INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

Fynbot- Artificial intelligence system for personal expense management

Varisha Ashraf
varisha193@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Ankit Jain
ankitjaincr@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

V B Sree Kumar
vbsreekumar97@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Varun Patwari
varun23patwari@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Dr. Rajesh S L
dr.samsirajesh@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Chandrakanta Saini
cksaini2013iitkgp@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

ABSTRACT

A financial Chatbot is one way for banks, financial service vendors, and other financial organizations to connect with customers on a more personal level. It is meant to simulate a conversation with another human. The user can type a question into a message box, and the Chatbot comes up with the answer instantly. A chat-bot is a conversational agent that can interact with human's turn by turn using natural language. In the world of personal finance, applications are being used by users to answer questions about bank balances, recent transactions, and spending trends which are controlled by operations that as simple as text input and click events. The main objective of this project is to build a Chatbot that enables users to micromanage their expenses in a fast and better way. With the combination of NLP techniques, machine learning techniques and certainly advanced algorithms based on AI we can solve the problem of building a Chatbot to manage personal expenses of an individual. The platform will generate reports and analyze user transaction data in order to manage their expenses and regulate the usage according to budget constraints.

Keywords: *Standford library, Chatbot, IBM Watson, Natural language processing, Artificial intelligence, Hidden Markov Model, Android.*

1. INTRODUCTION

Chatbots are computer programs that are written to simulate a conversation with the user by means of auditory or textual inputs. Recently, the popularity of using chatbots within companies increased, Companies area advancing technically and companies research and explore more possibilities in the area of artificial intelligence and machine learning. When using a new technology, such as chatbots, it is important to know how to implement this technology in a good way.

One way to measure the quality of technology is by looking at the user experience. Also, with conversational interfaces, in particular, chatbots, user experiences play a big part. The user experience of machine learning applications becomes more popular to research

because these technologies did not experience as much design innovations as other technologies have. Previous research has shown that personalization factors influences the user experience. Personalisation is "a process that changes the functionality, interface, information access and content or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals.

2. LITERATURE SURVEY

Chatbots

A conversational interface is an interface that the user can interact with by means of a conversation. This can be via speech, but also via typed natural language. The terms Natural User Interface (NUI) and Conversational Interfaces are sometimes used interchangeably. An NUI is an interface where you interact by using natural inputs like speech, touch and hand gestures Sometimes, NUIs can recognize faces, their environment is motions. A chatbot is an example of a conversational interface. When we speak about chatbots or conversational agents, we mean a computer program that is written to simulate a conversation with the user by means of auditory or textual inputs. In this study, we focus solely on chatbots that use textual inputs. Chatbots are used online and are sometimes driven by artificial intelligence.

Background

ELIZA was the first program that tried to conduct communication with humans. Its creator Joseph Weizenbaum at Massachusetts Institute of Technology (MIT) developed the system on an IBM 7094. The communication with the human was performing through a keyboard and monitor, the input to the computer was written in a natural language with normal punctuation and sentence structure. The only charterer that wasn't allowed where the question mark, it interpreted as line delete in the system. From here the development has been going forward to the present AI-bot A.L.I.C.E, stands for Artificial Linguistic Internet Computer Entity. A.L.I.C.E is somewhat an extension to the ELIZA program, but the two chatbots cannot be compared because of the huge amount of knowledge that has been presented to A.L.I.C.E.

A.L.I.C.E is an artificial intelligence natural chat robot that is based on Alan M. Turing's experiment from 1950 A.L.I.C.E first implementation was conducted in 1995 in the SETL programming language. In 1998 A.L.I.C.E was migrated to the JAVA-platform for platform-independence. At the same time a development of the Artificial Intelligence Mark-up Language (AIML) programming language for A.L.I.C.E was conducted, AIML is an XML like a syntax (Wallace, 2009).

In 1997 a new chatbot was introduced, Jabber wacky. The development began in 1988 and it is unique among AI-Chatbots, much because of that it saves all conversations and tries to learn from them. Jabber wacky is a chatbot that tries to simulate natural human chat in an interesting, entertaining and humorous manner (Carpenter, 2009). The only input that Jabber wacky gets is the interaction with users. This means that if the Jabberwacky is exposed to a foreign language it will learn it over time with the interaction by users. By using the contextual pattern matching technique that is the core for the Jabberwacky it can chat with users.

Conversational aspects

Human-system dialogue consists of the inquirer (the user), looking for information, and the expert (system), providing information. There are two ways that chatbots can converse with users. System-initiated chatbots, where the system leads the conversation, and user-initiated chatbots, where the user leads the conversation. Systems that contain both methods of initiation are called mixed-initiative systems to understand how a conversational interface should be represented, it is important to investigate how human dialogues work. Quarteroni, et al. (2006) researched the aspects and issues related to human dialogues and they pro-posed a list of essential items for an interactive question and answering system:

- Context Maintenance: utilizing the context of the conversation to correctly interpret the user's input. This is important for follow-up questions, or for clarification.
- Utterance Understanding: the detection of follow up and clarification within the context of the previous conversation.
- Mixed-Initiative: the user should be able to take initiative within the conversation.
- Follow-up Proposal: meaning that the system motivates the user to give feedback on the answers that the system gives Until the user has achieved his or her goal.
- Natural Interaction: covering and generating a variety of utterances to create a smooth conversation and to keep the dialogue active.

3. LITERATURE REVIEW

The purpose of a chat bot system is to simulate a human conversation; the chat bot architecture integrates a language model and computational algorithm to emulate information chat communication between a human user and a computer using natural language. With the improvement of data-mining and machine-learning techniques, better decision-making capabilities, availability of corpora, robust linguistic annotations/processing tools standards like XML and its applications, chat bots have become more practical in daily life applications such as help desk tools, information retrieval tools, automatic telephone answering systems, advertising, tools to aid in education, business and E-commerce.

In E-commerce, chat bot helps in information retrieval tasks, such as for searching and browsing, as menu-based navigation poses difficulties in locating the appropriate information. The dialogue system provides additional information on products and simplifies decision-making process to find a product that satisfies customer's requirements.

