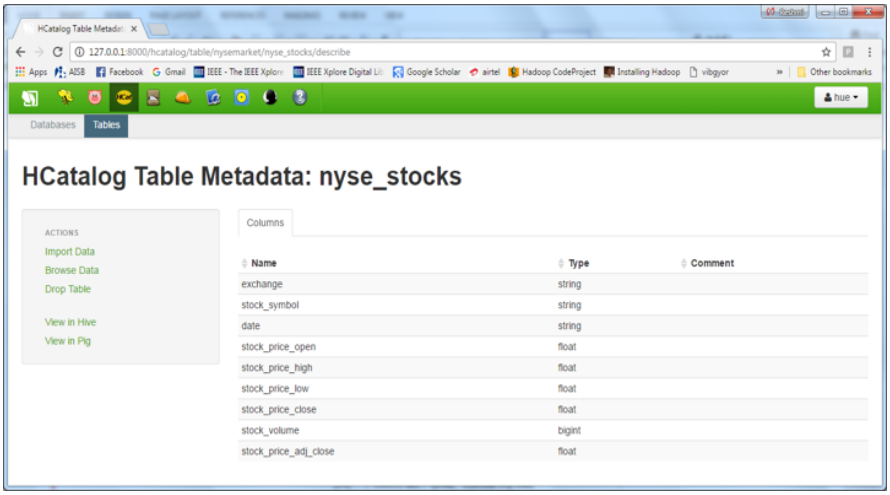


Figure 5.5: Metadata of uploaded data through HCatalog

The user interface of a simple interaction session between a user and the proposed big data integrated chatbot is shown in Fig. 5.6. The user is trying to understand the average stock value of company named IBN and highest stock volume of IBM company. It may be practically difficult for any legacy system to provide an accurate response to such queries through chat. Here the chatbot with the support of Hadoop and Hive has responded with an accurate answer.

5.4.1 Evaluation of Big Data Integrated Chatbot

Big data integrated chatbot is evaluated with the test script prepared by including direct queries which can fetch the response directly from the AIML pattern-response pair and those queries whose response generation require data from the big data environment (Big data queries).

A total of 100 queries were used for the evaluation and the success and failure response counts are tabulated in Table 5.1. The Big data integrated chatbot responded successfully for 36 direct queries and 28 big data queries resulting in an overall success rate of 64%.

The performance metrics of the big data integrated chatbot has also been computed and the metrics are summarised in Table 5.2. The proposed chatbot integrated with big data gives an F1 score of 0.83 and accuracy 0.74, for the given test case.

Table 5.1: Big data integrated chatbot

Query Type	**No. of Queries**	**Success Responses**	**Failure Responses**	**Success Rate**
Direct	50	36	14	72%
Big data	50	28	22	56%
Total	100	64	36	64%

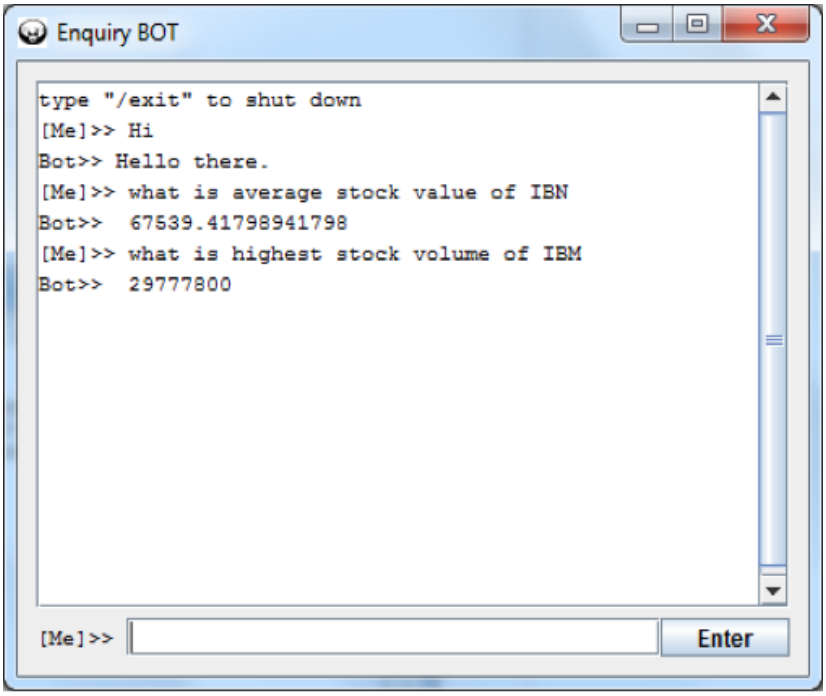


Figure 5.6: User interface of Big Data integrated chatbot

5.5 Integrated System Architecture

An integrated architecture for chatbots with different components discussed so far, has been proposed as illustrated in Fig. 5.7. The architecture constitutes a hybrid knowledge base which includes AIML, RDBMS and big data to handle static, dynamic and distributed information. The architecture also support inquisitiveness enhanced with NER. This integrated inquisitive chatbot has the following features. .

- Identify the missing information in query
- Probe the user for collecting required missing information
- Recognize the named entities of query through NER

Table 5.2: Performance metrics of big data integrated chatbot

Metrics	Big data Integrated Chatbot
Precision	0.8205
Recall	0.8421
F1 Score	0.8311
Accuracy	0.7400

- Dynamic data fetch from database
- Data analytical capability using big data integration

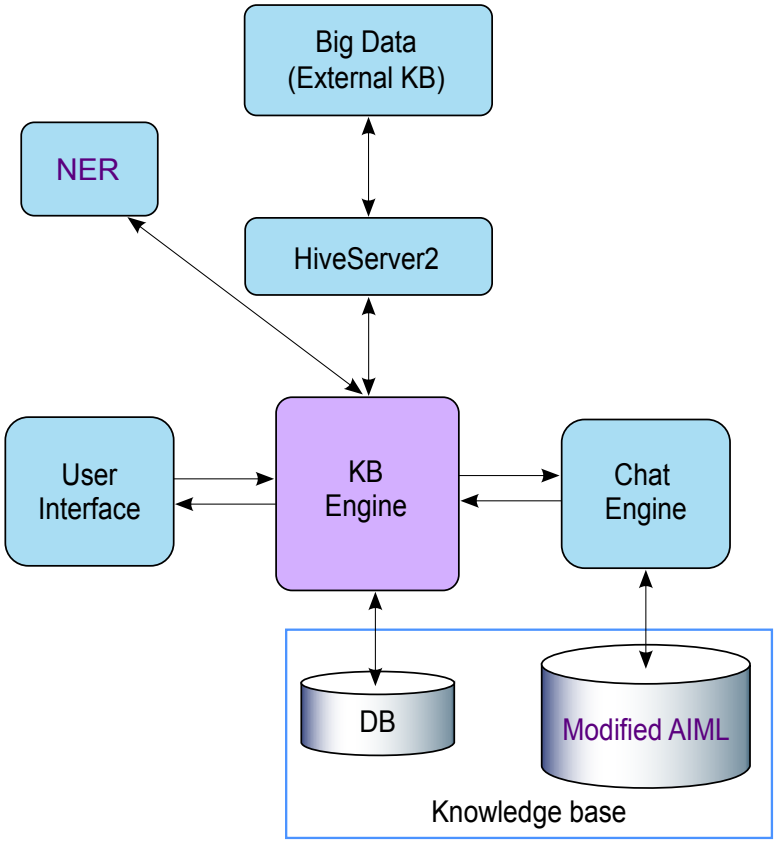


Figure 5.7: Integrated system architecture

5.6 Summary

The integration of big data as knowledge base into the chatbots can enable the generation of dynamic responses to user queries and can improve the analytical capability of the chatbots by handling data from distributed environments. By integrating big data to chatbots, even non-expert users without the knowledge of big data framework and HQL queries can fetch information through simple chats from large volumes of data from distributed ecosystems, which is not easily accessed by traditional chatbot applications. A case study to demonstrate the application of big data has also been presented with stock market data; along with the results of the evaluation carried out. The architecture of an integrated chatbot with hybrid knowledge bases has also been presented in this chapter.

Chapter 6

Language Localised Chatbot

6.1 Introduction

The English language is referred to as the universal language as it is the most commonly spoken language all over the world. Because of this popularity, most of the chatbots interact with users in English. Even though English is one of the official languages in India [159], only about 20% of the Indian population are English literate. Indians largely communicate in the regional languages of their State and not every Indian is comfortable using English. Thus, language localised chatbots in regional languages are essential to improve human-computer interactions as they can cater to the non-English speaking population of India.

Kerala is the only State in India with 100% literacy rate. As per the

Annual Status of Education Report 2014 [160], Kerala is the only state in the country where at least one person in 49% of families is computer literate, where the national average is only 22%. This indicates that a very good percentage of Keralites are using computers and computer enabled services on a daily basis. Thus the significance of a chatbot that can communicate in Malayalam is quite obvious. Chatbots that can converse in Malayalam will help those people who find it difficult to express themselves in English or in any other foreign languages. It, in turn, makes the user feel at ease and will result in repeated usage of the platform. Further, it can bridge the digital gap between the common man and the computer savvy people by providing them with an easy interface to interact with various information systems in their language.

In this chapter, the initial section discusses the peculiarities of the Malayalam language and its script. The next section discourses the implementation of the language localised Malayalam chatbot and its knowledge base composition. Section 6.5 explains the difficulties in creating a Malayalam knowledge base manually and the development of a Malayalam knowledge base synthesiser. Subsequently, a Malayalam chatbot application is considered as a case study and the same is demonstrated in section 6.4. We conclude the chapter in section 6.7.

6.2 The Malayalam Language

Malayalam is one among the twenty-two scheduled languages in India and is the principal language of the South Indian State of Kerala and the Union Territories of Lakshadweep and Puducherry. It is one of the major Dravidian Languages, spoken by more than 35 million people, and has the honour of being declared as a classical language by

the Government of India in 2013 [161]. It shares a strong relationship with Tamil and has got a significant influence from Sanskrit. The current Malayalam script is based on the Vatteluttu script, which was extended with Grantha script letters. As Malayalam speakers are peripatetic, the language is heard widely all over India as well as in the Middle East countries, United States, Europe and Australia [162] and this uniqueness enhance the significance of developing a chatbot that can communicate in Malayalam.

