CLOUD COMPUTING FINAL PROJECT

SHOPYST CHATBOT

Team:

Sai Shethabhish Naidu Palla

Veerabhadra Kartheek Vatti

Aishwarya Oza

ABSTRACT

Chatbots are software programs which are primarily modern web-based dialog systems used to interact with users, in real time, through a textual or audio medium. Chatbots use natural language processing coupled with artificial intelligence to engage users in meaningful personalized conversations. Chatbots can understand the customer needs, look through catalogs and websites, compare prices and resolve frequent queries. Chatbots are either part of a web portal or are integrated with the messenger applications.

E-commerce consumers want a fast, easy and self-serviceable medium to perform online transactions with a way to answer their queries instantly. Consumer engagement is important for retailers and e-commerce platforms as it increases sales. As per a research conducted by DigitasLBi, 33% of the Americans are willing to make a purchase through a chatbot and would easily spend around $55.80 per purchase. More than 50% of the Americans (59%) have or are willing to communicate with chatbot to receive offers and coupons (36%), get recommendations or suggestions (37%) or conduct payments and transactions

(14%).

In this project, we propose an artificial intelligence based personal assistant chatbot for e-commerce platform consumers. We will be utilizing artificial intelligence for our chatbot to scan the web, learn about user preferences, purchase history and habits, selecting and displaying the most appropriate information to a user. Important features supported by our chatbot will include filtered product search, product purchase, order tracking and history, providing personalized promotions and offers. The outcome will be a chatbot which will allow e-commerce platforms to increase user engagement and retention by providing personalized conversational commerce.

**Table of Contents**

[Chapter 1. Project Overview 1](#_Toc33099)

[Introduction](#_Toc33100)

[Proposed Areas of Study and Academic Contribution](#_Toc33101)

[Current State of the Art](#_Toc33102)

[Chapter 2. Project Architecture 4](#_Toc33103)

[Introduction](#_Toc33104)

[Architecture Subsystems](#_Toc33105)

[1. Chat Client](#_Toc33106)

[2. Chatbot Core Engine](#_Toc33107)

[3. Knowledge Base](#_Toc33108)

[4. Platform Integration API](#_Toc33109)

[5. Product API](#_Toc33110)

[*Chatbot core engine*](#_Toc33111)

[User reviews/comments sentiment analysis](#_Toc33112)

[Response Generation Module](#_Toc33113)

[Product and Platform integration API](#_Toc33114)

[Chapter 3. Technology Descriptions 15](#_Toc33115)

[*Chatbot*](#_Toc33116)

[*Recommendation engine*](#_Toc33117)

[*Sentiment analysis*](#_Toc33118)

[*Data-Tier Technologies*](#_Toc33119)

[*Ecommerce Database*](#_Toc33120)

[Chapter 4. Project Design 18](#_Toc33121)

[*Use Case Diagram*](#_Toc33122)

[Chapter 5. Project Implementation 24](#_Toc33123)

[Chapter 6. Performance and Benchmarks 31](#_Toc33129)

[Chapter 7. Deployment, Operations, Maintenance 37](#_Toc33130)

[Chapter 8. Summary, Conclusions, and Recommendations 38](#_Toc33131)

[Summary](#_Toc33132)

[Conclusions](#_Toc33133)

[Recommendations for Further Research](#_Toc33134)

[Glossary 40](#_Toc33135)

[References 41](#_Toc33136)

# Chapter 1. Project Overview

## Introduction

Over the past few years, e-commerce has grown from a mere fascination to an important part of people’s lives with over $2 trillion retail sales market in the year

2017. Research predicts a total e-commerce retail sale of over $4 trillion by the year 2020. Today about 70% of the consumers are shopping online to get the best products at best price. With the amount of competition and growing number of ecommerce platforms, it has become important for the online retailers or ecommerce platform owners to provide personalized services and help to their consumers. As per a research study was done by Customer Care Measurement & Consulting, 54% of the retail customers are not satisfied with their customer buying experience. Today’s consumers want fast, easy and self-serviceable ways to perform different transactions and get their questions answers instantly. Considering the demands of the current consumers, retailers need to find out and adopt innovative ways to provide fast and personalized services. One such way is ‘chatbots’.

Chatbots are software programs which are used to interact with users and consumers over e-commerce platforms. Chatbots can use either textual or audio medium for having conversations with the customer. Such chatbots can be used by e-commerce platforms to provide personalized buying experience for the consumers, thereby making them more satisfied. Sadly, chatbots till now have been built based on fixed decision paths, for specific scenarios and questions. This limits the quality of customer’s buying experience to a large extent.

As a part of our project, we will be building an artificial intelligence based chatbot with an inbuilt capability to learn through the consumer interactions over a period of time and thereby improving the quality of service during future conversations. The chatbot will be using artificial intelligence to study and learn from a consumer’s purchase history, web search, preferences, likes and dislikes to provide personalized buying experience on a given e-commerce platform through intelligent humanly conversations curated through natural language processing. Integrating such chatbot will provide huge benefits to modern e-commerce platforms. It will reduce consumer service costs, will improve consumer engagement on the platform, will provide highly personalized and enhanced buying experience to consumers in simpler, easier and faster manner.

## Proposed Areas of Study and Academic Contribution

Each one of us expects a bot to provide us with the most accurate information while we are interacting with it, and this is one of the main factors which contribute to the success of a chatbot. For a chatbot working as a personalized shopping assistant, this essentially means that it can figure out what its user is expecting, and synthesizes the best possible recommendations after crawling through e-commerce platforms. The goal of this project is to understand the impact of introducing chatbots in e-commerce platforms and study the user behavior.

## Current State of the Art

The “customer shopping pattern analysis apparatus” is a novel idea which was developed for in-store shopping analysis. Our solution of utilizing this idea in a chatbot for e-commerce platforms, for providing it artificial intelligence capabilities is a novel method which can greatly improve the chatbot’s capabilities in user’s behavior mining, resulting in making better recommendations and predictions. The correlations generated by our chatbot overtime will also help the chatbot to adapt to different user needs on a user-by-user basis.

We plan to couple this with the mining algorithms such as L2AP and L2Knng, which will boost the performance of the bot when it is going through huge product catalogues for finding the best matches based on the user inputs.

The above stated techniques will not only be used for recommendations, but will also contribute to the improvement of other features such as finding deals and promotions, customized offers etc.

We will also be incorporating some other features in our chatbot such as: x automated cart addition and checkout.

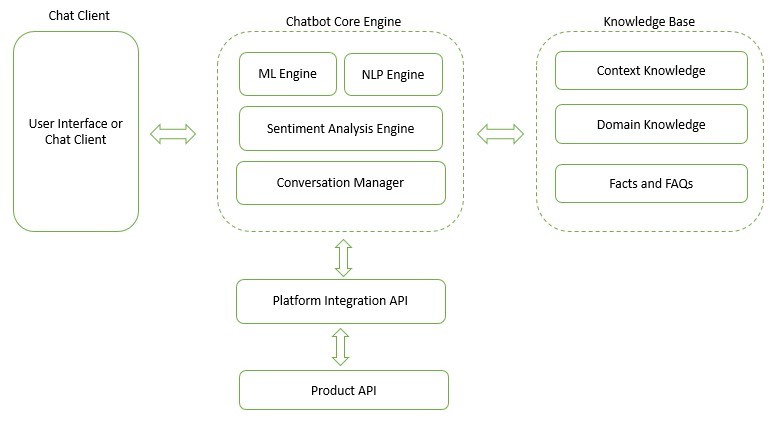
x automated payments

Using these techniques, we aim to make our chatbot effective in assisting online shopping and having a seamless shopping experience.

# Chapter 2. Project Architecture

## Introduction

The chatbot architecture consists of multiple sub-systems as described below:



*Figure 1 Project Architecture*

## Architecture Subsystems

As shown in the diagram, our project constitutes of five important modules. Each of these modules is described briefly below:

## 1. Chat Client

This is the module which is used by the consumers / users to interact with the chatbot and perform different transactions on the e-commerce platform. This module provides the chat window wherein the consumer can initiate the conversation, interact with the chatbot, ask queries etc. This module can be made available on mobile platforms through messaging applications and on web portal as an integrated facility. Few examples of chat client include Facebook messenger, slack, web site live chat windows etc.

## 2. Chatbot Core Engine

This is the most important module of our project architecture. It is responsible for controlling the conversation with the consumer, using machine learning models to learn about the user's behavior, choices, likes and dislikes. Additionally, it has a NLP Engine which is responsible for understanding the user inputs and providing appropriate responses to build a conversation. The sentiment analysis engine part of this module is used for understanding the sentiment and emotions of consumer during the conversation. This plays important role in increasing customer engagement through empathetic conversations.

This engine will interact with User API to enable user authentication, saving user conversation history and personalized preferences.

## 3. Knowledge Base

This module acts as a back end for our project. It supplies the core engine with all the knowledge required during the conversations. This includes domain specific knowledge like reviews and ratings for a product, contextual knowledge and knowledge about the facts and frequently asked questions. Chatbot Core engine uses this module in parallel to derive the correct responses to the user queries.

## 4. Platform Integration API

This module provides a set of API which can be used to integrate our chatbot with a third-party ecommerce platform. The primary purpose of this module is to interact with vendor specific product APIs to fetch different products, filter products, get product prices etc. It's the middleware between our chatbot core engine and the vendor specific product API.

## 5. Product API

This module represents the vendor specific product API which are made publicly available by the e-commerce platform. This API provides the enables the product search, product fetch, filter, order history, order tracking through simple service calls. This module is maintained by the ecommerce vendor separately.