According to Dr. Wallace, perhaps, the biggest market of chatbot is Entertainment Markets, in which, we can imagine that chatbots can act as a talking book for children and provide foreign language instruction or can be a tutor in Intelligent Tutoring System. One such study used an ALICE system to help Chinese university students practice their conversational English skills. The study was qualitative in nature and used pre-existing conversational English skills. The study focused more on user attitudes rather than on chatter bot efficiency. It was discovered that 62% of users chatted for 10 lines or less and that 8.5% of the time ALICE bot has no specific pattern to match the given input and had to rely on root level generic responses. In all of these conversational entities, one thing is common; and that is, they are having the difficulty of maintaining a dialogue for a sustainable period of time.

Another tutoring study focused on using ALICE as a course enhancement tools with Social and Political Theory knowledge. Chatter bot development is reasonably well studied ever since the Turing Imitation Game was first proposed. Eliza was the first famous chat bot, and ALICE was another milestone.

The Loebner Prize and The Chatterbox Challenge are both annual competitions which have their roots in TIG. However, these are typically text-only experiments, although some limited visual components are often added. This focus is on; however, whether with the text exchange alone, we can replicate human "behavior". This study found that most subjects used the system as a search engine rather than as a conversation partner. It was further concluded that their system was unable to function as a stand-alone tutor. Dialog system can adequately carry out the conversations with the user and can log the conversations which can be a good source for knowledge acquisition for domain-specific topic. Therefore, techniques of knowledge acquisition were rightly used in their system AZ-ALICE chat bot that is an extension in ALICE chatter bot.

3.1 Limitations of the Current Work

Artificial Intelligence (AI)is still not that accessible: The vast majority of chatbots aren't actually intelligent. They are built based on a decision-tree logic, where the response given by the bot depends on specific keywords identified in the user's input.

IF user's input contains 'shop' or 'buy';

THEN send a message with a product list

What that means is that decision-tree types of bots are as intelligent as the capacity (and thoroughness, and patience) of the designer/programmer who created it to anticipate all potential user use cases and inputs. Bots with linguistic and natural language learning capabilities are still quite rare.

Use cases are not that strong**:** Here's something that happens with every new technology that is put out in the world: designers and developers get really excited about it.

What we are seeing now is a gold rush of companies trying to be the first in their category to successfully deploy a bot. In that process, we will see a plethora of bots that are solving for irrelevant use cases or that offer really poor experiences.

It's a natural part of the cycle: our industry needs to learn from its failures before it is able to deploy bots that are truly relevant and smart.

Some bots lack transparency**:** The most successful bots out there make one thing clear from the very beginning of the experience: that the user is chatting with a robot, not with another human. Setting up the right expectations upfront will make users more forgiving about certain mistakes the bot might make.

You certainly want your bot to feel as human as possible but lying to your users and pretending to be something it is not can lead to irreversible loss of trust.

They don't understand the context: Humans are really good at conversations. We understand sarcasm, we can read between the lines, and we are constantly leveraging contextual information when we give someone a response.

They don't communicate with existing business systems: Another common temptation when building a chatbot is trying to recreate functionality from scratch.

Let's say you are creating a bot to book appointments in a spa. If your chatbot does not communicate with the spa's existing appointment management system, that means extra work for the business owner to handle requests coming through this new channel – and ultimately lack of consistency for the user.

4. PROBLEM DEFINITION

Money management tools are too complicated and overwhelming for the average user, inability to process continuous queries in current platforms. We aim to build a flexible system which is user-friendly which uses AI components to deliver personal finance solutions.

4.1 Scope and Relevance

Presently, people across the globe tend to use applications even for the smallest or slightest tasks. A platform which will allow people to cut their costs efficiently and that will help in maintaining a proper record of personal finances will have a good scope in terms of usage. Expenses are something that is mildly neglected by people due to lack of time or lack of awareness about the expenses, with features embedded in the application to draft analysis and help in recording the expenditure of an individual the audience of users is quite considerable.

4.2 OBJECTIVES

- Financial Advisor-The bot can be your personal financial guide giving you advice on how to spend your money. This bot can be a life-saver and the first step to organizing your financial profile. It can also turn complex finance terminology into layman's language saving your time and breaking the math down to simple interactions.
- Expense Saving Bot-Expense Saving Bots help you save and cut down extra spending in your day to day life.
- Simplicity – Messenger applications focus on interpersonal conversations without overload in the users with unnecessary information.
- Accessibility – Anyone with a smartphone and a mobile phone number can register for the service in just a few clicks.

TARGET-

- Students: Students are highly susceptible to make inefficient financial choices. Open up choices that help in making wiser decisions.
- Time-poor Working Class: People who can't spend much time on managing small transactions closely. Provides on-the-go analysis based on transaction history.