The Malayalam language is supposed to be very context sensitive and it is regarded as one of the toughest language in India. The agglutinative nature of Malayalam language makes it, even more, tougher to act upon for any information processing system. It can also be noted that there is enormous vocabulary growth in the Malayalam language as a large number of different word forms is derived for one word. It does not have case distinction and is written from left to write. The alphabet consists of 16 vowels (*Swarakshara*) and 36 consonants (*Vyanjanam*). It has got two additional diacritic character *viz. anusvara* and *visarga* [163].

6.3 The Proposed Malayalam Chatbot

As already discussed, the majority of intelligent conversational agents communicate with the users in English, sidelining the section of the population that are not comfortable to communicate in English. However, there have been some recent efforts to develop bots that can communicate with the users in their native languages, as discussed in chapter 2. Development of such chatbots that can interact with the users in the native language is very important, as it leads to seamless and more effective human-computer interactions.

A chatbot that can converse in Malayalam with the users employing a suitable chat interface has been proposed and has a retrieval based architecture that can cater closed domain queries. This chatbot model retrieves the response from a knowledge base of predefined responses and selects a proper response through a heuristic pattern matching algorithm. This language localised chatbot accepts input from users in Malayalam and also responds in Malayalam. Fig. 6.1 depicts the block diagram of the proposed Malayalam chatbot.

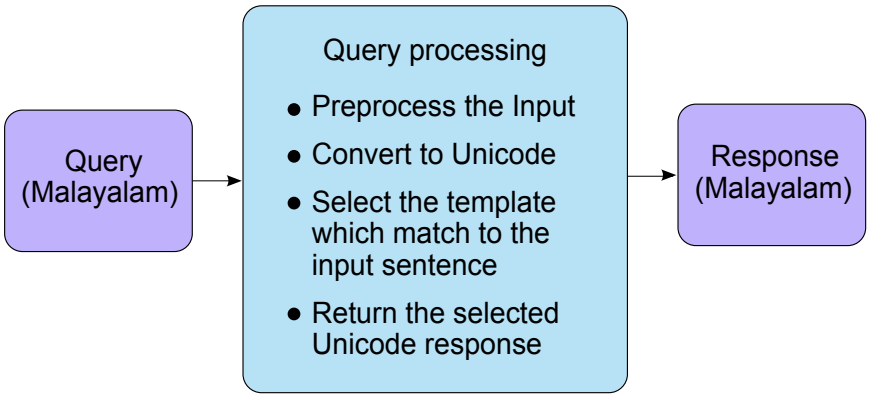


Figure 6.1: Block diagram of the proposed Malayalam chatbot

Queries in Malayalam are taken as input from the console of the conversational platform. The received input is pre-processed through different techniques that help in stop word removal, non Malayalam word substitution, removal of punctuations etc. As a part of pre-processing, the content which is provided as input is converted to corresponding Unicode notations. This is subsequently processed by means of a pattern matching algorithm. The algorithm will select the matching response for the query from the knowledge base and returns the generated response in Unicode, which will be displayed to the user in Malayalam.

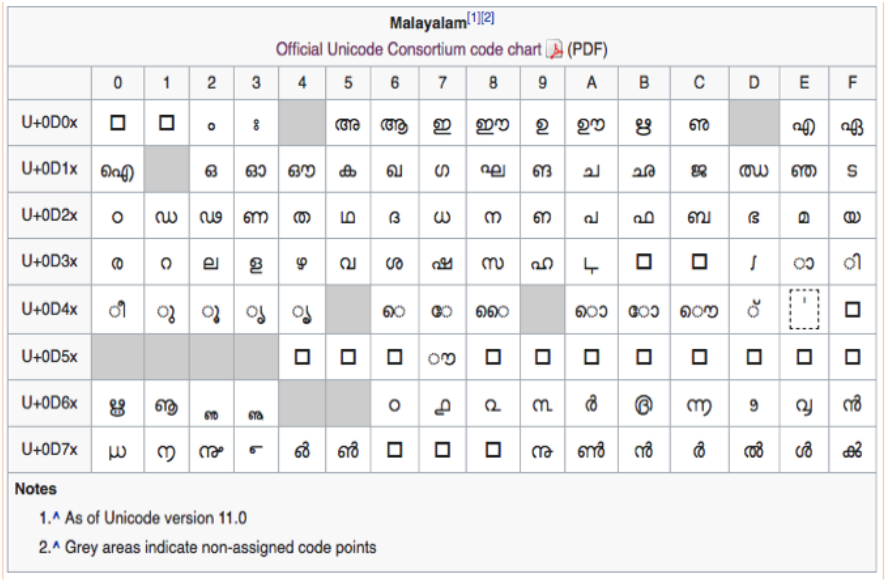
This has been achieved by modifying the AIML based ProgramD

platform. The chat engine is modified to get the response in Malayalam by training the chatbot through the Malayalam AIML knowledge base, applying additional pre-processing techniques and by modifying the input and output paradigms. Following subsections details the knowledge base composition, data pre-processing techniques and the implementation of this chatbot.

6.3.1 Knowledge Base Composition

Based on the knowledge base composition, the performance of response generation might vary for a chatbot. Hence knowledge base composition has an important role in the development of chatbot. Unlike English, many regional languages have consonant conjuncts, modifiers, and other graphemes and thus cannot be represented with ASCII encoding. One of the widely accepted methods to represent such regional languages is the Unicode representation. The Malayalam script used for the Malayalam language, which belongs to the Dravidian group under the Unicode, range between 0D00-0D7F [164]. Fig. 6.2 displays the Unicode chart of the Malayalam script which contains 128 code points. Out of 128 code points, 117 code points are assigned and the remaining 11 code points are reserved which are indicated as grey cells.

Each character can be expressed as Unicode numeric entity codes with each code prefaced with &# and is separated by a semicolon. The hexadecimal codes are prefixed with additional symbol 'x'. As an illustration, the decimal and hexadecimal entity code for the word '*Malayalam*' (writing as മലയാളം in Malayalam script) would be represented as in Fig. 6.3.



Malayalam[1][2]

Official Unicode Consortium code chart (PDF)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0D0x	□	□	ം	ഃ		അ	ആ	ഇ	ഈ	ഉ	ഊ	ഋ	ഌ		എ	ഏ
U+0D1x	ഐ		ഒ	ഓ	ഔ	ക	ഖ	ഗ	ഘ	ങ	ച	ഛ	ജ	ഝ	ഞ	ട
U+0D2x	ഠ	ഡ	ഢ	ണ	ത	ഥ	ദ	ധ	ന	ഩ	പ	ഫ	ബ	ഭ	മ	യ
U+0D3x	ര	റ	ല	ള	ഴ	വ	ശ	ഷ	സ	ഹ	ഺ	□	□	ഽ	ാ	ി
U+0D4x	ീ	ു	ൂ	ൃ	ൄ		ൊ	ോ	ൌ		ൊ	ോ	ൌ	്	ൗ	□
U+0D5x					□	□	□	ൗ	□	□	□	□	□	□	□	□
U+0D6x	ൠ	ൡ	ൢ	ൣ			൦	൧	൨	൩	൪	൫	൬	൭	൮	൯
U+0D7x	൰	൱	൲	൳	൴	൵	□	□	□	൹	ൺ	ൻ	ർ	ൽ	ൾ	ൿ

Notes

1.^ As of Unicode version 11.0

2.^ Grey areas indicate non-assigned code points

Figure 6.2: Unicode Consortium code chart of Malayalam

Unicode Decimal version of 'Malayalam' word:
മലയാളം

Unicode HexaDecimal version of 'Malayalam' word:
മലയാളം

Figure 6.3: Unicode representation of the word 'Malayalam'

The knowledge base of Malayalam chatbot is stored in AIML format, which is composed of Unicode entity codes for the Malayalam script. The AIML knowledge base contains pattern and template tags, and the values of these tags are stored as Unicode of Malayalam characters. The $< pattern >$ contains the query and its matching response that is saved as $< template >$.

6.3.2 Preprocessing Mechanism for Malayalam Chatbot

Chatbots require a pre-processing pipeline which can help to handle different variations or format of input data more efficiently to find the proper response. Chatbot's pre-processors are simple functions that modify the input statements received by the chatbot before the statement is actually processed. AIML interpreter engine will conduct normalisation techniques like substitution, sentence splitting and pattern fitting [7]. These normalisation techniques will take care of susceptible information loss, sentence splitting and punctuation removal. In addition, incorporated some other pre processing techniques which are detailed in the following sub sections. The complete preprocessing pipeline is shown in Fig. 6.4.

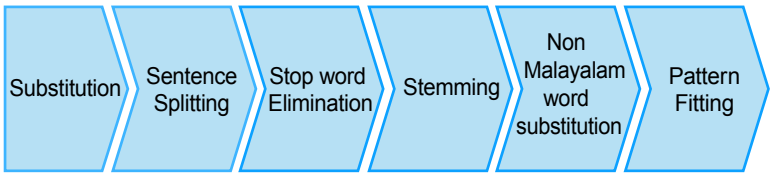


Figure 6.4: Pre-processing pipeline

6.3.2.1 Stop Word Elimination

Stop-words are frequently occurring words in a natural language which are considered as unimportant in an information retrieval system. Such stop words, can be filtered out before the processing of the text input [165]. Some of the main stop words in the Malayalam language are given below and read as *athu, ithu, ennal, athil, enniva, and ipozhum* respectively.

അത്, ഇത്, എന്നാൽ, അതിൽ, എന്നിവ, ഇപ്പോഴും

The stop word elimination has been implemented by means of a look up table. Each word in the input sentence is compared with the entries in the look up table which contains Malayalam stop words. If there is a match, the selected word in the sentence is removed and the next word is selected for comparison. By removing the stop words, we can reduce the knowledge base entries as there will be less number of variants for the same query.