The following text describes each of these components in depth, along with necessary implementation details.

## *Chatbot core engine*

Chatbot core engine is the central and most important module in the successful implementation of our project. This module is responsible for taking input from the client module, sanitizing the user input and processing it through the natural language processing pipeline to have correct interpretation of the sentence and produce the appropriate response. This engine is also responsible fetching user feedbacks and comments for the product, performing a sentiment analysis on the user comments to find the pros and cons about the product. It finds out the complaints and problems which current customers have faced during their personal use of the product. Additionally, this module also derives the tone and sentiment of the customer during conversation. This helps in adopting different strategies of conversation and changing the response during the conversation to make the conversation empathetic.

The third and one of the most important module part of chatbot core engine is a machine learning module. This module is responsible for training the defined machine learning classifier through continuous user interaction over the platform. The conversation manager is responsible for receiving, storing and managing conversations with customers. It makes use of the response templates to respond to customers in the natural way.

The architecture for these sub modules are shown below:

Input

Sanitization

Input

Segmentation

Tokenization

POS Tagging

Entity Recognition

Relationship Search

*Figure 2 NLP Engine Architecture*

The architecture of a NLP pipeline which we will be integrating in our project is as shown in figure 4. NLP pipeline is series of input processing steps which lead to understanding the meaning of the user input. At the start of the pipeline is the input sanitization module. The primary responsibility of this module/step is to perform spell checks and auto corrections on the user input. It splits the user input into a set of sentences to simplify the processing easy for the following modules/steps in the

NLP pipeline. The next two steps in the pipeline, Input segmentation and tokenization, divide the sanitized sentences into words to simplify processing in later stages of pipeline. All the words are classified into different classes of morphemes.

POS tagging is responsible for determining the part of speech for each word derived from former stages. Same word might refer to different meaning based on its placement at position in a sentence. POS tagging helps to disambiguate between multiple meanings for a word. For example, consider the word ‘charge’ when used as a noun and then ‘charge’ when used as a verb.

Lastly, entity recognition module is responsible for recognizing different entities part of the sentence. This includes dates, locations, people, streets etc. All these entities might be expressed in different formats and so needs to be converted to a universal format before processing further.

At the end of this pipeline, we get a set of clean and grammatically correct set of sentences which are further processed to find the intent of the sentence.

### User reviews/comments sentiment analysis

Cleanup and

Tokenizatio

n

Sentiment

Identificatio

n

Sentiment

Classificatio

n

Sentiment

Polarity

*Figure 3User reviews/comments sentiment analysis*

Sentiment analysis module consist of four sub modules which are responsible for extracting the sentiment from the customer review or comment. The cleanup and tokenization module takes care of the auto spell check and spell correct, it removes any unwanted strings and spaces. It divides into words and presents the set to next module, Sentiment identification. The lexicons generated in the previous stages can be compared against any lexicon dictionary like WordNet, SentiWordNet etc.

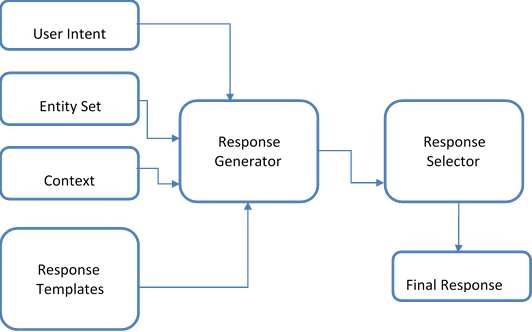
Once the words are identified against the lexicon dictionary, the sentiment polarity for each of the words and the resulting sentences is derived. The lexicon dictionary consists of terms and phrases which are user’s opinions. WordNet and SentiWordNet are two widely used lexicon dictionaries which can be used by a sentiment analyzer. The step by step process to perform sentiment analysis of a user comments is as given below:

1. Read user comment
2. Find the list of words in the comment which have a sentiment. This can be done using lexicon dictionary.
3. Calculate the sentiment score for each word with sentiment based on the number of times it occurs in the comment. The words which occur only once are assigned polarity with the highest sentiment score.
4. Finally, an average is taken of all the terms, for normalization, which has multiple occurrences to derive the sentiment score for the complete comment.

### Response Generation Module

Response generation model is used for generating a user specific response to suit the current tone of conversation. There are multiple ways to express the same thing. To provide a great value experience to users, the chatbot might need to tailor the responses to suit the conversation for specific users. Different metrics like length of conversation, customer rating, probability of getting orders can be used to dynamically create and provide responses during the conversation.

Response generation and response selection modules are developed separately to maximize the modularity during the implementation. The architecture for response generation module is as shown in figure 6 below.



*Figure 4Response Generation Module*

The response generator receives the set of entities which are generated from the user input, the context of the conversation and the predicted user intent from the above described modules. Intent identification module identifies the intent of the conversation. It uses the context of the current conversation, previous chat history related contexts, user preferences, wish list and search history. The entity recognition module extracts different entities part of the user input which is further used for sentiment analysis and understanding the context of the conversation too. This information is then used to generate a set of responses which are suitable for the user input. All these responses are filtered to be domain specific to avoid any random responses to be given.

These set of responses are then sent to response selector module to select the appropriate response. This module uses current context of the conversation to choose the most appropriate response. The response selector assigns a score to each of the responses and then finally decides upon a single response with highest score.

### Product and Platform integration API

The product and platform integration API are the components which are responsible for integration with specific e-commerce platforms such as Amazon, Walmart etc.

The platform API is responsible for handling:

x Authorizations and authentications while using services of specific vendors.

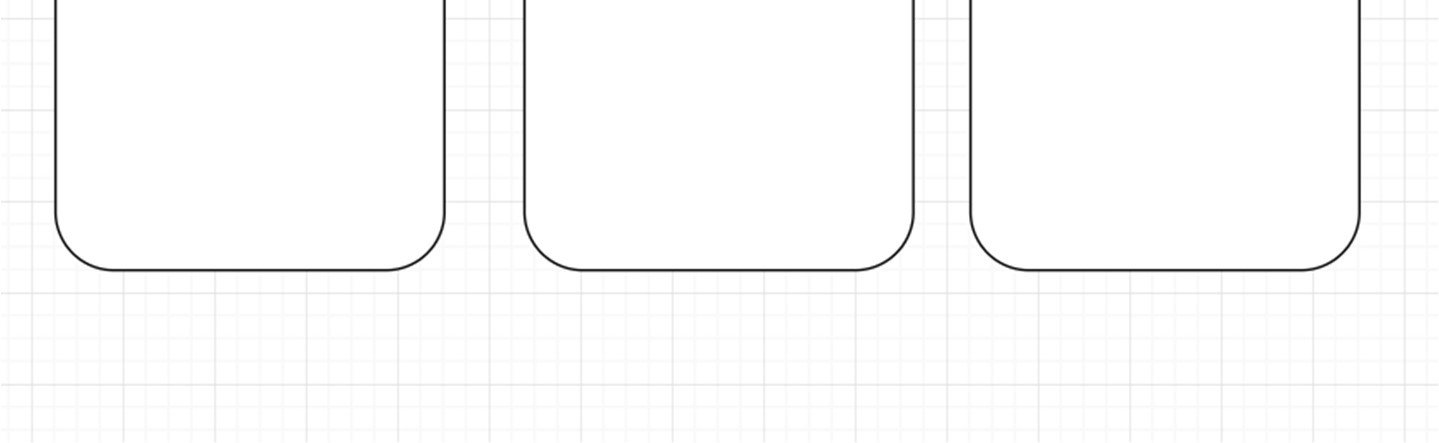
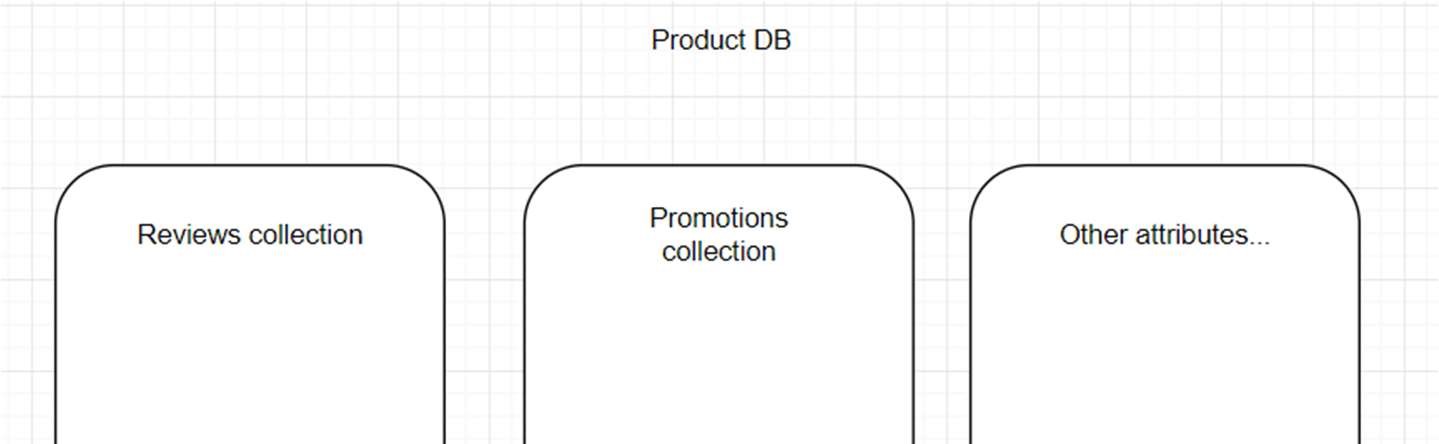
x When the customer is ready to purchase the items, the chatbot sends an HTML form to vendor API and the vendor completes the purchase by getting purchase information, such as payment method and shipping address. Vendor then fulfills the order by shipping the items.