5. OVERALL SYSTEM ARCHITECTURE

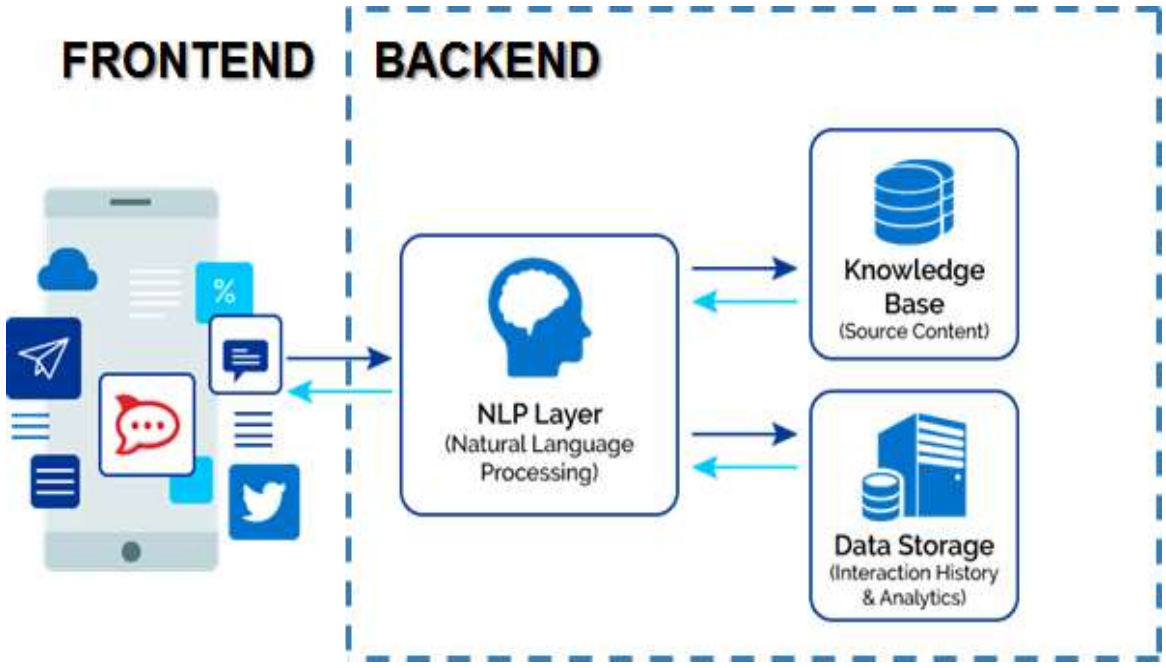


Fig 5.1: Working Model

5.1 Components

UI Component:

- The presentation layer element, responsible for input and output from the user.

NLP layer:

- The language processing engine which comprises various algorithms to receive and analyze human input.

Knowledge base:

- The database where we store the core logic which is used to interpret the information that has been processed.

Dynamic Data Storage:

- The component stores all the user information and interactions such that they can be used to provide analytics on parameters like usage, etc.

5.2 Methodology

Our system uses a Hidden Markov Model, taking the input according to its requirement and helping in identifying the financial terms which are then processed to identify and obtain the type of expenditure and type of value. Comparison of the value and balance is done to validate the expense. The expense priorities are checked and a flag is set or unset upon comparison and prioritization. Based on flag value the bot responds to the user with an easily understandable chat response.

STEP 1: Input is taken in the form of text by an interactive chat interface.
STEP 2: NLP layer will model the data with respect to Hidden Markov Model (HMM). Hidden Markov Model helps in filtering the input and retains the content that is required for processing.
STEP 3: Information is retrieved according to the processed query from a user and the query is logged. Each logged query can be used to retrieve data quickly (dynamic programming).
STEP 4: Simulate this system to attain various input states and populate the knowledge base and storage.

5.3 Algorithm &Activity Diagram

Algorithmic Rules

The main objective of our proposed system is aimed at building a Chatbot that enables users to micromanage their expenses in a fast and better way. With the combination of NLP techniques, machine learning techniques and certainly advanced algorithms based on AI we can solve the problem of building a Chatbot to manage personal expenses of an individual.

We propose a new algorithm which is used to parse and tokenize user input assigning index for easier and efficient expense prioritization. This helps the users to make wiser decisions about their expenses by prioritizing their needs making their task of management cash easier.

STEP 1: Parse the text which has been input.

STEP 2: Create CORENLP pipeline.

STEP 3: Tokenize the input.

STEP 4: Annotate the tokens.

STEP 5: Perform named entity recognition by finding the nature of the element.

5.1 Select parameter for analysis.

5.2 Select value attached to each parameter.

STEP 6: Assign index for the entity.

Activity Model

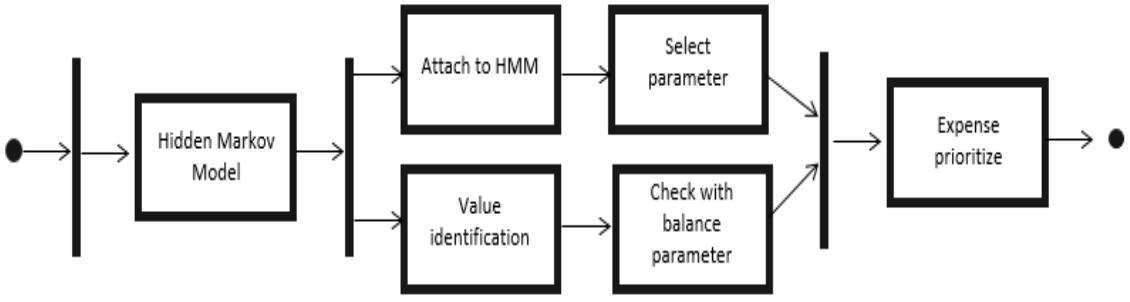


Fig 5.2: HMM Layer Activity Diagram

The text input is modeled according to Hidden Markov Model which helps in identifying the financial parameters. The HMM coded string can now be simplified and made into a parameter-value pair. This process will help in obtaining the type of expenditure and

the value attached to it. The value is then compared with balance for processing and validating the expense. The expense priorities are checked and a flag is set or unset upon comparison and prioritization. Depending on flag value the bot responds to the user with an easily understandable chat response.

6. SYSTEM ANALYSIS

Analysis includes a definite investigation of the present framework, prompting particulars of another framework. Analysis is a point by point investigation of different activities performed by a framework and their connections inside and outside the framework. Amid analysis, information is gathered on the accessible records, choice focuses and exchanges survey are the instruments utilized for framework investigation. The fundamental focuses to be talked about in framework analysis are: Particulars of what the new framework is to achieve in light of the client necessities. Useful order demonstrating the capacities to be performed by the new framework and their association with each other. The procedures of programming building standards framework study and investigation, framework necessity details, framework plan, framework coding, framework testing and execution were gotten from the book, Fundamentals of Software Engineering and also IBM Watson Dashboard and Building an Android Application Essentials.

6.1 Project Features

The featured project section gives the outline of the different errands that are there in the undertaking alongside their understandings in the stages in the task. In the framework study and examination stage the current framework was contrasted and the proposed framework by methods for the investigation done over the span of the undertaking. The attainability ponder was additionally done in this stage. Every one of the necessities that are required in the undertaking; both programming and equipment prerequisites are indicated in the framework prerequisite determination period of the venture. The framework configuration stage in the product improvement life cycle is an inescapable part in the advancement of the task. The whole model for outline work of any product advancement life cycle lies.