6.3.2.2 Stemming

The agglutinative nature of the Malayalam language calls for the process of Stemming which reduces the inflected words to their root form. Due to inflection, the same word can appear with different forms in the sentences which may affect the process of pattern matching [166]. The *Silpa Stemmer* by Swathanthra Malayalam Computing group has been employed to convert the words from the user query to their root form. The stemmer removes longest matching suffix from each word with proper replacement to get the base word.

6.3.2.3 Non Malayalam Word Substitution

There is a possibility of entering commonly used non-Malayalam words also in the input query, like time notation (AM, PM), date notations (15-Oct-18) and abbreviations, and the Malayalam chatbot will not be able to identify these constructs as part of the pattern matching process. Hence a Non Malayalam Word Substitution method

which could replace such non Malayalam constructs has been proposed and implemented. By using substitution one can reduce the non-matching pattern issues with non Malayalam input.

As depicted in Fig. 6.5, some queries by the user might get mixed up with Malayalam, English or Numeric text. In such cases where the user has entered non-Malayalam text, it will be identified by evaluating the Unicode range of Malayalam. If the Unicode values of words are not found in the range of Malayalam, it is then considered for numeric range validation. If the same is not found under numeric range, it will be validated with English alphabet range.

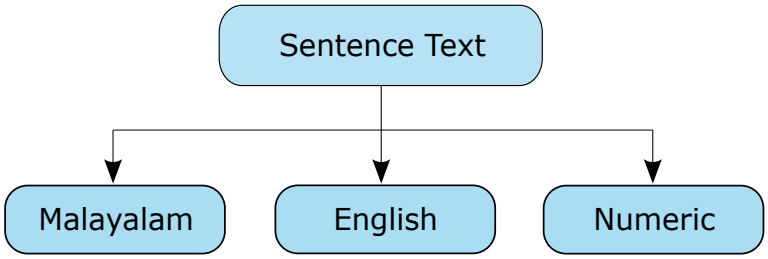


Figure 6.5: Text structure in input sentence

Thus, the non-Malayalam texts are categorised either as English or numeric types. A Malayalam reference word list has been created which contain commonly used English words or numeric notations and their corresponding Malayalam representation.

6.3.3 Response Generation

The proposed system accepts queries in Malayalam and responds to the user in Malayalam. In this implementation, Karthika font, which is an OpenType font with TrueType outlines for the Indic script Malayalam, has been made use of. The user interface of the Malayalam

chatbot handles the display of Malayalam characters with the support of Unicode values and Malayalam fonts. The user can input Malayalam text to the chatbot using Malayalam enabled screen keyboard.

The input entered by the user is preprocessed and is converted to corresponding Unicode values, and the same is fed to AIML interpreter. AIML engine is re-trained with the AIML file which stores Malayalam Unicode values. As these AIML files are refashioned with UTF-8 encoding, the AIML interpreter is able to recognize Malayalam characters. Chat engine tries to match the input, word by word to find a matching pattern in AIML and provides the corresponding template as a response to the user. The response generated will be displayed in Malayalam text. The algorithm of the proposed chatbot is given in Algorithm 6.1.

6.4 Prototype Implementation of Malayalam Chatbot

A good proportion of Kerala's population are aspirants of Government jobs. Out of the total Government job vacancies, a major chunk is for class IV, where the minimum qualification is SSLC, and most of the applicants are not that comfortable with English. As per the data published in the official website of Employment Department of Government of Kerala, the total number of applicants who have registered for Government jobs are 35,89,093 and out of these applicants 22,17,004 are matriculates [167]. This calls for the need for a system that can communicate with the applicants in their mother tongue.

Algorithm 6.1 Malayalam chatbot response generation

Input: User query in Malayalam

Output: Malayalam Response to query

```
1:  procedure MALAYALAMCHATBOT
2:      Retrieve Input Malayalam user query
3:      Normalise the input by substitution normalisation technique
4:      Split the query into sentence by sentence splitting normalisation
5:      Eliminate Malayalam stop words and stemming
6:      Convert text into Unicode
7:      Applies non Malayalam word substitution
8:      Find matching pattern from loaded Malayalam AIML KB
9:      if match is found then
10:         Generate the response by using the template tag value stored
             in AIML
11:     else
12:         Pick the stored static message as response
```

As information gathering, application filling and submission is done through Kerala Public Service Commission (KPSC) website. A chatbot that can respond in Malayalam can help the applicants to clear their doubts in local language itself. Fig. 6.6 displays the user interface of the proposed Malayalam chatbot with a simple interaction session between a user and chatbot. The applicant is trying to figure out what all documents have to be submitted while applying for the job. The chatbot has replied that there is no need to submit any documents along with the application. Few sample entries of the AIML knowledge base is illustrated in Fig. 6.7. For readability the actual Unicode entries in knowledge base have been converted to the corresponding

Malayalam characters.

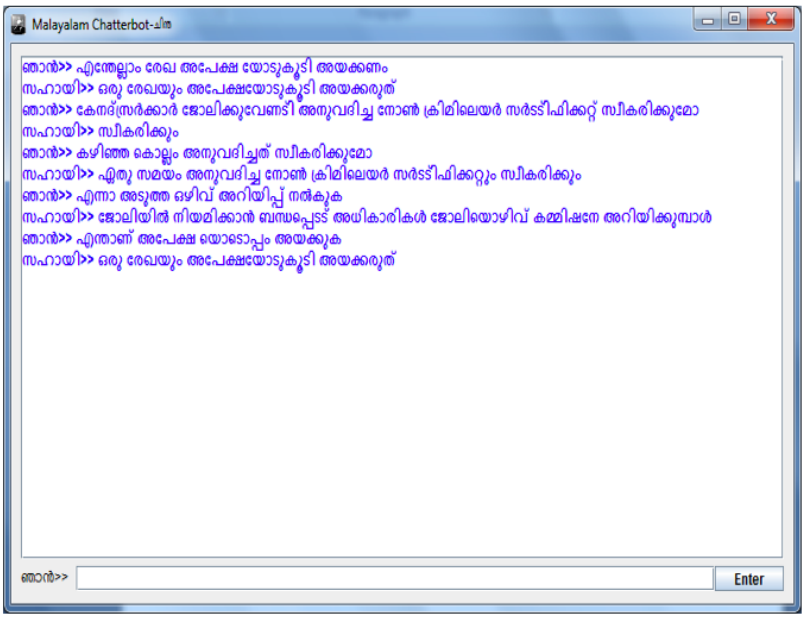


Figure 6.6: User interface of the Malayalam chatbot

To the subsequent query from the applicant regarding the validity of non-creamy layer certificate provided earlier by the central government, the chatbot has replied that the same will be accepted. The chatbot has also ascertained that the same certificate that was obtained a year above is still valid. The applicant is also trying to know when he/she will get information about next the job opening, to which the chatbot has replied that it will be notified when the concerned authority informs the commission. The bot is also confirming that there is no need to submit any proof along with the application when the user is asking the same question which asked first in a different way with the help of <srai> tag implementation.

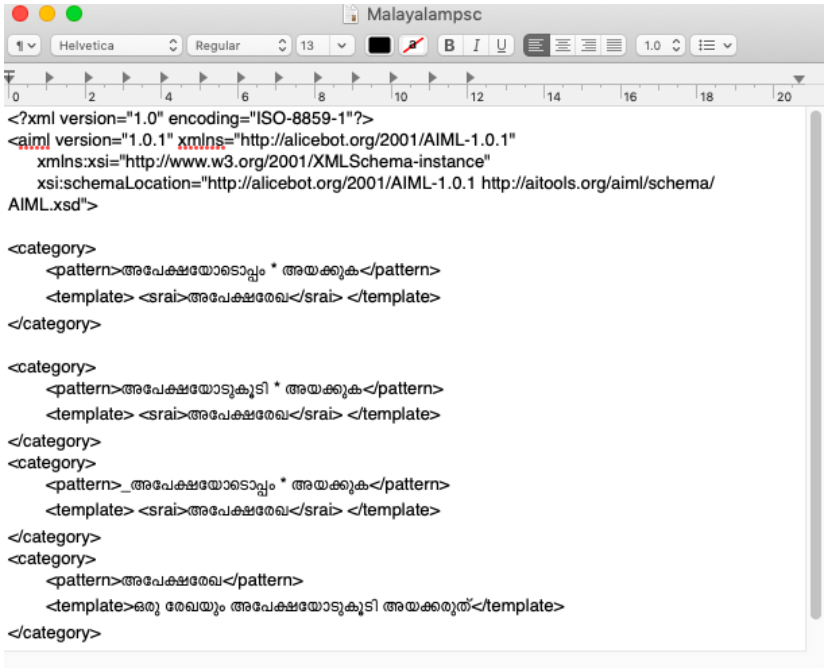


Figure 6.7: Sample of Malayalam AIML KB

6.5 Malayalam Knowledge base Synthesiser

Knowledge base provides the required intelligence to a system. The ability of the chatbot to respond to questions might vary depending upon the depth of the knowledge base provided to the chatbot and thus, a lot of attention has to be given while creating the knowledge base. Manually synthesizing the AIML based Malayalam knowledge base by using Unicode entity codes is a tedious task as the user should have knowledge of Unicode entity codes for each Malayalam letter. Thus, a tool for generating the AIML knowledge base for Malayalam scripts is essential for the Malayalam chatbot, which can help in the easy construction of the knowledge base in Malayalam.

For saving the KB as AIML file, first, transliterate the Malayalam script input entered through UI to Unicode entity code sequence. The next step is to transform the input parameters to AIML format with appropriate tags and Unicode values. This transformed format is saved as AIML knowledge base to the specified path. The process flow of this KB synthesiser is portrayed in Fig. 6.8.