The product API is responsible for handling:

x Access to the data used by vendor including the items for sale, customer reviews, seller reviews, as well as most of the functionality you see on the vendor’s website, such as finding items, displaying customer reviews, and product promotions. The product API will interact with the vendor services to achieve this.

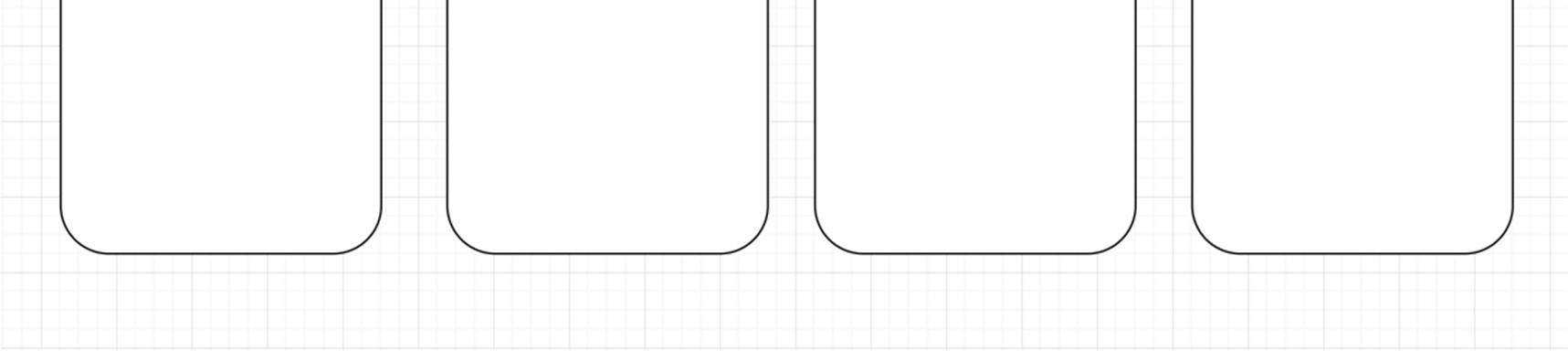
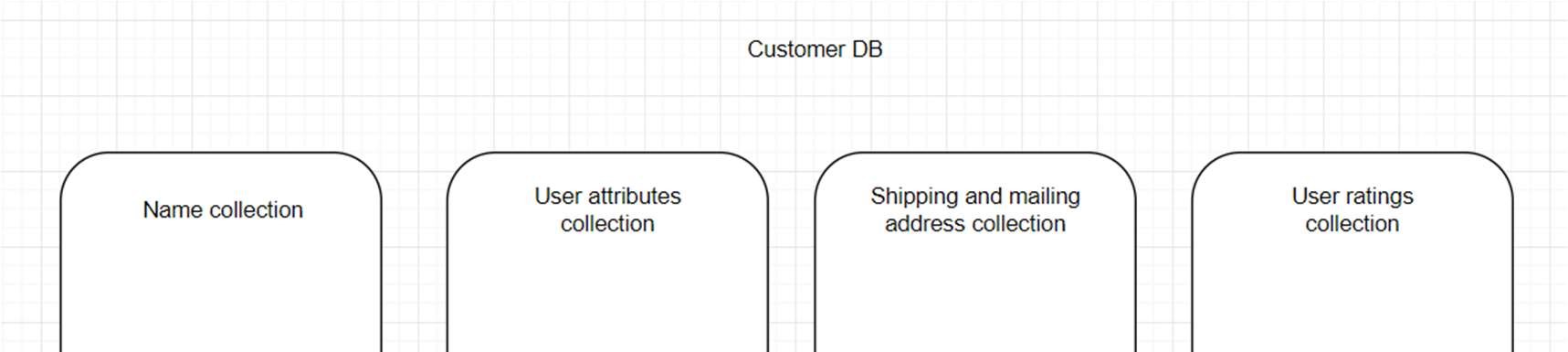
The product and platform APIs will have callable routines for every task mentioned above. All the product related data will be stored in a NoSQL database(Mongo). A background service in the database will remove all the product data after a specific amount of time.

Following diagram shows a high-level breakdown of how product related information would be stored in the database:



*Figure 5Product DB*

Customer related information is also stored in a different database in the following manner:



*Figure 6Customer DB*

# Chapter 3. Technology Descriptions

This project has 5 main components. This chapter will talk more about the technologies used in these components.

## *Chatbot*

The chatbot has three primary technologies used in it.

1. **Node.JS:**

Node.js is an efficient programming language for managing asynchronous I/O.

Node.js is fast because it is JavaScript and it has Event Loop in its heart. Also, with

NPM packages, speed of development can be increased.

1. **Microsoft Bot Framework**

Microsoft bot framework provides all the tools and libraries which are required to build the an intelligent chatbot which can have natural conversations with the user. It provides support for both development in node js and c#. It also supports integration with different services including slack, skype, office 365 etc.

1. **LUIS**

LUIS stands for a Language Understanding. It’s a natural language processing API built and provided by Microsoft. LUIS adds the capability to understand user statements captured during conversation by the chatbot. Using LUIS we can build models to identify different user intents and entities which are part of the statement. LUIS models improve with time by learning through conversations. This increases the accuracy and efficiently of the chatbot during conversation.

## *Recommendation engine*

**Mahout:**

Using Mahout’s GenericUserBasedRecommender the recommendation engine was written. It uses the TanimotoCoefficientSimilarity coefficient to determine similarity between users and k-nearest-neighbors to build recommendation list. It can be paired with any database as it does not keep record of user/item info.

## *Sentiment analysis*

**Python**

Python is a programming language that lets you work quickly and integrate systems more effectively. It is suitable for processing large amounts of data and working with vectors.

## *Data-Tier Technologies*

1.**Redis**

Redis is an in-memory data structure store, which is used for recommendation engine computation. It also supports most of the data types and sorted sets that are useful for the recommendation engine implementation.

2.**Mongo DB**,

MongoDB is a document database with flexibility that was can query. MongoDB store data in flexible, JSON-like document. This means we can change the fields document to document. Indexing, Ad hoc queries, and real time

aggregation provide amazing ways to analyze your data.

3.**PostgreSQL**

PostgreSQL is an open source object-relational database system. It has a proven architecture that has earned it a strong reputation for data integrity and reliability. It runs on all major operating systems, including Linux, UNIX and Windows. It is fully ACID compliant, has full support for joins, foreign keys, stored procedures, and views.

## *Ecommerce Database*

**Moltin**

Moltin is a platform which allows developers to create an complete ecommerce store on the cloud and provides API for seamless integration across multiple devices and applications. Few important features provided by moltin are –

1. Carts and checkouts

2.Inventory 3. Orders

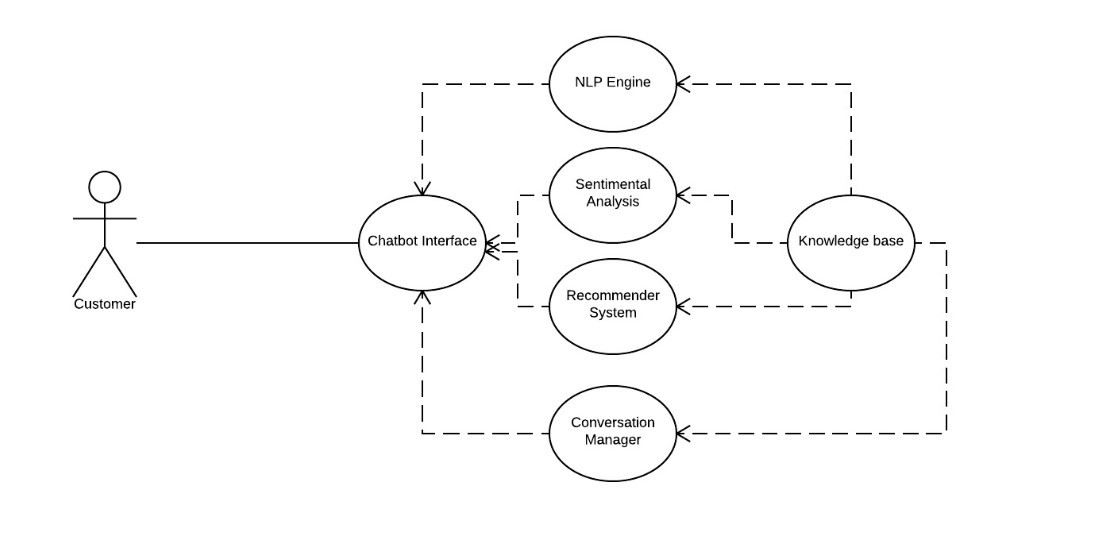
4.Payments

5. Content and schemas

# Chapter 4. Project Design

## *Use Case Diagram*

User will interact with the chatbot which is deployed on the shopping website. Chatbot