6.2 Requirement Analysis

System Analysis is the investigation of frameworks sets of cooperating substances, including PC frameworks. This field is firmly identified with tasks inquire about. It is additionally an express formal request completed to help somebody, alluded to as the leader, recognize a superior strategy and settle on a superior choice than he may have generally made. Work using system analysis incorporates frameworks examiner, business investigator, fabricating designers, undertaking and planner. System Analysis is the way toward looking at a business circumstance to develop a framework answer for an issue or concocting upgrades to such a circumstance. Prior to the improvement of any framework can start, an undertaking proposition is set up by the clients of the potential framework and additionally by framework investigators and submitted to a suitable administrative structure inside the association. So the goal of the system analysis stage is the foundation of the prerequisites for the framework to be procured, created and introduced.

6.3 Feasibility Analysis

Feasibility study is a framework proposition as indicated by its workability, effect on the association, capacity to address client issues and productive utilization of assets. Feasibility study recognizes, portrays and assesses the hopeful framework and chooses the most ideal framework for the activity. A vital result of the preparatory examinations is deciding if the framework asked for is plausible or not.

Economic Feasibility

Economic Feasibility assesses whether the framework benefits more noteworthy than the cost. The proposed framework is taken a toll – powerful one since the advantage of the product exceeds the cost acquired in introducing it. It can be produced under ideal costs with the accessible equipment and programming. The automated framework gives simple and quicker data recovery and spares a great deal of time and labour.

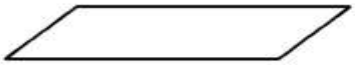
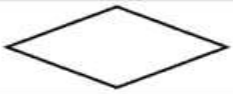
Technical Feasibility

Technical Feasibility includes money related contemplation to oblige specialized improvement. The proposed framework requires a mobile handset and web access alongside cloud bolster for its working. The highlights of the proposed framework like great reaction time, exactness and volume of information taking care of limit and so forth recommend that the framework is actually practical and feasible.

6.4 Data Flow Diagrams [DFD]

The data flow diagrams were first created by Larry Constantine as a method for communicating framework prerequisites in a graphical shape: this prompted a measured plan. A DFD, otherwise called "Bubble Chart", has the reason for clearing up framework necessities and distinguishing real change that will move toward becoming projects in framework outline. A DFD comprises a grouping of bubbles joined by lines. The bubbles represent information change and the line represent the information stream in the framework.

Table 6.1: DFD Notations

Name	Symbol	Function
Start/End		Used to markup the starting and ending point
Arrows		Used for connection
Input/Output		Used for input and output information
Process		Used to represent single step
Decision		Used for branching or decision making

The flow diagram below indicates our system data flow.

- Initially, the user will provide text input through a messaging interface**.**
- The input will then be modeled according to the Hidden Markov Model to help identify the financial terms.
- The HMM coded string is then simplified and tokenized to replace sensitive data with unique identification symbols that retain all the essential information about the data and segmenting the text without compromising its security.
- The tokenized string is stemmed i.e. reducing inflected or derived words to their word stem and a spell check is done on them.
- Followed by obtaining a parameter-value pair.
- The value is then compared with balance for processing and validating the expense.

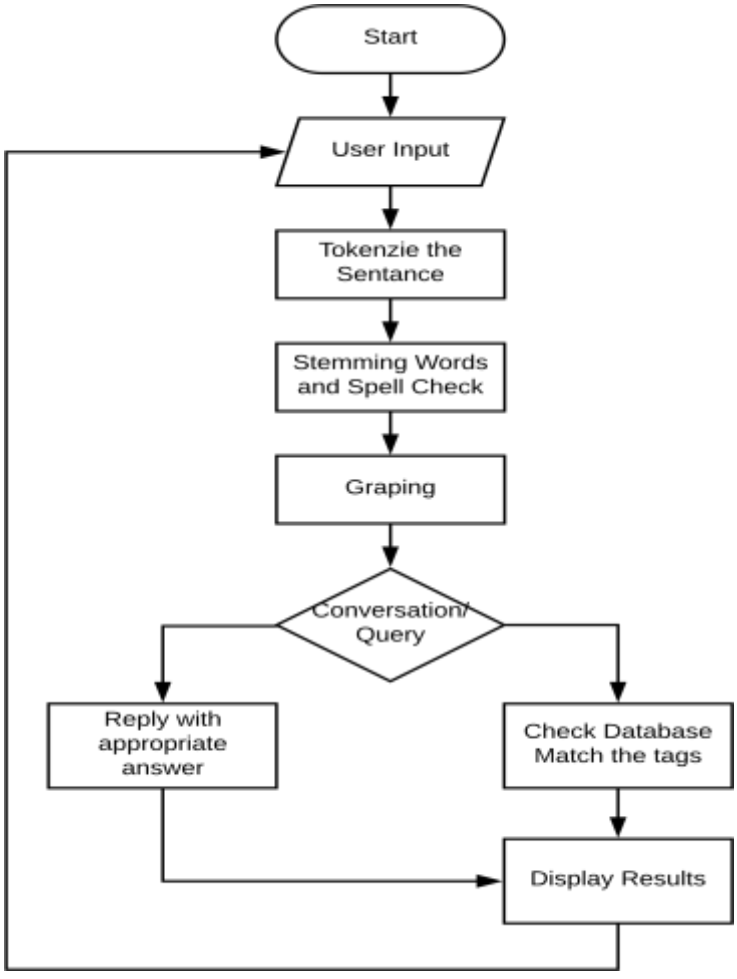


Fig 6.2: Dataflow Diagram

- The expense priorities are checked and a flag is set or unset upon comparison and prioritization.
- Based on flag value the bot responds to the user with an easily understandable chat response.

7. SYSTEM DESIGN

The system design specification includes input design specification and output design specification.

7.1 User Interface Design

Input design is a method toward converting client input into an android understandable based format. Information is gathered and sorted out into groups of comparable information. Once recognized, appropriate information is chosen for processing this information and are planned to utilize GU controls. In this way, the mistake might be kept away from.