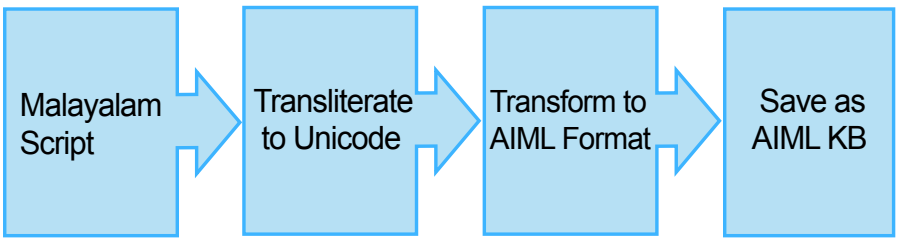


Figure 6.8: Process flow of the KB synthesiser

The user interface of knowledge base synthesizer which helps in entering Malayalam question and response to update the knowledge base is illustrated in Fig. 6.9. The file will be saved with an extension of .aiml and name of the file with the path can be selected from the UI. The entered input data is saved to AIML knowledge by adding proper AIML objects with header and tags. Query and response will be saved as <pattern> and <template> correspondingly under the <category> object. The Malayalam text entered in 'Pattern (Question)' and 'Template (Answer)' text box of the user interface is saved as the Unicode value for pattern and template tags respectively.

This tool also has an option to select any existing pattern for mapping to <srai> tag. Simple Recursive Artificial Intelligence (<srai>) tag [34] is an additional property of template, which can map many patterns of input to the same response and hence helps to solve the issues of synonymous input patterns. By selecting SRAI check box, we can select

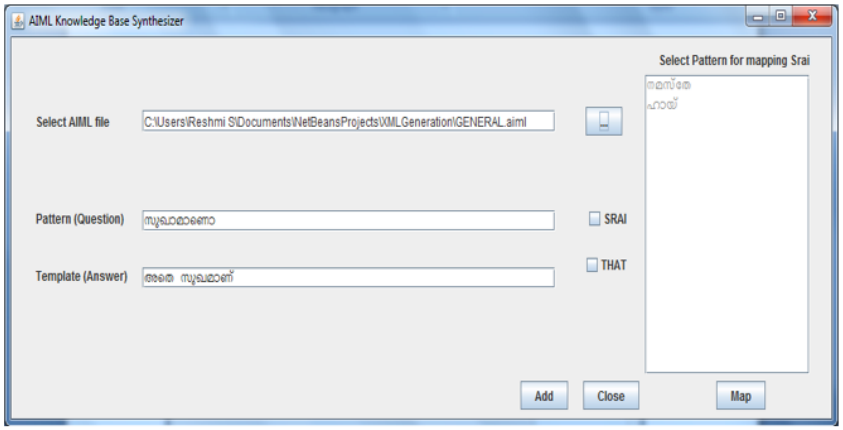


Figure 6.9: The UI of the Malayalam knowledge base synthesiser

any existing patterns already stored in AIML file from the list displayed in the right side of this window. A provision for saving <that> tag has also been incorporated using the 'THAT' check box. <that> tag carries the previous response of the chatbot and indicates whether the current input pattern depends on the last output response.

6.6 Evaluation of the Malayalam Chatbot

The prototype chatbot was evaluated with a query set composed manually. Feed backs of the users were also collected to ascertain the need and ease of use of the chatbot .

6.6.1 Measure of Success Response Rate

To evaluate the effectiveness of the Malayalam chatbot, a test script has been prepared and ran on query set which has been composed manually by selecting 100 frequently asked queries related to the PSC

forum. From this script, randomly selected four query set (Q1 to Q4) with number of queries 25, 50, 75 and 100 respectively. The system was evaluated with this query set by counting both success and failure responses.

The success rate for each query set has been computed as the percentage of the ratio of the correctly received responses to the total number of queries in the test set. The outcome of the performance results is summarised in Table 6.1. It is seen the Q1 set exhibited the lowest success rate of 56.0% and Q4 set with highest success rate of 66%. An average success rate of 64.15% has been obtained.

Table 6.1: Success rate of Malayalam chatbot

Query Set	**No. of Queries**	**Success Responses**	**Failure Responses**	**Success Rate (%)**
Q1	25	14	11	56.0
Q2	50	32	18	64.0
Q3	75	53	22	70.6
Q4	100	66	34	66.0
Average				64.15

6.6.2 Performance Measures

Evaluated the quality of Malayalam chatbot on the basis of precision, recall, accuracy and F1-scores relative to the correct chatbot response. The experimental results for Malayalam chatbot is as presented in Table 6.2. The set Q3 resulted in highest accuracy of 76%. While the chatbot yielded, an average F1-score of 81%, the average accuracy was 70%.

Table 6.2: Malayalam chatbot performance metrics

Query Set	Precision	Recall	F1-Score	Accuracy
Q1	0.7368	0.7778	0.7568	0.6400
Q2	0.7805	0.8421	0.8101	0.7000
Q3	0.8413	0.8689	0.8548	0.7600
Q4	0.8354	0.8148	0.8250	0.7200
Average	0.7985	0.8259	0.8116	0.7050

6.6.3 User Satisfaction Survey

Chatbots being an interactive software application, can be evaluated from a usability perspective and the same has to be measured based on factors like user friendliness, user satisfaction etc. From a user perspective, the main aim of an interactive system is to maximize the user satisfaction and the level of satisfaction can be ascertained using an open-ended feedback mechanism. The normal process followed by an applicant to gather information regarding PSC job is to either contact PSC office through telephone or visit the official website or refer to newspapers. However, these ways are time-consuming and more than that it is difficult to get the answer for specific questions. Deploying a chatbot with a specifically tailored knowledge base can help the users to obtain answers to their queries.

The survey was conducted on 25 applicants by means of Google Forms which is a free platform for conducting surveys. After chatting with the Malayalam chatbot, they were requested to complete this survey. Out of the 25 applicants, 23 responded. The applicants who were selected to participate in the survey are from different educational background and a deliberate attempt was made to select such a sample for evaluation as the chatbot system will be accessed by applicants who

basically falls in different educational levels. The educational levels considered in this survey are High School (L1), Higher Secondary (L2), Graduate (L3) and Post Graduate (L4). An evaluation sheet was prepared which contains three information-seeking questions. Through the survey we intend to understand from the applicants whether the system is user friendly, whether they were able to get relevant information using the chatbot and whether they prefer such a system to traditional way of getting information. The screen shot of the Google survey form used is given in Fig. 6.10.

Twenty three users tried the system and participated in the survey; 9 members were L1 level, 5 users were with L2 level education, 4 were graduates level (L3) and the rest were postgraduates (L4). Satisfaction scores were measured by ten-point Likert scales for each question (Q1, Q2 and Q3), with 1 as the lowest point and 10 if the user is fully agreeing. Table 6.3 summarises the number of points awarded by each group for each question. Overall 168 points were obtained by Q1 which is about the user friendliness of the system. Q2, whether the chatbot answers their queries gained 134 points, while 162 points out of 230 was obtained for Q3, indicating the user preference over conventional methods.

Table 6.3: Malayalam chatbot survey result

Question	**L1**	**L2**	**L3**	**L4**	**Overall**
Q1: Easy to use	71	32	29	36	168
Q2: Getting Answers	40	30	22	42	134
Q3: Preferred	68	33	23	38	162

The computed mean and percentage of the scores of each question for each user group along with the overall values are also furnished in Table 6.4 and has been plotted in Fig. 6.11. The question Q2, whether one is getting relevant answers from the Malayalam chatbot, got 134 points

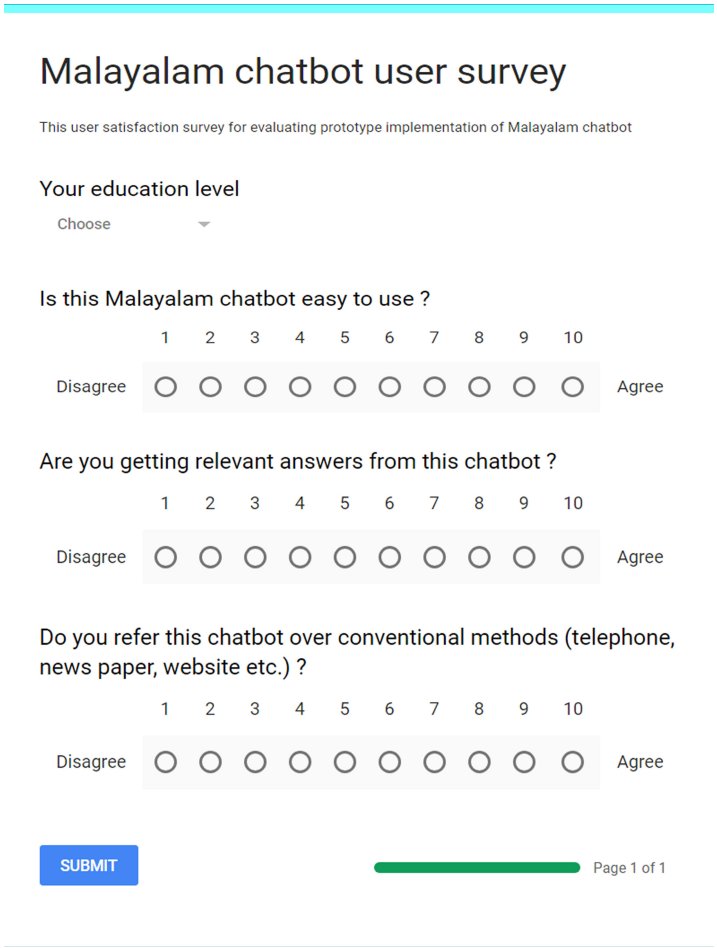


Figure 6.10: Malayalam chatbot user survey form

out of 230, resulting in an overall acceptance of 58%. Since there is no specific format to ask the questions, there are cases where some users could find answers while others could not. The success in finding the answers is based on the way the questions are formed with keywords. Another reason could be the complexity of Malayalam language and it is found that a more extensive knowledge base is needed for practical implementation. As a first attempt of this kind in Malayalam, the results are motivational for further research activities. The percentage of scores