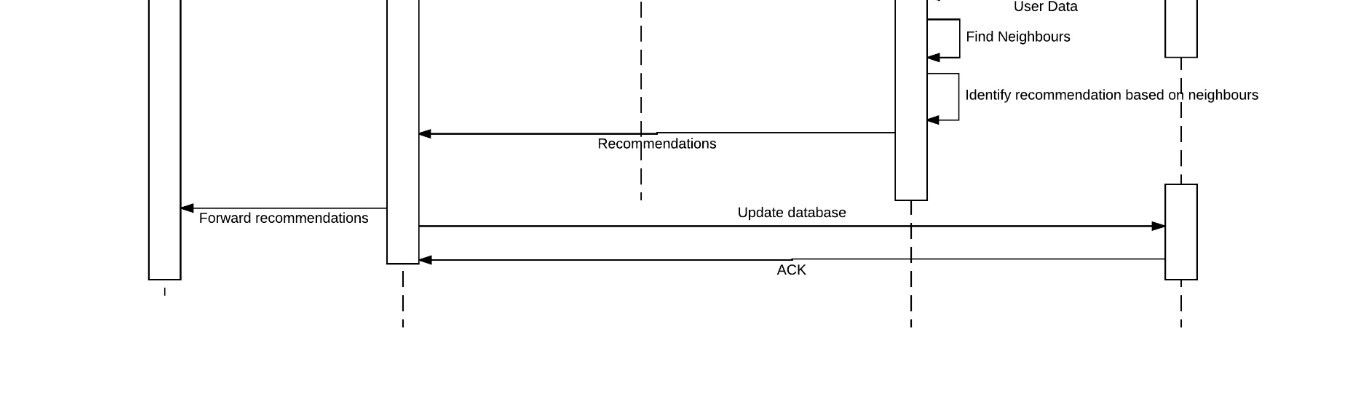
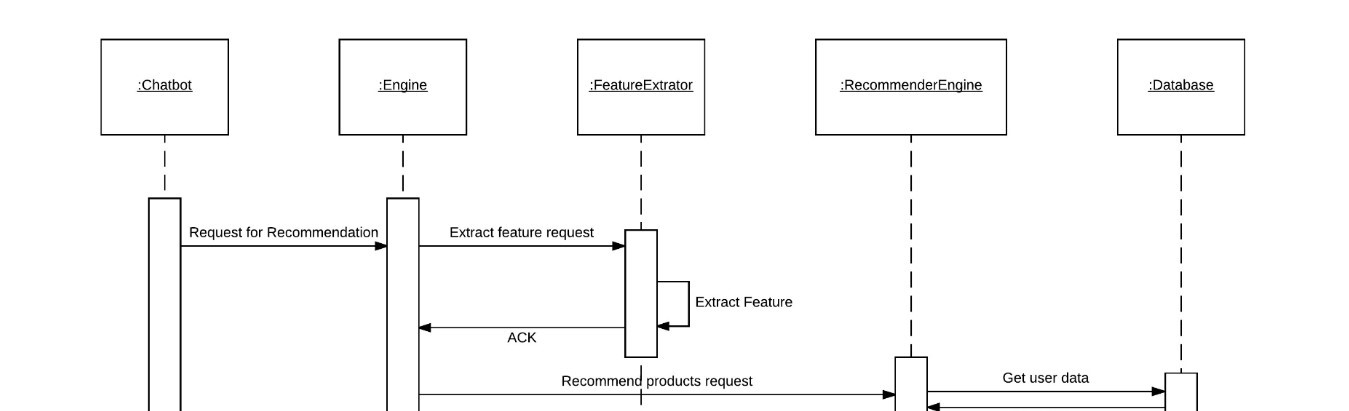
*Figure 7 Use Case Application*

Interface will be the point of contact for the user and This interface can pull information for other components.

NLP engine will help the system understand what the user means and what he really wants. Sentimental analysis engine will help understand weather the user will like the product based on the ratings and reviews the product has in past. Recommender System will predict the best products for the user based on User Based or Item Based Collaborative systems.

Conversation manager will decide how to frame the response and rate the system response based on the user feedback. This is critical to improve performance of the system overtime.

***Sequence Diagram***

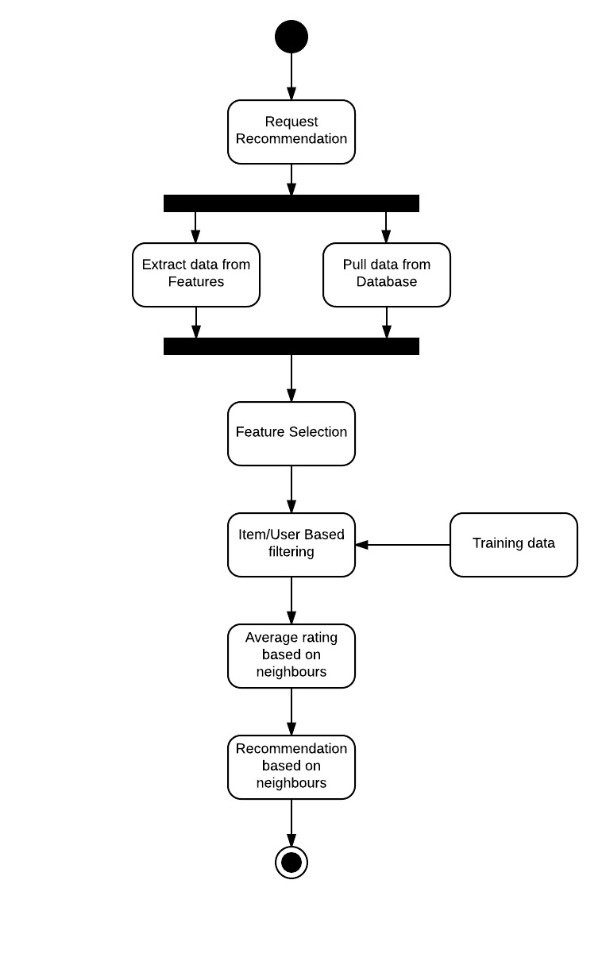


*Figure 8 Sequence Diagram for recommendation flow*

When user starts its conversation with the chatbot on the website the chatbot tries to understand with the customer is looking for. Once we get sufficient information from the customer the chatbot will send the request to the Engine and the engine will forward the request to Feature Extractor to understand the user inputs and convert it into recommendation engine input format.

After that the recommendation engine will identify K-nearest neighbors from that user input vector and recommend N products for the customer. The data will be sent back to the engine and engine will save the results in database as well as it will send back the results to the user.

***Activity Diagram***



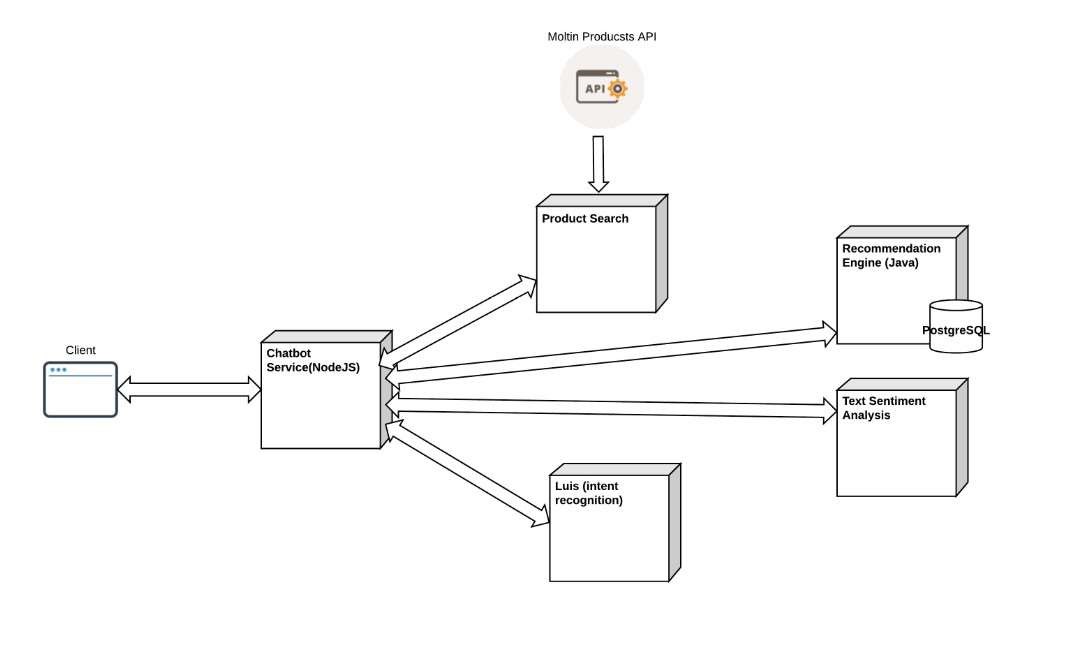
*Figure 9 Activity Diagram*

When user sends a request for the recommendation the request will trigger two things in parallel. First, the feature extractor will extract import information from the request and

Secondly, the engine will pull more data from the database regarding the same product.

After this process we will select necessary features for list and apply the collaborative filtering on it. After this we will get Neighbors from which we can get the average user rating for products. Based on this we can recommend highest rated product to the user.

# Chapter 5. Project Implementation



*Figure 10 System Architecture*

**NLP Engine**

The primary purpose of this module is to understand the customer inputs which are provided in human language i.e. English. This module reads the input from the chat interface, analyzes, understands and derive meaning from it based on the current context of conversation. The output of this module is then used for various purposes including sentiment analysis, named entity recognition, relationship extraction etc.

Implementation details of submodules in NLP Engine

1. **Input Sanitization**

The input provided by the user during the interaction with Chatbot might contain incomplete information, wrong data, special characters, spelling mistakes and other inaccuracies. It is very difficult to perform the intent classification on such user input. Performing intent classification directly on the raw user input might lead to a lot of false positives being accepted as valid inputs. Some of the examples of false positives user input are-

* 1. Spelling mistakes - ‘Orrder me a pizza’
  2. A keyword is a substring of the other word ‘Can I pay through my PayPal account?’
  3. ‘I made an order by mistake. I won’t pay. ‘

1. **Input Tokenization**

This feature of NLP engine divides the user input sentences into tokens. A token can be a simple word or a punctuation. A lexer function is used to generate tokens from user input. Tokens often contain special characters which are used as token boundaries.