Platform

Mobile applications (or 'apps') have seen a surge in popularity since the release of smartphones and are a widespread target for small pieces of software, such as financial organization**.** Our model is based on working on the Android OS and is designed to be a mobile application. We are using the Android Studio which is the officially integrated development environment for Google's Android operating system for building our application.

Android Studio offers:

Adaptable Gradle-based form framework, Code formats to enable you to fabricate normal application highlights, Rich design proof-reader with help for simplified subject altering, Build up devices to get execution, ease of use, rendition similarity, and different issues and much more.

The Android application we designed is written in Java. The Android SDK gathers the code alongside any information and asset documents into an APK, an Android bundle, which is a chronicle record with a .apk postfix. One APK document contains every content of an Android application and is the record that Android-fuelled gadgets use to introduce the application.

Login Form

This form is used by target audience to log in to the system using user-id and password. Thus, it provides security to the system. It also has the provision to change the password of the user who has currently logged in.

Chat Interface Screen

This screen is used by the user to interact with the bot providing a user-friendly texting interface. It provided a one to one interaction domain where for each request it provides a suitable response in a text format. Through this screen, the user can provide his/her expense details and asks for suitable suggestions to manage expenses.

Fig 7.1: The start-up interface of the Fynbot application

7.2 Analysis Module

The output design phase is another very important one. The outputs are mainly used to communicate with the user, processing the input data given by the user etc. It is documented at each stage of the project to ensure error-free output.

IBM Watson

Natural Language Processing

The initial step is Question Analysis, in which Watson runs regular dialect handling procedures to order the issue, including shallow parsing, profound parsing, semantic part marking, gathering relations, and named-element acknowledgment, the greater part of which are talked about in the segment on characteristic dialect preparing (NLP). Deep parsing systems in Watson include taking any information sentence and performing tokenization and division, trailed by various phonetic investigations (e.g. morpho lexical and syntactic investigation). Tokens are changed over into single word expressions, and afterward additionally handled into groupings of various word states that convey meaning. Through this procedure, Watson can separate a sentence into its segment parts, mark each, and comprehend the utilitarian and linguistic centrality of each for use accordingly age.

Response Generation

The second step is Hypothesis Generation. In view of its comprehension of the inquiry, Watson looks its outer information bases for an answer and creates all conceivable substance that could contain the appropriate response. Pursuit methods include: content web indexes, information base hunt, archive inquiry, and entry seek and incorporate various inquiry inquiries per question.

The third step is hypothesis and evidence scoring, in which the framework gathers proof on every speculation, and applies a positioning calculation to decide the no doubt reaction. Scoring capacities measure: "the level of match between a section's predicate-contention structure and the inquiry, entry source dependability, transient connections, ordered grouping, the lexical and semantic relations the candidate is known to participate in, the applicant's connection with question terms, its fame and so on.

The fourth step is ranking and confidence estimation. Watson utilizes a machine learning way to deal with accomplish this progression; it takes in the above measurements as highlights and prepares a classifier on preparing information with known right answers, along these lines making weights for each measurement. Given the assortment of measurements, distinctive classifiers are relevant to each. To take care of this issue, Watson trains isolate classifiers to deal with sub-sets of highlights, and afterward utilizes a gathering of these classifiers to produce a general positioning.

Knowledge Base

Watson's knowledge, essentially, is produced through access to an accumulation of online records through two stages: Indri and Lucene. Getting to these stages, Watson can combine inquiries with report sets, and concentrate data containing the appropriate response either from the titles of the archives of from the substance of the records.

Lucene is in like manner a content internet searcher library. Lucene gives positioned looking, "express inquiries, trump card questions, nearness questions, run inquiries, handled seeking, multiple indexes seeking with combined outcomes, and a configurable capacity motor.

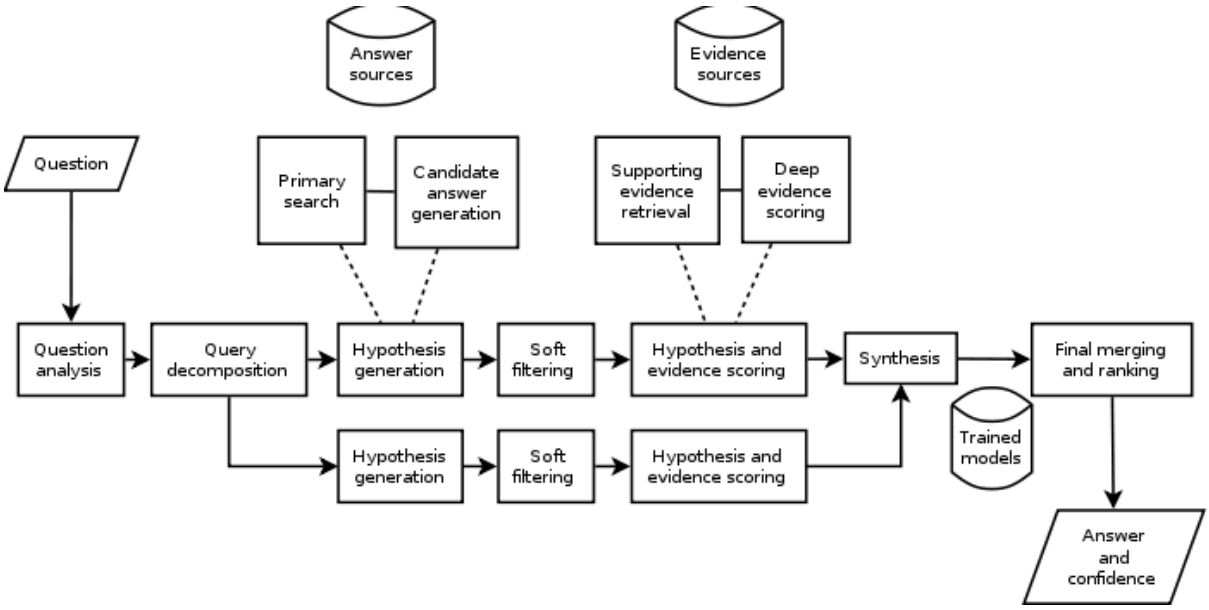


Fig 7.2: Watson's Information Retrieval Process

8. MATHEMATICAL MODEL

8.1 Probability of an observed sequence

The task is to compute in the best way, given the parameters of the model, the probability of a particular output sequence. This requires summation over all possible state sequences:

The probability of observing a sequence

$$Y = y(0), y(1), \ldots .., y(L-1)$$

of length, L is given by

$$P(Y) = \sum P(Y|X)P(X),$$

where the sum runs over all possible hidden-node sequences

$$X = x(0), x(1), \ldots , x(L-1),$$

Applying the principle of dynamic programming, this problem, too, can be handled efficiently using the forward algorithm.