Table 6.4: Mean and Percentage of the user survey scores

User	**Q1**		**Q2**		**Q3**	
	Mean	Percentage	Mean	Percentage	Mean	Percentage
L1	7.88	30.86%	4.44	17.39%	7.55	29.56%
L2	6.40	13.91%	6.00	13.04%	6.60	14.34%
L3	7.25	12.60%	5.5	09.56%	5.75	10.00%
L4	7.20	15.65%	8.40	18.26%	7.60	16.52%
Overall	7.30	73.04%	5.82	58.26%	7.04	70.43%

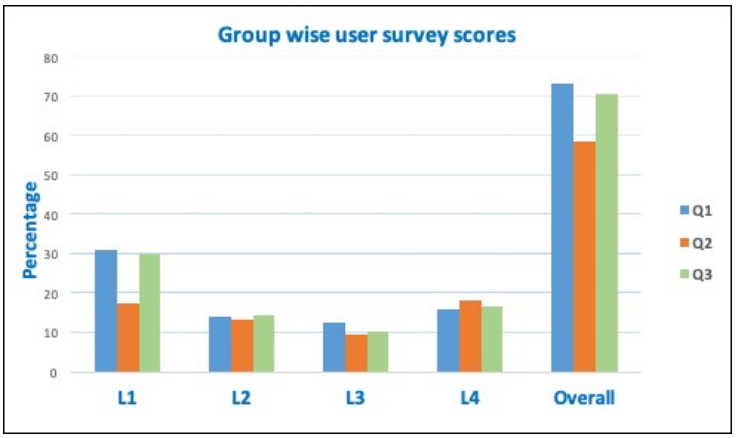


Figure 6.11: Percentage of user survey score

given by the users who preferred the chatbot over conversational means is 29% for L1 group, 14% for L2, 10% for L3 and 16% for L4, resulting in overall proportion of 70%. Also, the overall user-friendliness of the system was reported to be 73%.

6.7 Summary

The significance of a chatbot that can communicate in Malayalam lies in the fact that it can support a large chunk of the population who are not comfortable with English. We hypothesize that this implementation will help to build different types of goal-oriented

chatbots in Malayalam. A retrieval based closed domain chatbot has been developed, which can interact with the users in Malayalam and it is one of its first kind in Malayalam language. This system has been trained by the AIML knowledge base with Unicode encoding - applied additional preprocessing techniques of stop word elimination, stemming and word substitution to improve the response generation.Chatbot was also evaluated in terms of success rate calculations, performance measures and user satisfaction survey. Furthermore, a knowledge base synthesiser for generating query-response mechanism was devised, which could be used for the easy formation of the knowledge base in Malayalam. This synthesizer, makes the Malayalam chatbot as an easily configurable system.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

We can see that a new trend is growing in the field of artificial intelligence in recent years pertaining to the development of chatbots or conversational agents and their applications in different domains. This opens up new arenas for a considerable amount of research and development both conceptually and technically to ensure that these conversational agents can be put into work in a completely unsupervised manner. This thesis addresses one of the emerging topics in the field of conversational agents, *viz.* the design of an inquisitive chatbot. Such a chatbot can probe the users for additional information and makes the communication between man and machine more interactive.

The work reported in the thesis entitled A Hybrid Knowledge Base Chatbot Framework with Enhanced Inquisitiveness and Language Localisation addresses one of the emerging topics of chatbots. In this thesis, we have subjectively described the application of inquisitiveness in chatbots and have laid down a set of user problems and respective constraints. With the support of NLP techniques including NER, the system has been made more reliable for practical implementations. The work also includes the design of language localised chatbot in Malayalam. A detailed review of literature along with various design methodologies is conducted, which helped in thoroughly understanding existing technologies and the drawbacks.

The following are the salient highlights of the thesis.

7.1.1 Design of Inquisitive Chatbot

An inquisitive chatbot framework based on AIML which can analyse the user query and intelligently identify the missing information has been designed and implemented. This system is able to handle first level inquisitiveness by probing the user for the missing information collection. Modified AIML constructs along with RDBMS has been employed for this first level inquisitive implementation. The performance of the system has been further improved with the application of NER by identifying the exact type of missing entities.

7.1.2 Multilevel Inquisitiveness using Ontology

Though the first level inquisitive chatbot can identify and probe the user for the missing information to answer the query, it fails if the query

has more than single missing information. An ontology-based approach has been proposed to overcome this limitation and extend the system to handle multilevel missing information queries. The relational nature of the ontology helps in inferring the objects required to find the missing information.

7.1.3 Hybrid Knowledge based Integration

The knowledge base is a key element in the chatbot implementation as it represents the actual "knowledge" of the system. The proposed chatbot system uses a hybrid knowledge base that includes a modified AIML system, RDBMS along with an integrated big data framework as an external knowledge base. This enables dynamic response generation and improves the analytical capability of chatbots with data from a distributed environment.

7.1.4 Regional language (Malayalam) Integrated Chatbot

Communicating with the chatbot in one's mother tongue makes the communication process easy and comfortable for the user. An attempt has been made for the development of a chatbot that can communicate in the Malayalam language. The user can type in the queries in Malayalam and the chatbot generates responses in Malayalam using a language localised AIML knowledge base. A knowledge base synthesiser for the Malayalam language knowledge base creation has also been implemented. This can be considered as the first exclusive work on a language localised chatbot in Malayalam.

7.2 Future Directions

The work presented in this thesis has a significant role to play in view of its practical applications in chatbot implementation. This work also provides substantial scope for further research towards improving the overall system performance, especially considering the relatively unexplored domain. In this section, we have provided some future enhancement in the purview of the thesis viewpoint and also some future directions that are beyond the viewpoint of this work.

7.2.1 Sentiment Analysis Capability

The inquisitive nature of chatbot can be improved by augmenting sentiment analysis of user query. By sentiment analysis, we are able to compute and understand the emotional tone behind a series of words. This understanding helps to gain the attitude, emotions and opinion of the user behind the certain topic. It is based on this analysis chatbots can inquest and infer in a more meaningful way.

7.2.2 Contextual Understanding by Chat History Analysis

Deep learning methodologies help in analysing the conversation history of the user session. Recursive Neural Network (RNN) can be implemented to analyse the past conversation of a particular user. Depending on this analysis, chatbot can find details of users and can avoid repetitive queries on each session.

7.2.3 Enhancing the Malayalam Chatbot

The language localisation can be further enhanced by adopting more NLP techniques to Malayalam and customising it to suit the properties of the language along with an extensive knowledge base. The design and incorporation of Malayalam ontology models can also be a topic for further research.

7.2.4 Social Media Integration

The proposed techniques in this thesis could be further customised to enable social media integration. As the user engagement in social media platforms is increasing day-by-day, this could leverage more interaction between the bot and the user and the channelisation of information. This can also be extended to reap the benefits of social media marketing and analysis from a commercial perspective.

7.3 Summary

In this chapter, an attempt has been made to bring out the salient highlights of the work carried out for the design of the inquisitive chatbot and the language localisation. A discussion on the scope and directions for future research works in this area has also been presented.

References

[1] A. Lokman and J.M. Zain, "One-Match and All-Match Categories for Keywords Matching in Chatbot," *American Journal of Applied Sciences*, vol. 7, pp. 1406–1411, oct 2010.

[2] S. A. and D. John, "Survey on Chatbot Design Techniques in Speech Conversation Systems," *International Journal of Advanced Computer Science and Applications*, vol. 6, pp. 72–80, oct 2015.

[3] D. Vrajitoru and J. Ratkiewicz, "Evolutionary sentence combination for chatterbots," 01 2004.

[4] G. Galvao, A.M. ; UFPE ; Barros, F.A. ; Neves, A.M.M. ; Ramalho, "Persona-AIML : An Architecture for Developing Chatterbots with Personality," in *Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004*, pp. 1266 – 1267, 2004.

[5] B. Hettige and A. S. Karunananda, "First Sinhala Chatbot in action," in *Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI)*, no. September, pp. 4–10, 2006.

[6] J. Jia, "CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning," *Knowledge-Based Systems-ELSEVIER*, vol. 22, no. 4, pp. 249–255, 2009.

[7] R. S. Wallace and N. Bush, "AIML: Artificial Intelligence Markup Language." `http://web.archive.org/web/20130827100001/http://www.alicebot.org/TR/2005/WD-aiml/`.

[8] N. Albayrak, A. Özdemir, and E. Zeydan, "An overview of artificial intelligence based chatbots and an example chatbot application,"

in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 2018.

[9] L. Benotti, M. C. Martínez, and F. Schapachnik, "Engaging High School Students Using Chatbots," in *ITiCSE '14 Proceedings of the 2014 conference on Innovation & technology in computer science education*, pp. 63–68, 2014.

[10] S. Gupta, D. Borkar, C. D. Mello, and S. Patil, "An E-Commerce Website based Chatbot," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 6, no. 2, pp. 1483–1485, 2015.

[11] Oracle, "Can Virtual Experiences Replace Reality?," tech. rep., 2016.

[12] Z. Ji, Z. Lu, and H. Li, "An Information Retrieval Approach to Short Text Conversation," *CoRR*, vol. abs/1408.6, no. Hang Li, pp. 1–21, 2014.

[13] L. Shang, Z. Lu, and H. Li, "Neural Responding Machine for Short-Text Conversation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, (Stroudsburg, PA, USA), pp. 1577–1586, Association for Computational Linguistics, mar 2015.

[14] M. J. Pereira and L. Coheur, "Just.Chat -a platform for processing information to be used in chatbots," 2013.

[15] B. A. Shawar and E. Atwell, "A chatbot system as a tool to animate a corpus," *ICAME Journal: Computers in English Linguistics*, vol. 29, pp. 5–23, 2005.

[16] J. Jia, "The Study of the Application of a Keywords-based Chatbot System on the Teaching of Foreign Languages," *CoRR*, vol. cs.CY/0310, pp. 1–11, oct 2003.