1. **Part-of-speech Tagging**

Every word in a user input might have different meaning based on the part of speech i.e. noun, verb, adjective. This feature of ELP engine assigns part of speech to each of the words in the user input. Knowing the part of the speech can help to disambiguate the meaning of the sentence.

1. **Entity Recognition**

There are different entities like dates, values, places, numbers, proper nouns which can be part of the user input. This feature identifies the entities in the user input. Dates and values can be written in different formats. For example, ‘Nov 16, 2017’ and ‘11/16/2017’, ‘100,000’ and ‘100K’, ‘New York’ and ‘NY’ etc. This module detects and converts all the entities in the user input into a common universal format.

1. **Relationship Derivation**

This module is responsible for determining the relationship between the entities or events such as ‘treats’, ‘causes’ and ‘occurs with’.

**Recommendation System Model**

From the pre-processed data we will build a space vector for performance improvement.

Based on the selected KNN algorithm we will pre-calculate all the neighbors. And when the user input arrives we can predict K-nearest neighbors form that user inputs and recommend products based on the k-neighbor’s order history. After the model is ready, trained and tested, the Recommend system engine would be built as a pipeline around this prediction model in the following manner:

x The user data will be received as input and is preprocessed to generate feature vectors.

x The vector will be sent to the engine to k-nearest neighbors.

x The top products in the recommendation list are provided to the user as search

results.

**Sentiment Analysis Engine**

We need to build a model which would perform the sentiment analysis, and mark a review as positive or negative. After building the model, it needs extensive training using a large training dataset. Finally, the model needs to be tested for accuracy before we can use it in our engine.

After the model is ready, trained and tested, the sentiment analysis engine would be built as a pipeline around this prediction model in the following manner:

x The reviews for the mined products received as input are preprocessed to generate feature vectors out of the reviews.

x The reviews for each product are sent to the prediction model, one by one for each product, and based on the prediction of the reviews, a product is ranked in a recommendation list of products.

x The top products in the recommendation list are provided to the user as search

results.

**Conversation Manager**

**Category Based Question Set**

One of the important module is the search query builder module. During the conversation, the chatbot should be able to dynamically ask category based specific filter questions to user. LUIS can be used to identify the category, but the filter specific questions for category need to be defined and exposed separately. This will help in collecting the right information to build the correct search query, which then will be sent to the product search engine for further processing.

**Response Generator -**

Machine learning engine will help to understand the meaning of the user input. Sentiment analysis will provide the intent of the input. Also, there will be response templates, so with the help of all these factors response generator must generate a response which could be static i.e. one per intent or we can use predefined templates along with the intent, context and knowledge base to generate more intelligent Reponses.

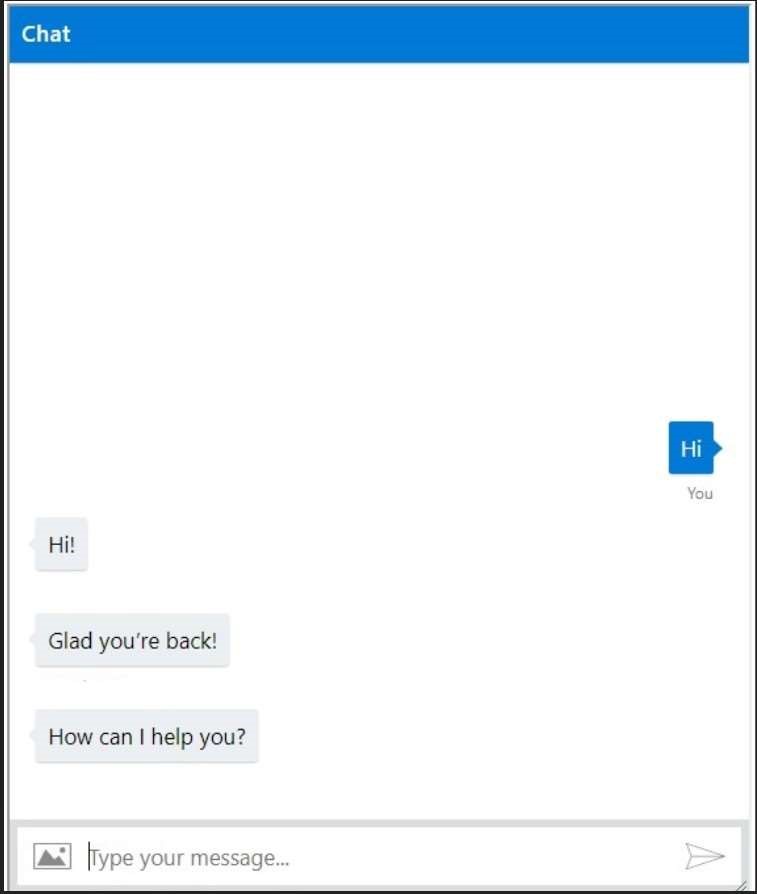
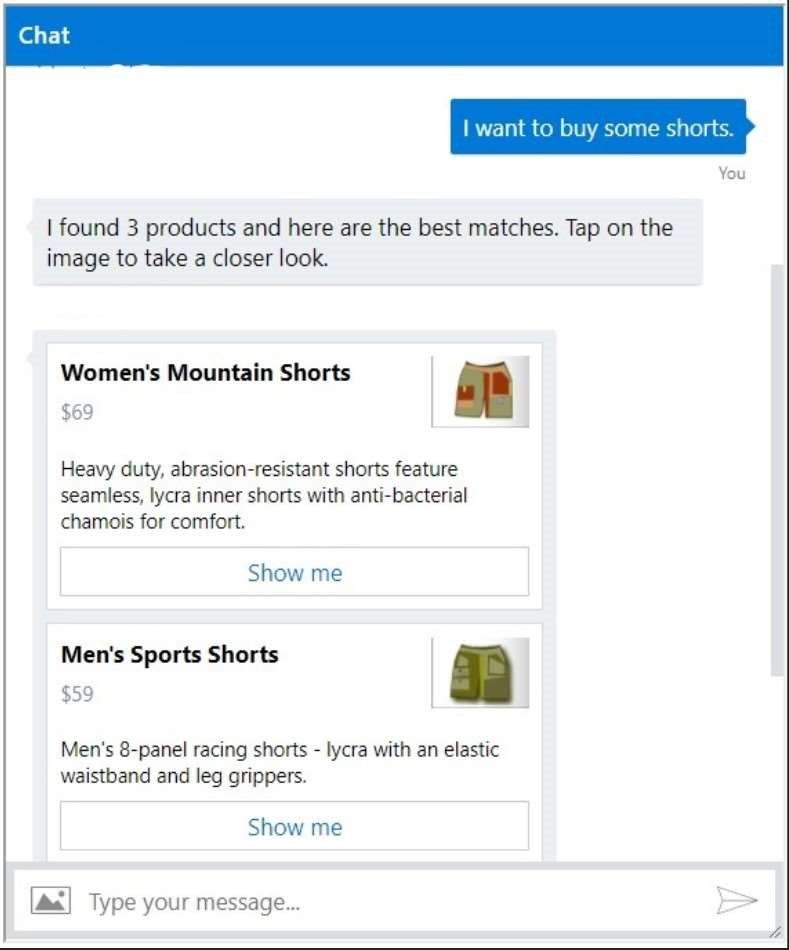
**Response Selector -**