This is used to predict the queries by relating the observations and identifying patterns from the sequences.

8.2. Poisson hidden Markov model:

Poisson hidden Markov models (PHMM) are special cases of hidden Markov models where a Poisson process has a rate which varies in association with changes between the different states of a Markov model. PHMMs are not necessarily Markovian processes them because the underlying Markov chain or Markov process cannot be observed and only the Poisson signal is observed.

9. SOFTWARE TESTING

Software testing is an examination led to furnish partners with data about the nature of the item or administration under test. Software testing can likewise give a target, autonomous perspective of the product to enable the business to acknowledge and comprehend the dangers of programming usage. Test methods incorporate the way toward executing a program or application with the plan of discovering programming bugs (blunders or different deformities) and checking that the product item is fit for utilize.

Software testing includes the execution of a product segment or framework segment to assess at least one properties of intrigue. By and large, these properties demonstrate the degree to which the segment or framework under test:

- meets the necessities that guided its outline and improvement,
- responds accurately to a wide range of sources of info,
- performs its capacities inside a worthy time,
- is adequately usable,
- can be introduced and keep running in its expected surroundings, and
- Achieves the general outcome of its partner's want.

9.1 Test Case

A Test case is a record, which has an arrangement of test information, preconditions, expected outcomes and post conditions, created for a specific test situation keeping in mind the end goal to check consistence against a particular necessity.

Test case goes about as the beginning stage for the test execution, and subsequent to applying an arrangement of info esteem; the application has a complete result and leaves the framework at some end point or otherwise called execution post condition.

Fig 9.1(a): Conversation between the user and bot using text input. (Salutation)

Fig 9.2(b): Conversation between the user and bot using text input. (Enquiry and result)

The above image depicts a conversation between the bot and client. The bot identified the presence of interaction triggered by the user through text input. As the conversation progressed the financial jargon was easily parsed and identified, the bot responds with appropriate answers to queries related to budget and expense management. The overall experience of the chat remained simple and precise.

Tests were conducted and the simulations helped us make the following observations:

1. The bot was able to identify the user and converse even when conditions were improper (like wrong spelling, irregular context, and incorrect sentence formation).
2. The UI/UX is simple yet responsive making it easy to handle without knowing many features.
3. Text-to-Speech can also come handy when responses from the bot are too huge and can be time taking to read completely. Allows the user to go hands-free once the query is submitted.