[17] B. A. Shawar and E. Atwell, "Using the Corpus of Spoken Afrikaans to generate an Afrikaans chatbot," *Southern African Linguistics and Applied Language Studies*, vol. 21, pp. 283–294, nov 2003.

[18] Z. Yan, N. Duan, J. Bao, P. Chen, M. Zhou, Z. Li, and J. Zhou, "DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 516–525, 2016.

[19] Y. Wu, G. Wang, W. Li, and Z. Li, "Automatic chatbot knowledge acquisition from online forum via rough set and ensemble learning," in *Proceedings - 2008 IFIP International Conference on Network and Parallel Computing, NPC 2008*, pp. 242–246, 2008.

[20] L. Pichponreay, J. H. Kim, C. H. Choi, K. H. Lee, and W. S. Cho, "Smart answering Chatbot based on OCR and Overgenerating Transformations and Ranking," in *International Conference on Ubiquitous and Future Networks, ICUFN*, vol. 2016-Augus, pp. 1002–1005, 2016.

[21] B. A. Shawar and E. S. Atwell, "Using corpora in machine-learning chatbot systems," *International Journal of Corpus Linguistics*, vol. 10, no. 4, pp. 489–516, 2005.

[22] W. Y. Gang, S. Bo, S. M. Chen, Z. C. Yi, and M. P. Zi, "Chinese Intelligent Chat Robot Based on the AIML Language," in *Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 1, pp. 367–370, IEEE, aug 2014.

[23] S. Höhn, "A data-driven model of explanations for a chatbot that helps to practice conversation in a foreign language," in *Proceedings of the SIGDIAL 2017 Conference*, no. August, pp. 395–405, 2017.

[24] C. Segura, J. Luque, and M. R. Costa-juss, "Chatbol , a chatbot for the Spanish "La Liga"," in *International Workshop on Spoken Dialog System Technology 2018 (IWSDS)*, pp. 1–12, 2018.

[25] J. Weizenbaum, "Eliza - A Computer Program For the Study of Natural Language Communication between Man and Machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[26] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[27] K. Colby, "Simulation of belief systems," in *Computer Models of Thought and Language* (R. Schank and K. Colby, eds.), pp. 251–286, Freeman, 1973.

[28] B.Raphael, *The thinking computer*. New York:Freeman, 1976.

[29] G. Güzeldere and S. Franchi, "Dialogues with colorful personalities of early AI," in *Constructions of the Mind. Artificial Intelligence and the Humanities. Special Issue of the Stanford Humanities Review*, vol. 4, pp. 161–170, 1995.

[30] R. Carpenter and J. Freeman, "Computing machinery and the individual: the personal Turing test," tech. rep., 2005.

[31] R. Carpenter, "Jabberwacky chatbot." `http://www.jabberwacky.com/j2about`.

[32] O. V. Deryugina, "Chatterbots," *Scientific and Technical Information Processing*, vol. 37, pp. 143–147, apr 2010.

[33] A.L.I.C.E. AI Foundation, "ALICE." `https://www.chatbots.org/chatbot/a.l.i.c.e/`.

[34] R. Wallace, "The Elements of AIML Style," *ALICE A.I Foundation*, 2003.

[35] Novell, "Watson Supercomputer to Compete on 'Jeopardy!' – Powered by SUSE Linux Enterprise Server on IBM POWER7," *The Wall Street Journal*, 2011.

[36] J. Aron, "How innovative is Apple's new voice assistant, Siri?," *New Scientist*, vol. 212, no. 2836, p. 24, 2011.

[37] D. A. Orr and L. Sanchez, "Alexa , did you get that? Determining the evidentiary value of data stored by the Amazon® Echo," *Digital Investigation*, vol. 24, pp. 72–78, mar 2018.

[38] G. López, L. Quesada, and L. A. Guerrero, "Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces," in *Advances in Human Factors and Systems Interaction. AHFE 2017*, pp. 241–250, 2017.

[39] R. Chawla and J. Anuradha, "Counsellor Chatbot," *International Research Journal of Computer Science (IRJCS) Issue 03, Volume 5 (March 2018)*, vol. 5, no. 03, pp. 126–136, 2018.

[40] A. Følstad and P. B. Brandtzæg, "Chatbots and the new world of HCI," *interactions*, vol. 24, pp. 38–42, jun 2017.

[41] D. L. Beaty, D. Quirk, and J. Jaworski, "Changing landscape of data centers," *ASHRAE Journal*, vol. 59, no. 9, pp. 78–84, 2017.

[42] R. Pirrone, V. Cannella, and G. Russo, "GAIML: A New Language for Verbal and Graphical Interaction in Chatbots," in *2008 International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 715–720, IEEE, 2008.

[43] A. Neves, F. Barros, and C. Hodges, "iAIML: a Mechanism to Treat Intentionality in AIML Chatterbots," in *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pp. 225–231, IEEE, nov 2006.

[44] B. A. Shawar and E. Atwell, "Fostering Language Learner Autonomy Through Adaptive Conversation Tutors," in *CL2007, Proceedings of the Corpus Linguistics Conference*, p. Article #51, 2007.

[45] A. Augello, G. Pilato, G. Vassallo, and S. Gaglio, "A Semantic Layer on Semi-Structured Data Sources for Intuitive Chatbots," in *2009 International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 760–765, IEEE, mar 2009.

[46] B. Abu Shawar and E. Atwell, "An Arabic chatbot giving answers from the Qur'an," in *Proceedings of TALN04*, no. January, pp. 197–202, 2004.

[47] A. Augello, G. Pilato, A. Machi, and S. Gaglio, "An Approach to Enhance Chatbot Semantic Power and Maintainability: Experiences within the FRASI Project," in *2012 IEEE Sixth International Conference on Semantic Computing*, no. June 2015, pp. 186–193, IEEE, sep 2012.

[48] K. Zdravkova, "Conceptual framework for an intelligent chatterbot," in *22nd International Conference on Information Technology Interfaces ITI 2000*, (Pula, Croatia), pp. 189 – 194, IEEE, 2000.

[49] M. L'Abbate, U. Thiel, and T. Kamps, "Can Proactive Behavior turn Chatterbots into Conversational Agents?," in

IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 173–179, IEEE, 2005.

[50] R. Rosa, "Fairytale Child Chatbot," in *14th Conference on Information Technologies – Applications and Theory (ITAT 2014)*, vol. 1214, pp. 79–84, 2014.

[51] C. Chakrabarti and G. F. Luger, "A semantic architecture for artificial conversations," in *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, pp. 21–26, IEEE, nov 2012.

[52] K.-M. Kim, J.-H. Hong, and S.-B. Cho, "A semantic Bayesian network approach to retrieving information with intelligent conversational agents," *Information Processing & Management*, vol. 43, pp. 225–236, jan 2007.

[53] S. Agrawal, M. Haidri, and N. Shruthi, "A Neural Conversational Model for Chatbots," *International Journal of Research in Computer and Communication Technology*, vol. 5, no. 10, pp. 516–519, 2016.

[54] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma, "Topic Aware Neural Response Generation," in *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) Topic*, pp. 412–415, jun 2017.

[55] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep Reinforcement Learning for Dialogue Generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Stroudsburg, PA, USA), pp. 1192–1202, Association for Computational Linguistics, jun 2016.

[56] L. Bradeško and D. Mladenić, "A Survey of Chabot Systems through a Loebner Prize Competition," *Proceedings of Slovenian Language Technologies Society Eighth Conference of Language Technologies*, pp. 34–37, 2012.

[57] H. Al-Zubaide and A. A. Issa, "OntBot: Ontology based chatbot," in *International Symposium on Innovations in Information and Communications Technology*, pp. 7–12, IEEE, nov 2011.

[58] A. Hallili, "Toward an Ontology-Based Chatbot Endowed with Natural Language Processing and Generation," in *26th European Summer School in Logic, Language & Information*, 2014.

[59] J. Lee, J.-H. Park, M.-J. Park, C.-W. Chung, and J.-K. Min, "An intelligent query processing for distributed ontologies," *Journal of Systems and Software*, vol. 83, pp. 85–95, jan 2010.

[60] A. Vegesna, P. Jain, and D. Porwal, "Ontology based Chatbot (For E-commerce Website)," *International Journal of Computer Applications*, vol. 179, pp. 51–55, jan 2018.

[61] O. Gambino, A. Augello, A. Caronia, G. Pilato, R. Pirrone, and S. Gaglio, "Virtual conversation with a real talking head," in *2008 Conference on Human System Interactions*, pp. 263–268, IEEE, may 2008.

[62] A. Santangelo, A. Augello, A. Gentile, G. Pilato, and S. Gaglio, "A Chat-Bot based Multimodal Virtual Guide for Cultural Heritage Tours," in *PSC*, pp. 114–120, 2006.

[63] M. Santos-Perez, E. Gonzalez-Parada, and J. Cano-garcia, "Mobile embodied conversational agent for task specific applications," *IEEE Transactions on Consumer Electronics*, vol. 59, pp. 61o–614, aug 2013.

[64] P. Andrews, M. D. Boni, and S. Manandhar, "Persuasive Argumentation in Human Computer Dialogue," in *Proceedings of the AAAI 2006 Spring Symposium on Argumentation for Consumers of Healthcare*, 2006.

[65] D. Vrajitoru, "NPCs and Chatterbots with Personality and Emotional Response," in *2006 IEEE Symposium on Computational Intelligence and Games*, pp. 142–147, IEEE, may 2006.

[66] R. Fabian and M. Alexandru-nicolae, "Natural language processing implementation on Romanian ChatBot," in *SMO'09 Proceedings of the 9th WSEAS international conference on Simulation, modelling and optimization*, pp. 440–445, 2009.

[67] J. Sjobergh and K. Araki, "A Very Modular Humor Enabled Chat-Bot for Japanese," in *Pacling 2009-Conference of the Pacific Association for Computational Linguistics*, 2009.

[68] S. Higuchi, R. Rzepka, and K. Araki, "A casual conversation system using modality and word associations retrieved from the web," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, no. October, p. 382, 2008.