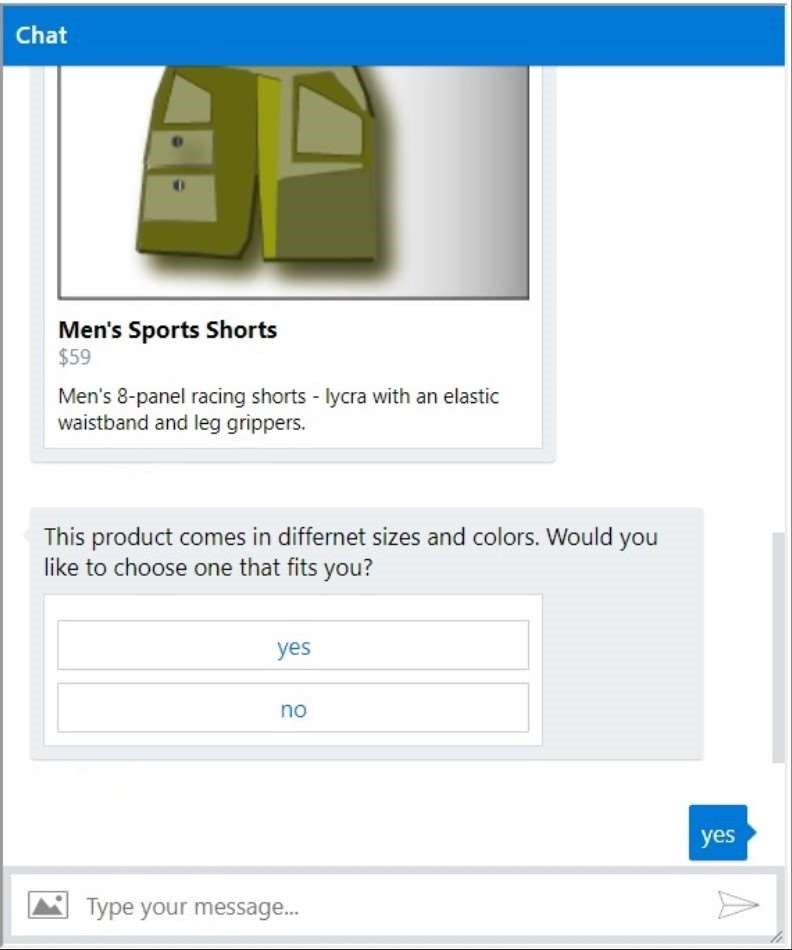
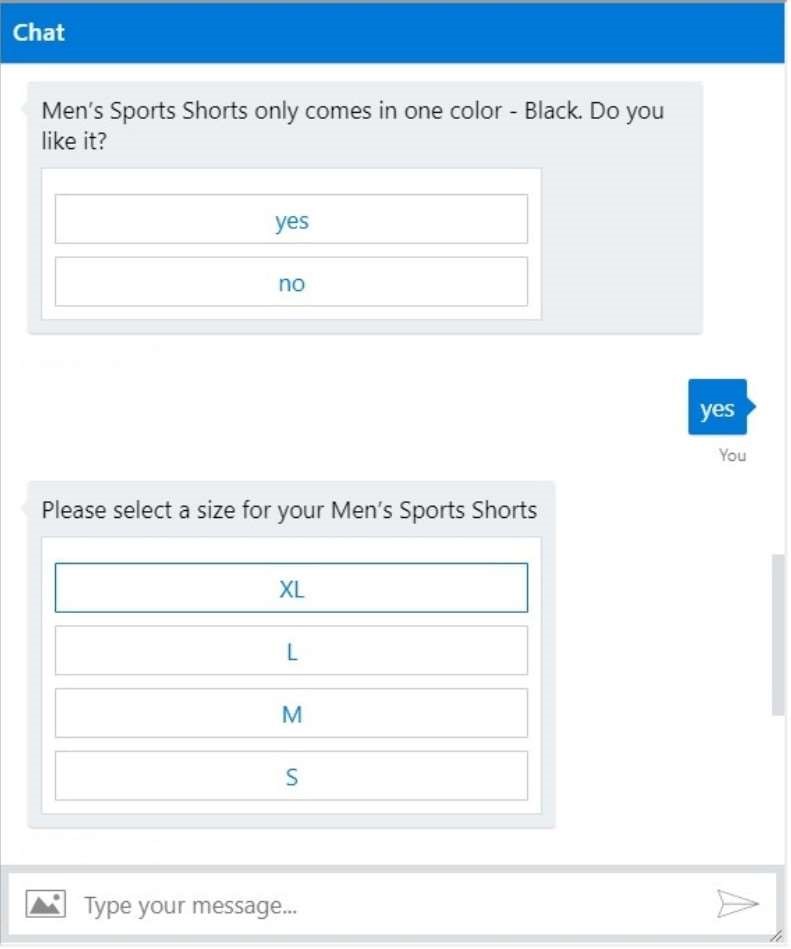
Chatbots response generator can come up with different responses for the same intent using different messages or responses. Response selector should analyze these responses. It can be done using previous conversation history from the knowledge base and several other metrics such as length of the conversation, probability of selling the product etc. Response selector gives score to each response generated by the response generator based on the above factors and selects the most appropriate one specific to the user.

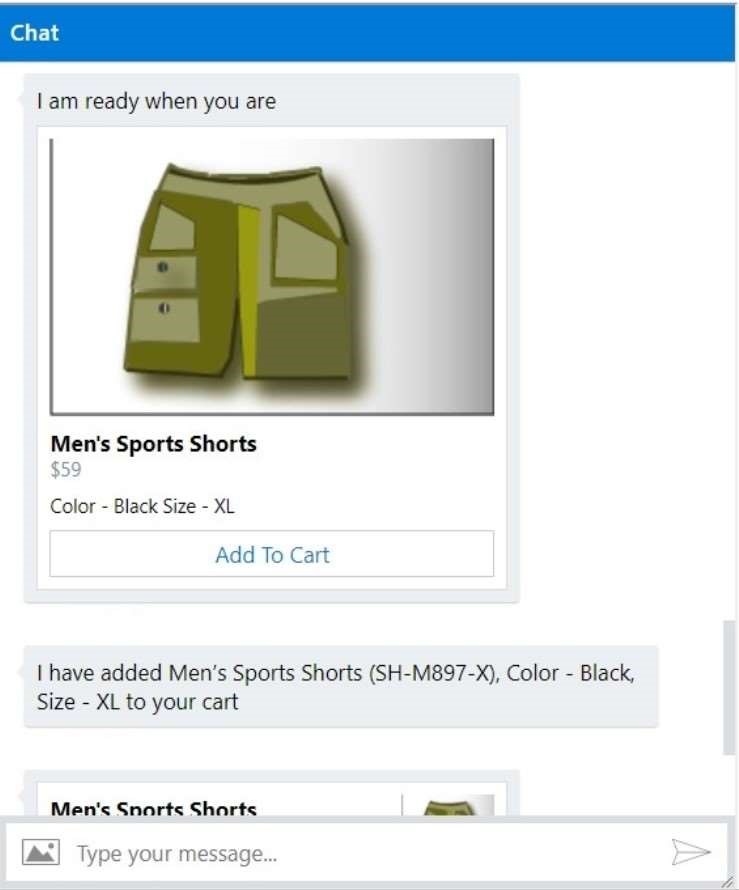
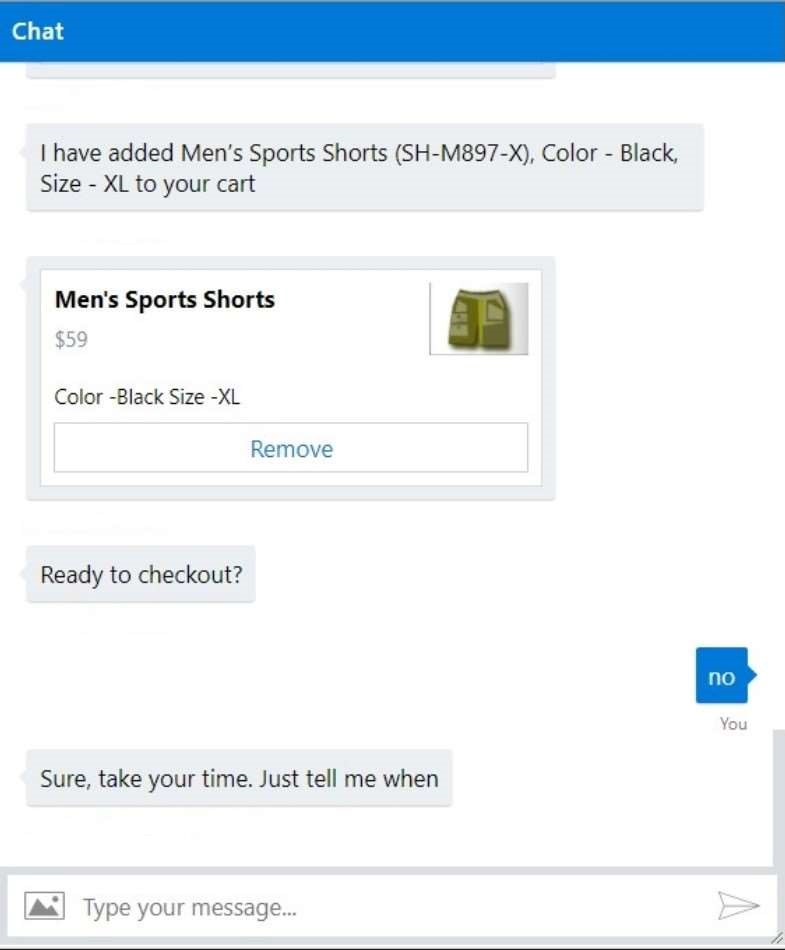
**Chat Interface**

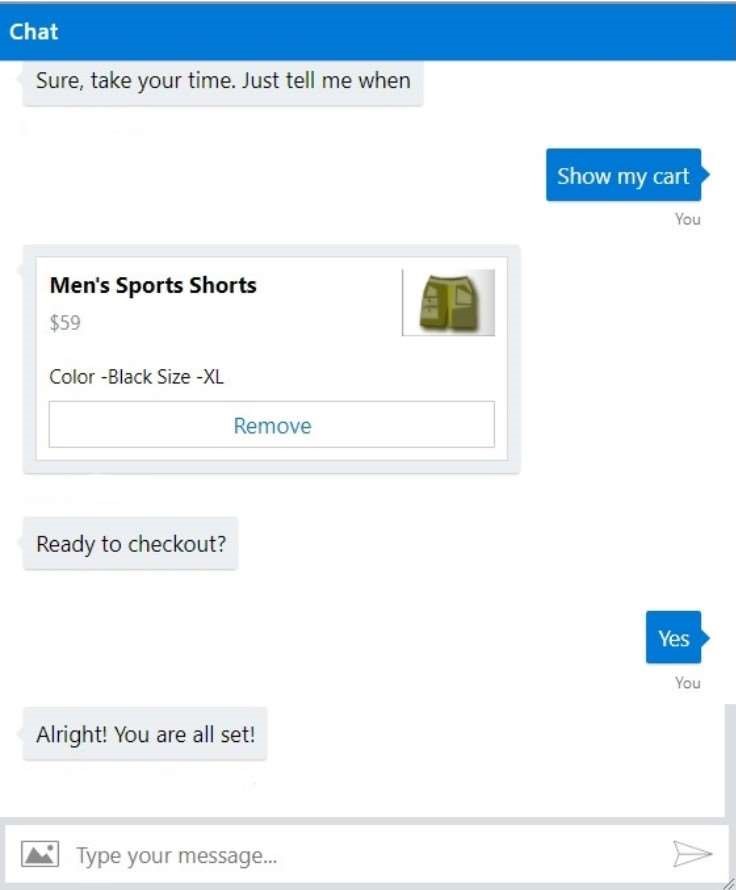
We have used a web-based chat interface to allow user to interact with the chatbot.