10. RESULTS AND DISCUSSION

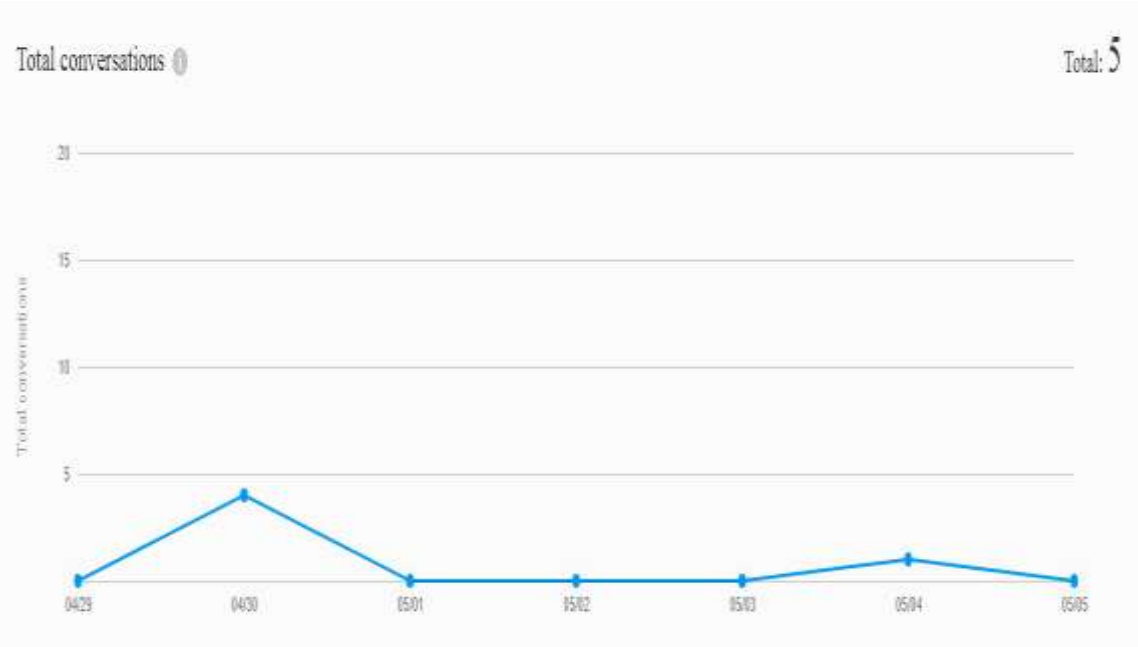


Fig 10.1: The graph depicting total number of conversations of a user

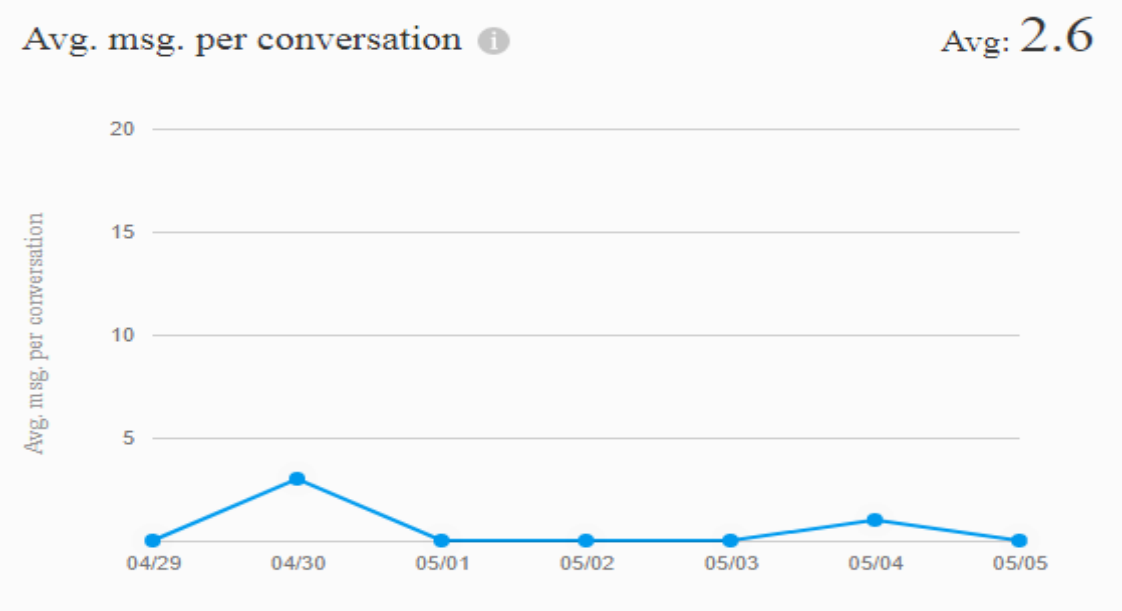


Fig 10.2: The graph depicts the average number of messages per conversation for user

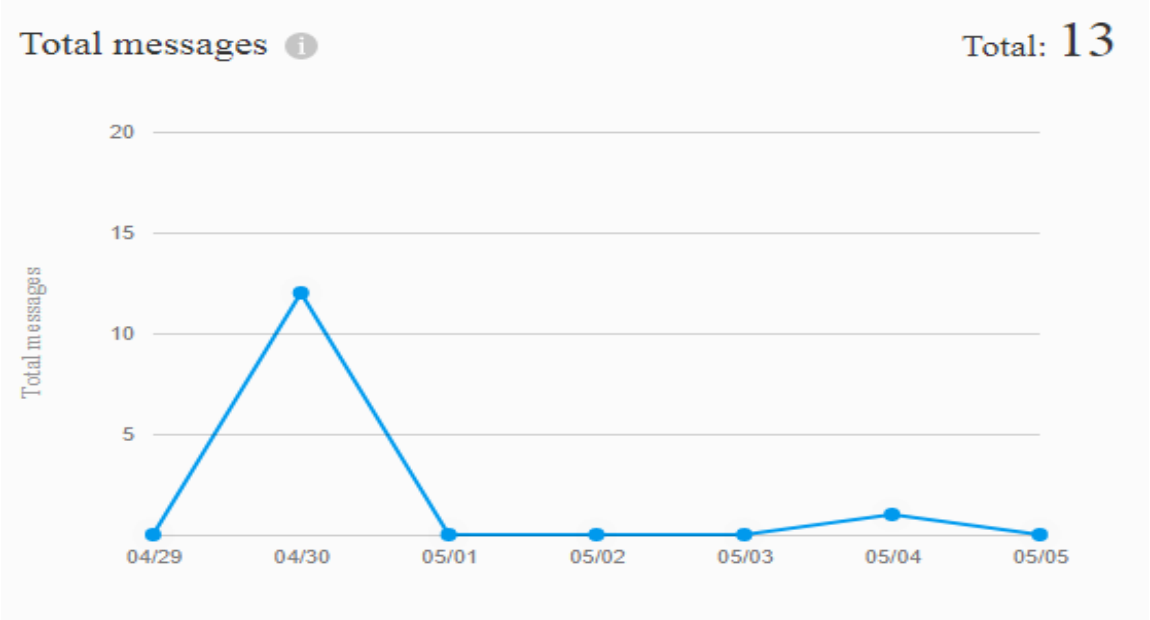


Fig 10.3: The graph depicts the message exchanges from the user end.