[69] P. Dybala, M. Ptaszynski, S. Higuchi, R. Rzepka, and K. Araki, "Humor Prevails! - Implementing a Joke Generator into a Conversational System," in *AI '08 Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, pp. 214–225, 2008.

[70] R. Jindal, R. Kumar, R. Sahajpal, S. Sofat, and S. Singh, "Implementing a Natural Language Conversational Interface for Indian Language Computing," *IETE Technical Review*, vol. 21, no. 4, 2015.

[71] T. Kalaiyarasi, R. Parthasarathi, and T. V. Geetha, "POONGKUZHALI - An Intelligent Tamil Chatterbot," in *SIXTH TAMIL INTERNET 2003 CONFERENCE*, pp. 86–95, 2003.

[72] H.-Y. Shum, X. He, and D. Li, "From Eliza to XiaoIce: challenges and opportunities with social chatbots," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, pp. 10–26, jan 2018.

[73] M. Soliman and C. Guetl, "Implementing Intelligent Pedagogical Agents in virtual worlds: Tutoring natural science experiments in OpenWonderland," in *IEEE Global Engineering Education Conference (EDUCON)*, pp. 782–789, IEEE, mar 2013.

[74] Y.-T. Huang, J.-C. Yang, and Y.-C. Wu, "The Development and Evaluation of English Dialogue Companion System," in *Eighth IEEE International Conference on Advanced Learning Technologies*, pp. 864–868, IEEE, 2008.

[75] L. Fryer and R. Carpenter, "EMERGING TECHNOLOGIES Bots as Language Learning Tools," *Language Learning & Technology*, vol. 10, no. 3, pp. 8–14, 2006.

[76] Y. Fei and S. Petrina, "Using Learning Analytics to Understand the Design of an Intelligent Language Tutor – Chatbot Lucy," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 11, pp. 124–131, 2013.

[77] E. Rodriguez, J. C. Burguillo, D. A. Rodriguez, F. A. Mikic, J. Gonzalez-Moreno, and V. Novegil, "Developing virtual teaching assistants for open e-learning platforms," in *19th EAEEIE Annual Conference*, pp. 193–198, IEEE, jun 2008.

[78] G. Dyke, D. Adamson, I. Howley, and C. P. Rose, "Enhancing Scientific Reasoning and Discussion with Conversational Agents," *IEEE Transactions on Learning Technologies*, vol. 6, pp. 240–247, jul 2013.

[79] D. Griol and Z. Callejas, "An Architecture to Develop Multimodal Educative Applications with Chatbots," *International Journal of Advanced Robotic Systems*, vol. 10, p. 1, 2013.

[80] V. Kumar and A. Keerthana, "Sanative Chatbot For Health Seekers," *International Journal Of Engineering And Computer Science*, vol. 05, no. 16022, pp. 16022–16025, 2016.

[81] H. Kazi, B. S. Chowdhry, and Z. Memon, "MedChatBot: An UMLS based Chatbot for Medical Students," 2012.

[82] B. E. V. Comendador, B. M. B. Francisco, J. S. Medenilla, S. M. T, and T. B. E. Serac, "Pharmabot : A Pediatric Generic Medicine Consultant Chatbot," *Journal of Automation and Control Engineering*, vol. 3, no. 2, pp. 137–140, 2015.

[83] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, "SuperAgent: A Customer Service Chatbot for E-commerce Websites," in *Proceedings of ACL 2017, System Demonstrations*, (Stroudsburg, PA, USA), pp. 97–102, Association for Computational Linguistics, 2017.

[84] A. R. Asadi and R. Hemadi, "Design and implementation of a chatbot for e-commerce," in *Information Communication Technology Development and Doing Business*, 2018.

[85] H. Joshi, "Proposal of Chat Based Automated System for Online Shopping," *American Journal of Neural Networks and Applications*, vol. 3, no. 1, p. 1, 2017.

[86] C. J. Kulathunge, R. S. M. M. S. Seneviratne, K. K. J. Kasthuriarachchi, and B. H. S. H. Perera, "Automated Online Customer Care System Using Artificial Intelligence," in *PNCTM*, vol. 3, pp. 74–78, 2014.

[87] A. K. Pani and P. Venugopal, "Implementing e-CRM using intelligent agents on the internet," in *2008 International Conference on Service Systems and Service Management*, pp. 1–6, IEEE, jun 2008.

[88] S. Ghose and J. J. Barua, "Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 1–5, IEEE, may 2013.

[89] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, "A New Chatbot for Customer Service on Social Media," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, (New York, New York, USA), pp. 3506–3510, ACM Press, 2017.

[90] R. P. Mahapatra, N. Sharma, A. Trivedi, and C. Aman, "Adding interactive interface to E-Government systems using AIML based chatterbots," in *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, pp. 1–6, IEEE, sep 2012.

[91] a. Hoshino, K. Kato, J. Takeuchi, and H. Tsujino, "A chat information service system using a humanoid robot," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pp. 468–473, IEEE, 2005.

[92] R. D. Dharani and A. Suthar, "Integration of Aiml Chatter Bot for News Application on Whatsapp," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 5, pp. 2158–2161, 2016.

[93] A. S. Lokman and J. Zain, "Extension and Prerequisite: An Algorithm to Enable Relations Between Responses in Chatbot Technology," *Journal of Computer Science*, vol. 6, pp. 1212–1218, oct 2010.

[94] R. Kar and R. Haldar, "Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, pp. 147–154, 2016.

[95] R. S. Wallace, "The Anatomy of A.L.I.C.E.," in *Parsing the Turing Test*, pp. 181–210, Dordrecht: Springer Netherlands, 2009.

[96] M. d. G. Bruno Marietto, R. V. Aguiar, G. D. O. Barbosa, W. T. Botelho, E. Pimentel, R. d. S. Franca, and V. L. da Silva, "Artificial Intelligence Markup Language: A Brief Tutorial," *International Journal of Computer Science & Engineering Survey*, vol. 4, pp. 1–20, jun 2013.

[97] N. Bush, "programd - AIML-based conversational bot server." `https://github.com/noelbush/programd`.

[98] K. BRENNAN, "The Managed Teacher : Emotional Labour , Education , and Technology," *Educational Insights*, vol. 10, no. 2, pp. 55–65, 2006.

[99] F. A. Mikic, J. C. Burguillo, M. Llamas, D. A. Rodriguez, and E. Rodriguez, "CHARLIE: An AIML-based chatterbot which works as an interface among INES and humans," in *2009 EAEEIE Annual Conference*, pp. 1–6, IEEE, jun 2009.

[100] V. Hung, M. Elvir, A. Gonzalez, and R. Demara, "Towards a Method For Evaluating Naturalness in Conversational Dialog Systems," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1236–1241, 2009.

[101] A. Venkatesh, C. Khatri, A. Ram, F. Guo, R. Gabriel, A. Nagar, R. Prasad, M. Cheng, B. Hedayatnia, A. Metallinou, R. Goel, S. Yang, and A. Raju, "On Evaluating and Comparing Conversational Agents," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 1–10, 2017.

[102] O. S. Goh, C. Ardil, W. Wong, and C. C. Fung, "A Black-box Approach for Response Quality Evaluation of Conversational Agent Systems," *International Journal of Computational Intelligence*, vol. 3, no. 3, pp. 195–203, 2007.

[103] B. Kirkpatrick and B. Klingner, "Turing's Imitation Game: a discussion with the benefit of hind-sight," *Berkeley Computer Science course*, p. 13, 2009.

[104] P. Hingston, "A Turing Test for Computer Game Bots," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, pp. 169–186, sep 2009.

[105] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, "PARADISE: A Framework for Evaluating Spoken Dialogue

Agents," *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 271–280, 1997.

[106] B. A. Shawar and E. Atwell, "Different measurements metrics to evaluate a chatbot system," in *NAACL-HLT-Dialog '07 Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, no. April, pp. 89–96, 2007.

[107] D. Coniam, "The linguistic accuracy of chatbots: usability from an ESL perspective," *Text & Talk*, vol. 34, pp. 545 – 567, jan 2014.

[108] M. O. Meira and A. M. P. Canuto, "Evaluation of Emotional Agents ' Architectures : an Approach Based on Quality Metrics and the Influence of Emotions on Users," in *Proceedings of the World Congress on Engineering 2015*, vol. I, pp. 143–150, 2015.

[109] M. Kaleem, O. Alobadi, J. O. Shea, and K. Crockett, "Framework for the Formulation of Metrics for Conversational Agent Evaluation," in *RE-WOCHAT: Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents - Development and Evaluation*, pp. 20–23, 2016.

[110] N. E. Fenton and S. L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*. PWS Publishing Company, 1997.

[111] M. Hijjawi, Z. Bandar, and K. Crockett, "A General Evaluation Framework for Text Based Conversational Agent," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 3, pp. 23–33, 2016.

[112] H. Shah, P. Bhandari, K. Mistry, S. Thakor, M. Patel, and K. Ahir, "Study of Named Entity Recognition for Indian Languages," *International Journal of Information Sciences and Techniques (IJIST)*, vol. 62, no. 1, pp. 11–25, 2016.

[113] R. Sharnagat, "Named Entity Recognition Literature Survey," in *11305R013*, 2014.

[114] M. M. L. Patawar and M. A. Potey, "Approaches to Named Entity Recognition: A Survey," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 12, pp. 12201–12208, 2015.

[115] D. Chopra, "Hindi Named Entity Recognition By Aggregating Rule Based Heuristics and Hidden Markov Model," *International Journal of Information Sciences and Techniques*, vol. 2, pp. 43–52, nov 2012.

[116] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[117] D. M. Bikei, S. Miller, R. Schwartz, and R. Weischedel, "Nymble : a High-Performance Learning Name-finder," in *Fifth Conference on Applied Natural Language Processing*, pp. 194–201, 1997.

[118] R. G. Andrew Borthwick , John Sterling , Eugene Agichtein, "NYU : Description of the MENE Named Entity System as Used in MUC-7," in *Seventh Message Understanding Conference (MUC-7)*, 1998.