Screenshots of sample interaction are as given below:



*Table 1 Luis Testing*

# Chapter 6. Performance and Benchmarks

Following criteria are used in performance testing of our chatbot:

x **Satisfaction rate:**

This rate is determined by questioning about bot’s abilities from users and how satisfied the users were after using the chatbot.

We observed a high satisfaction rate for our project. For this, we asked 25 different students to use our chatbot and later asked them about their opinion. Out of 25, 19 people were satisfied with the bot’s responses to their queries.

x **Activation rate:**

In the context of a chatbot, the activation rate refers to a user responding to the chatbot’s initial message with a question or answer that is relevant to your business goals.

For example, a chatbot designed to provide you with weather updates would receive an activation rate when you enter your location — thus allowing the bot to provide you with the information. x **Confusion triggers:**

Even bots with the most robust natural language processing are unable to understand everything a user says.

These errors are a useful indicator for measuring whether you need to improve your chatbot’s matching. We rated the bot based on the replies it gave when in a confusion and whether those replies were in accordance with the context of the conversation. This helped us improve our chat client.

x **Speed**:

Speed is one of the primary criteria’s for benchmarking a chatbot, and in our case achieving a good response time was difficult due to the usage of multiple systems at the backend. Ultimately, we were satisfied by the average response time of our bot, which was under 7 seconds.

For our sentiment analysis and recommendation engine, we used the following performance benchmarks: x **Root mean square error:**

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) (or sometimes root-mean-squared error) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values observed. The RMSD represents the sample standard deviation of the differences between predicted values and observed values. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation, and are called prediction errors when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSD is a measure of accuracy, to compare forecasting errors of different models for a particular data and not between datasets, as it is scale-dependent.

x Precision and recall:

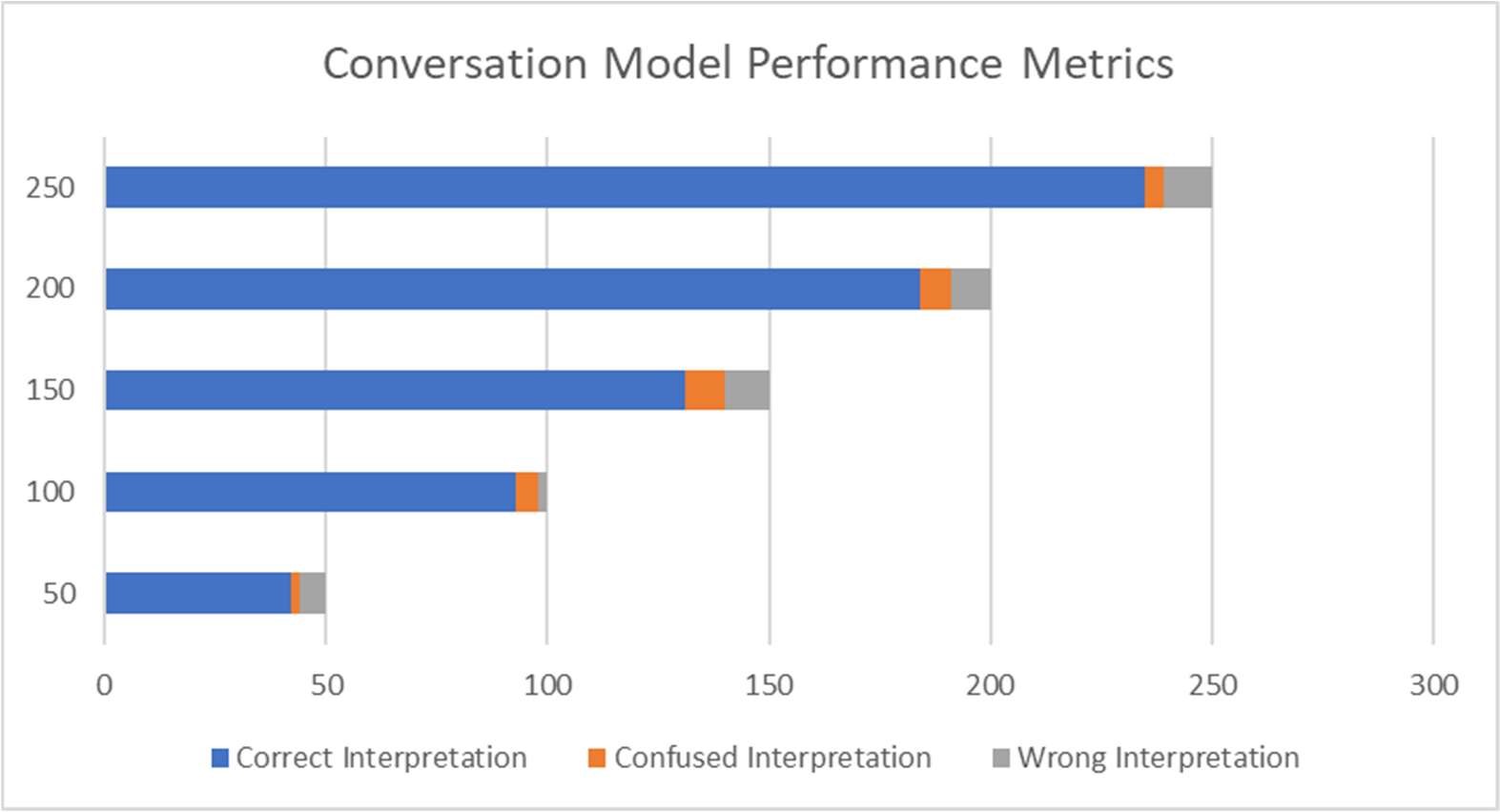
In pattern recognition, information retrieval and binary classification, precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance.

We have measured the performance of chatbot against some of these parameters. The performance results for our chatbot are as given below:

1. Conversation model performance metrics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. of messages | Correct  Interpretation |  | Confused  Interpretation |  | Wrong  Interpretation |
| 50 |  | 42 |  | 2 | 6 |
| 100 |  | 93 |  | 5 | 2 |
| 150 |  | 131 |  | 9 | 10 |
| 200 |  | 184 |  | 7 | 9 |
| 250 |  | 235 |  | 4 | 11 |

*Table 2 Conversational Model Testing Results*



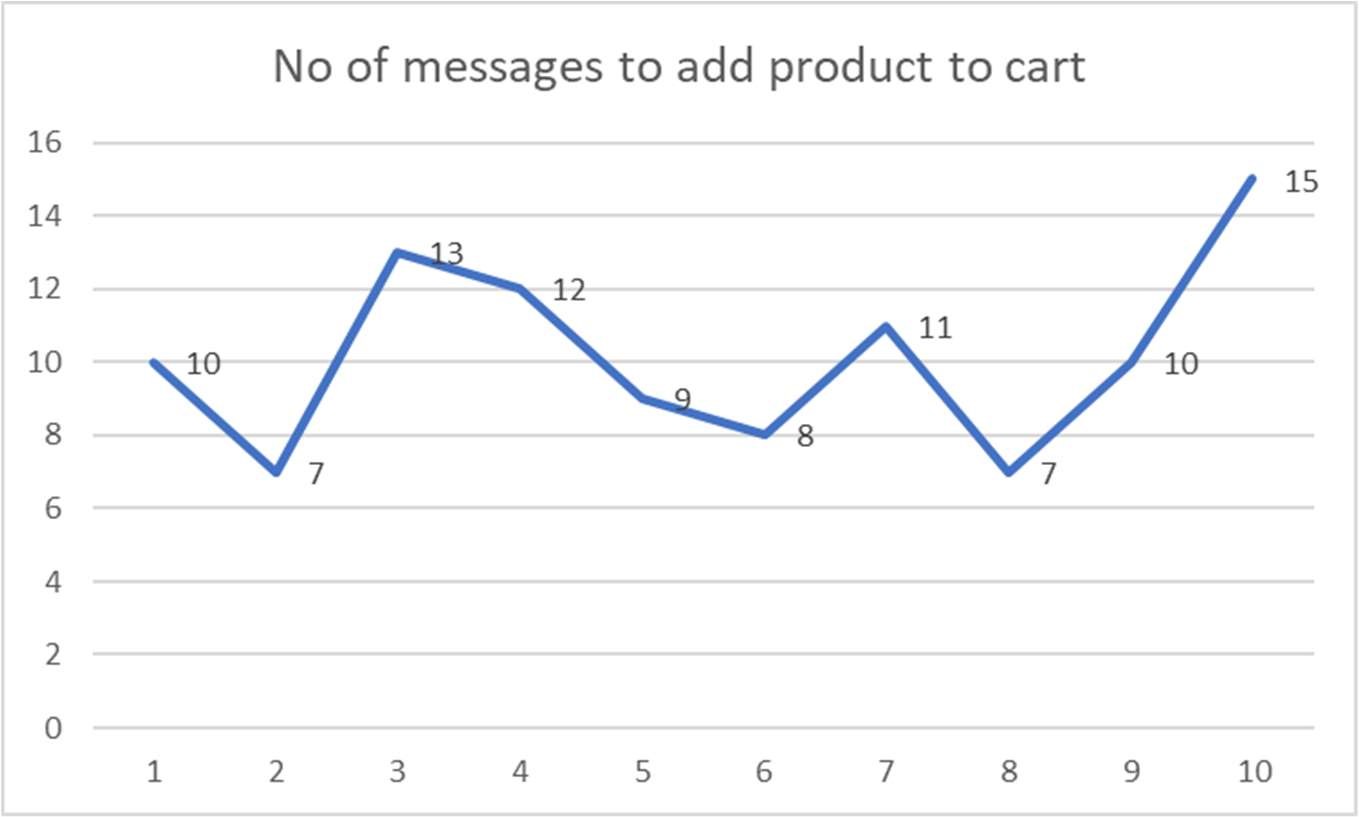
*Figure 11 Chart showing performance of conversation model*