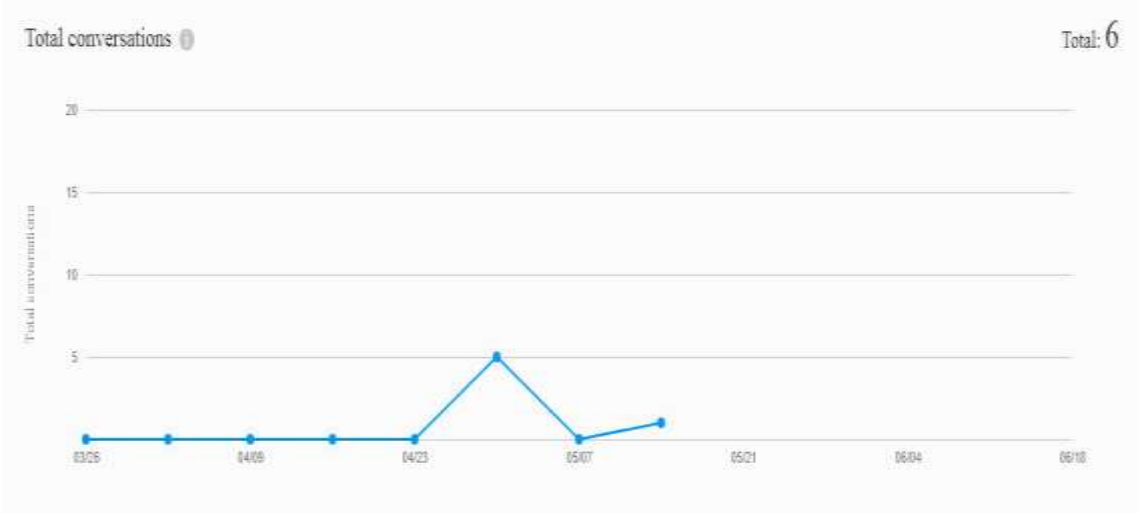


Fig 10.4: The graph depicting total number of conversations of a bot

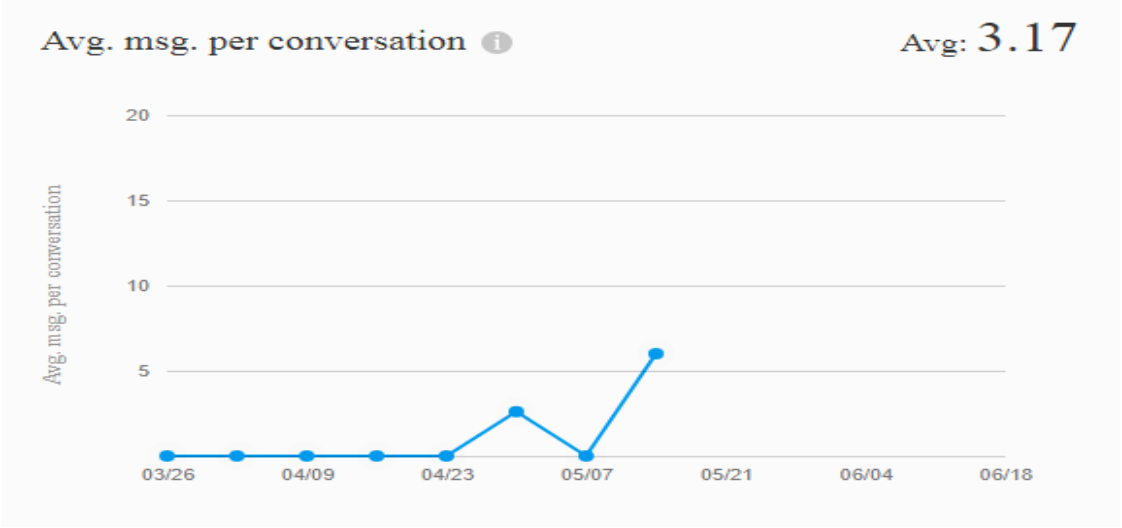


Fig 10.5: The graph depicts the average number of messages per conversation for bot

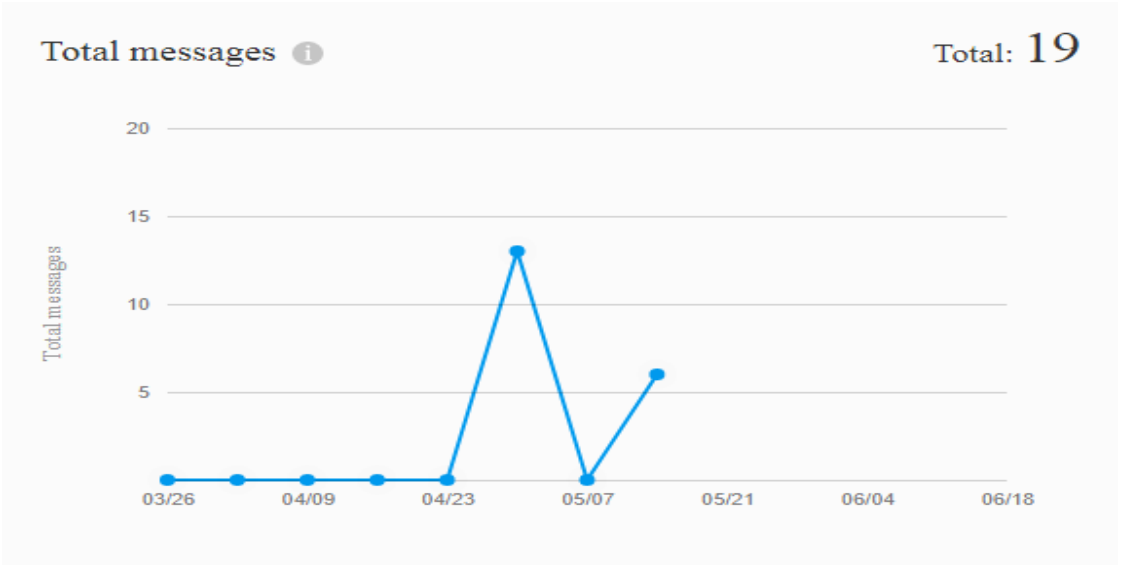


Fig 10.6: The graph depicts the message exchanges from bot

11. CONCLUSION AND FUTURE SCOPE

Conclusion

Chatbots are the future. Everyone should be ready to integrate one with their business. With so many chatbot platforms at our disposal, it has become extremely easy to build a bot. Fintech application has made their place in the market (17.2 Billion USD). Fintech applications are much needed in today's world because the majority of people might be educated and literate but they end up being financially illiterate, being financially literate will help the people make smart decisions in the financial part of their lives. Usually, analytics is used in almost all the fields today, so why not use it in the financial sector with a little touch of AI. With our fintech application (FynBot), people would be able to interact with our smart AI. Our AI uses special algorithms defined by us to analyze and respond to the input given.

Future Scope

a) Audio Input Tweaks

In the existing version of the application Text-to-Speech and Speech-to-Text APIs are incorporated which enables users to give another type of input.

The inclusion of audio input would increase the ease in using the application. The overall appeal of the application would be enhanced.

The major concern with respect to the audio input is the accent detection; accent differs from person to person making it very difficult to distinguish certain pronunciations. With improved APIs with respect to pronunciation and variable accent detection, we can ensure better experience by incorporating the audio input.

b) Accessing data points from multiple existing financial sources

The analytics of the bot can be improved if data points can be identified within existing financial services.

Applications that track trends related to expenses, debt management, and money management can be used or linked to the bot with APIs to obtain data points in the form of metadata that can help the analysis procedure to reach an improved and advanced stage.

This procedure would not only improve the processing but will also enable efficient data storage, by reducing redundancy.

The services chosen for linking must also be trusted if the data points considered or obtained are sensitive in nature such as bank details and personal client data.