[119] V. Vapnik, *Statistical learning theory.* 1998.

[120] H. Yamada, T. Kudo and Y. Matsumoto, "Japanese Named Entity Extraction using Support Vector Machine," *Transactions of IPSJ*, vol. 43, no. 1, pp. 44–53, 2001.

[121] John Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pp. 282–289, 2001.

[122] S. Kale and S. Govilkar, "Survey of Named Entity Recognition Techniques for Various Indian Regional Languages," *International Journal of Computer Applications*, vol. 164, pp. 37–43, apr 2017.

[123] C. Sutton and A. Mccallum, "An Introduction to Conditional Random Fields," *Foundation and Trends in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[124] U. Irmak and R. Kraft, "A scalable machine-learning approach for semi-structured named entity recognition," in *Proceedings of the 19th international conference on World wide web - WWW '10*, (New York, New York, USA), p. 461, ACM Press, 2010.

[125] N. Patil, A. S., and B. V., "Survey of Named Entity Recognition Systems with respect to Indian and Foreign Languages,"

International Journal of Computer Applications, vol. 134, pp. 21–26, jan 2016.

[126] "Stanford CoreNLP-Natural language software." `https://stanfordnlp.github.io/CoreNLP/index.html`.

[127] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

[128] "Stanford CoreNLP-Named Entity Recognition." `https://stanfordnlp.github.io/CoreNLP/ner.html`.

[129] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, no. 1995, pp. 363–370, 2005.

[130] T. Gruber, "Ontology," 2008.

[131] N. F. Noy and D. L. Mcguinness, "Ontology Development 101 : A Guide to Creating Your First Ontology," tech. rep., 2000.

[132] N. Malviya, N. Mishra, and S. Sahu, "Developing University Ontology using protégé OWL Tool : Process and Reasoning," *International Journal of Scientific & Engineering Research Volume*, vol. 2, no. 9, pp. 1–8, 2011.

[133] F. Baader and W. Nutt, "Basic Description Logics," in *The Description Logic* (F., P. Baader, D. Calvanese, D. McGuinness, D. Nardi, and Patel-Schneider, eds.), pp. 47–100, 2002.

[134] Z. ChuangLu, "Research on the Semantic Web Reasoning Technology," in *AASRI Conference on Computational Intelligence and Bioinformatics Research, Elsevier*, pp. 87–91, Zhu Chuanglu, 2012.

[135] K. Dentler, R. Cornet, A. Ten Teije, and N. De Keizer, "Comparison of reasoners for large ontologies in the OWL 2 EL profile," *Semantic Web*, vol. 2, no. 2, pp. 71–87, 2011.

[136] B. Liu, J. Li, and Y. Zhao, "Repairing and reasoning with inconsistent and uncertain ontologies," *Advances in Engineering Software*, vol. 45, no. 1, pp. 380–390, 2012.

[137] S. Abburu, "A Survey on Ontology Reasoners and Comparison," *International Journal of Computer Applications*, vol. 57, no. 17, pp. 33–39, 2012.

[138] R. Shearer, B. Motik, and I. Horrocks, "HermiT: A Highly-Efficient OWL Reasoner," in *5th International Workshop on OWL: Experiences and Directions (OWLED 2008)*, dec 2008.

[139] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, "HermiT: An OWL 2 Reasoner," *Journal of Automated Reasoning*, vol. 53, pp. 245–269, oct 2014.

[140] E. Alatrish, "Comparison Some of Ontology Editors," *Management Information Systems,*, vol. 8, pp. 18–24, jan 2013.

[141] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu, "The evolution of Protégé: an environment for knowledge-based systems development," *International Journal of Human-Computer Studies*, vol. 58, no. 1, pp. 89–123, 2003.

[142] M. A. Musen, "The protégé project: A Look Back and a Look Forward," *AI Matters*, vol. 1, pp. 4–12, jun 2015.

[143] A. Escórcio and J. Cardoso, "Editing Tools for Ontology Construction," *Semantic Web Services: Theory, Tools and Applications (Eds.)*, no. June, pp. 1–27, 2007.

[144] N. F. Noy, M. Crub, R. W. Fergerson, H. Knublauch, S. W. Tu, J. Vendetti, M. A. Musen, and S. M. Informatics, "Protege-2000 : An Open-Source Ontology-Development and Knowledge-Acquisition Environment," in *AMIA 2003 Open Source Expo Symposium*, p. 953, 2003.

[145] J. Heidrich, A. Trendowicz, and C. Ebert, "Exploiting Big Data's Benefits," *IEEE SOFTWARE*, vol. 33, no. 4, pp. 111–116, 2016.

[146] M. Beyer and D. Laney, "The Importance of 'Big Data': A Definition," tech. rep., Gartner, G00235055, 2012.

[147] S. K. Garima Rani, "Hadoop Technology to Analyze Big Data," *International Journal of Engineering Development and Research (IJEDR)*, vol. 3, no. 4, pp. 949–952, 2015.

[148] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, "Big data analytics on Apache Spark," *International Journal of Data Science and Analytics*, vol. 1, no. 3-4, pp. 145–164, 2016.

[149] Apache Hadoop, "Apache Hadoop." `http://hadoop.apache.org/`.

[150] Apache Hive, "Apache Hive documentation." `https://cwiki.apache.org/confluence/display/Hive/Home`.

[151] Edupristine, "Hadoop Ecosystem and its components." `http://www.edupristine.com/blog/hadoop-ecosystem-and-components`.

[152] Apache HCatalog, "HCatalog documentation." `https://cwiki.apache.org/confluence/display/Hive/HCatalog+UsingHCat`.

[153] J. Dittrich and J.-a. Quian, "Efficient Big Data Processing in Hadoop MapReduce," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2014–2015, 2012.

[154] Hortonworks Sandbox, "Learning the Ropes of the Hortonworks Sandbox." `http://hortonworks.com/hadoop-tutorial/learning-the-ropes-of-the-hortonworks-sandbox/`.

[155] A. Oussous, F. Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, "Big Data technologies: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018.

[156] Apache HiveServer2, "HiveServer2 Clients documentation." `https://cwiki.apache.org/confluence/display/Hive/HiveServer2+Clients{\#}HiveServer2Clients-JDBC`.

[157] Hortonworks Hive, "Process data with Apache Hive." `http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-hive/`.

[158] NY Stock Exchange data, "New York Stock Exchange data." `https://s3.amazonaws.com/hw-sandbox/tutorial1/NYSE-2000-2001.tsv.gz`.

[159] "Constitutional Provisions: Official Language Related Part-17 of The Constitution Of India." `http://rajbhasha.nic.in/en/constitutional-provisions`.

[160] ASER, "Annual Status of Education Report (Rural) 2014," tech. rep., 2015.

[161] "'Classical' status for Malayalam." https://www.thehindu.com/todays-paper/tp-national/classical-status-for-malayalam/article4744630.ece, 2013.

[162] V. Govindaraju and S. R. Setlur, *Guide to OCR for Indic Scripts*. Advances in Pattern Recognition, London: Springer London, 2010.

[163] T. N. M. B. Don M. de Z. Wickremasinghe, *Malayalam Self-taught by the Natural Method with Phonetic Pronunciation (Thimm's System)*. Asian Educational Services, 2005.

[164] "The Unicode Standard 11.0, Malayalam." `http://www.unicode.org/charts/PDF/U0D00.pdf`.

[165] R. R. K. P, "Malayalam News Classification Based On Topic Modeling," *International Journal of Computer & Mathematical Sciences*, vol. 6, no. 10, pp. 18–24, 2017.

[166] S. S. K. P, P. C. R. Raj, and V. Jayan, "Unsupervised Approach to Word Sense Disambiguation in Malayalam," in *International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015)*, vol. 24, pp. 1507–1513, Elsevier B.V., 2016.

[167] "The Official Website of the Employment Department, Govt. of Kerala." `https://www.employmentkerala.gov.in/index.php?option=com{\_}content{\&}view=article{\&}id=21{\&}Itemid=18`.

Publications brought out in the field of research

1. Reshmi. S, Kannan Balakrishnan, "Configurable Malayalam Interface of Intelligent Conversional Agent", National Conference on Indian Language Computing (NCILC – 2011), Feb 2011.
2. Reshmi. S, Kannan Balakrishnan, "Implementation of an inquisitive chatbot for database supported knowledge bases", Sādhanā- Springer – Academy Proceedings in Engineering Sciences, Vol 41, Issue 10, pp-1173–1178, Oct 2016. DOI: https://doi.org/10.1007/s12046-016-0544-1.
3. Reshmi. S, Kannan Balakrishnan, "Empowering Chatbots with Business Intelligence by Big Data Integration", International Journal of Advanced Research in Computer Science, Vol 9, Issue 1, Jan 2018. DOI: https://doi.org/10.26483/ijarcs.v9i1.5398.
4. Reshmi. S, Kannan Balakrishnan, "Enhancing Inquisitiveness of Chatbots through NER Integration", 2018 International Conference on Data Science and Engineering (ICSDE)/ IEEE Xplore, Aug 2018. DOI: 10.1109/ICDSE.2018.8527788.
5. Reshmi. S, Kannan Balakrishnan, "Implementation of an Ontology Based Inquisitive Chatbot", International Journal of Human Computer Studies (Communicated).

Other Publications

1. Reshmi. S, Dr. K. V. Pramod and Dr. Kannan Balakrishnan, "Emotion Recognition from Facial Expressions using Kohonen Neural Network", International Conference on Recent Trends in Computational Science, Kochi, June 2008.
2. Reshmi. S, Dr. K. V. Pramod and Dr. Kannan Balakrishna, "Principal Component Analysis of HRTFs for Enhanced 3D Sound Synthesis", International conference on Intelligent Systems and Networks, Institute of Science and Technology Klawad, February 2008.
3. Mohankumar. K, Reshmi. S and Rathish. R, Grey Technologies Research and Devolopment centre, Cochin, "Automatic Fault Detection in Machineries using Higher Order Spectra", National conference on condition monitoring, Visakhapatnam, December 2006.

Subject Index