1. Messages / Goal

Here, we have considered ‘Adding product to the cart’ as the goal here. We observed no of messages required across 10 conversations.

|  |  |  |  |
| --- | --- | --- | --- |
| conversation number |  | no of messages to place order |  |
|  | 1 |  | 10 |
|  | 2 |  | 7 |
|  | 3 |  | 13 |
|  | 4 |  | 12 |
|  | 5 |  | 9 |
|  | 6 |  | 8 |
|  | 7 |  | 11 |
|  | 8 |  | 7 |
|  | 9 |  | 10 |
|  | 10 |  | 15 |

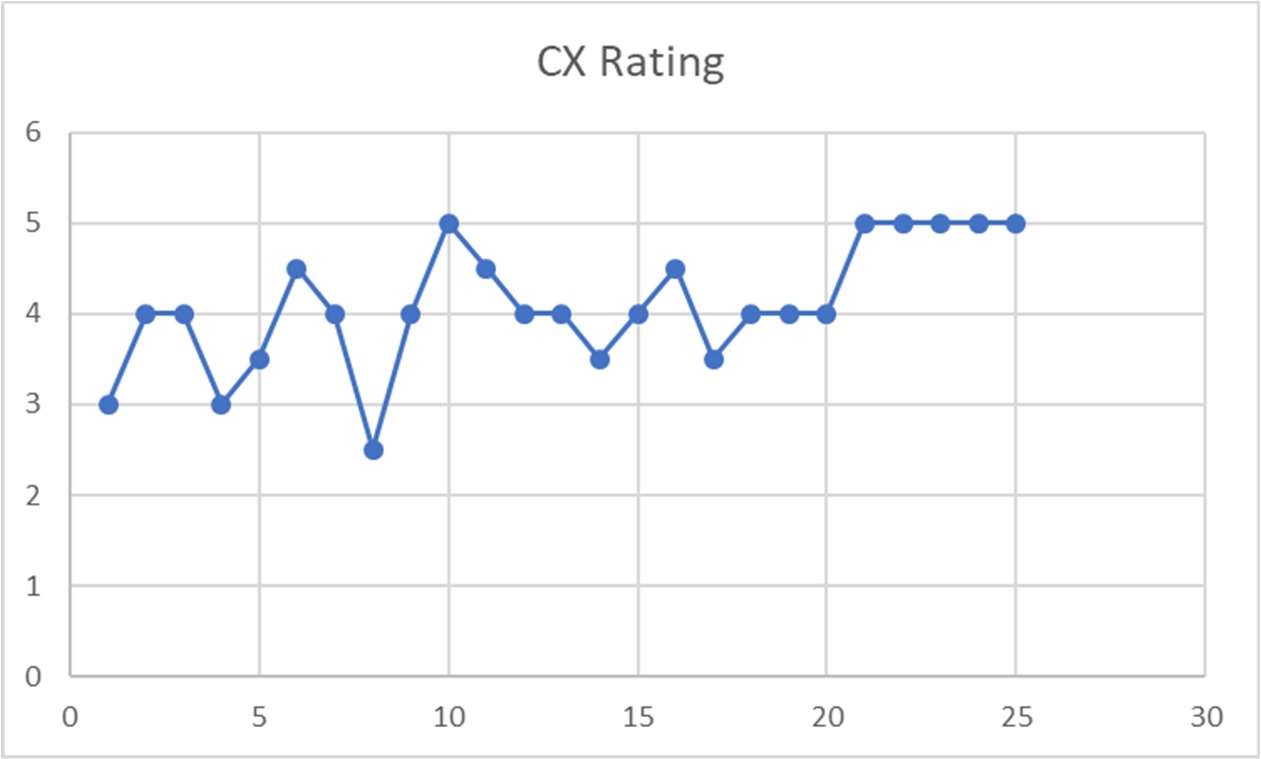
*Table 3 No of messages to reach goal test results*



*Figure 12 Chart showing average no of messages to reach goal*

1. Conversation Experience Ratings by User (1-5) –

We asked user to rate their experience with our chatbot on the scale of 1-5 and here are the results.



*Figure 13 Conversation Experience Rating by User for chatbot*

1. The average response time for our chatbot is 600-900 milliseconds.
2. Out conversation model has an accuracy of over 90% to generate accurate responses.
3. The confusion rate for our chatbot is less than 5% for the domain specific conversation.
4. Out chatbot supports 5+ communication channels for integration including skype, Cortona, MS Teams, Slack, Hangout etc.
5. Our chatbot is easily able to recognize 5+ intents including greeting, search product, add to cart, show more, place order etc. Additionally, it is very easy to add support for new intents, since it only requires training the LUIS model for the intent.

# Chapter 7. Deployment, Operations, Maintenance

**Deployment strategies:**

For production environment, we have the following strategy for deployment:

x Deploy the sentiment analysis engine and recommendation engine across 4 to 5 different instances by exposing it as a service in each of the instances. There would be a load balancer which receives requests, forwards to instances and delivers response. We have deployed our chatbot on Azure cloud through git code imports. x The recommendation system would have the sentiment analysis capability locally available to it, thus enhancing processing times. Both these engines are deployed separately on different AWS instances.

x The chat client would be integrated on various platforms, along with having an individual interface hosted on a different instance than the recommendation engine. For development and testing, we have these modules available on our own local systems, as well as on the cloud for verifying round trip times.

**Operational needs and maintenance:**

One of the core operational needs of our chatbot is the efficiency of our chatbot client in interacting with our users and providing them accurate information in minimal amount of time.

In order to achieve good efficiency and training our bot to reply properly, we have integrated the chatbot in our Slack channel and each of the team members talks to the chatbot and later reviews the conversations. Thus, in this way our bot is gaining conversational skills each day.

We plan to expose a public channel as well, so that anyone interested can have a word with our bot.

Apart from that, there are minimal maintenance requirements, which would mostly be in the form of enhancements.

# Chapter 8. Summary, Conclusions, and Recommendations

## Summary

As a part of project implementation, we have used Microsoft Bot builder framework implement a conversational chatbot which can be easily integrated within slack messenger platform. Currently, it supports basic conversation and collects product requirements from user through a series of questions. Few activities supported by our chatbot are search product, view product details, add product to cart, add product to Wishlist and order products. We have built a basic recommendation engine which works independently from the chatbot client to recommend the most appropriate products to the user based on the different criteria provided by him during the conversation.

We have built a basic running sentiment analysis engine which currently takes an array of user comments, analyses it and assigns scores from 1-5 indicating the connotation of the comment. Currently, we are working on improving the conversation engine to support more features, build a more robust product interfacing API and strengthening, securing the other modules. We have built user API controllers which can be easily plugged in with any ecommerce vendor to support user specific operations for that vendor.

## Conclusions

1. It is important to apply sentiment analysis not only to the user comments on the products recommended by the recommendation engine, but also to the conversation dynamically while the chatbot is interacting with the customer/user. This is important in improving the accuracy of the chatbot responses during the conversation.
2. The different modules in the system should be designed in a way to reduce the inter-module communication latency. This is important to avoid any delays and waiting periods during the conversation.
3. The accuracy of the conversation engine improves with the increase in the number of the conversations that it gets engaged in.

## Recommendations for Further Research

1. Add support for Conversation analytics metrics and parameters for the chatbot to understand and optimize the effectiveness of the conversation.
2. Study and optimize the accuracy of recommendation engine by adding more context-based parameters including user profile, user purchase history, browser cooking for web integrations etc.
3. Language translation support for chatbot conversation engine to enable multilingual communication.

# Glossary

x **Chat-Bot:** Is a program which interacts with the users x **Recommender Engine:** This program will select the products based on users buying pattern x **Sentiment Analysis:** Will understand what to recommend based on current customer needs.

x **Conversation Manager:** Will decide what to give as output based on customer

input.

x **Context Knowledge:** Is the pool of messages that will help chatbot understand what user input means.

x **API:** Application Programming Interface x **Features:** Are the list of attributes that entity has

# References

1. Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical*

*Informatics Association: JAMIA*, *18*(5), 544–551. http://doi.org/10.1136/amiajnl-

2011-000464

1. NLP pipeline, Figure 4, Chapter 6, Project Workbook 2
2. Matt Kiser (2016), Introduction to Natural Language Processing (NLP*),* article link: https://blog.algorithmia.com/introduction-natural-language-processing-nlp/
3. Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.
4. Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (pp. 160167). ACM.
5. Creating a Sentiment Analysis Model using google prediction API:

https://cloud.google.com/prediction/docs/sentiment\_analysis

1. Spam detection in comments:

https://cloud.google.com/prediction/docs/scenarios#spamdetection

1. Google Cloud Storage Documentation https://cloud.google.com/storage/docs/
2. Time-based sentiment analysis for product and service features(**US 20140219571**

**A1)** https://www.google.com/patents/US20140219571

1. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Mining of massive datasets. Cambridge university press.
2. Aggarwal, C. C. (2016). Recommender systems. Springer International Publishing.
3. Cahn, J. (2017). CHATBOT: Architecture, Design, & Development (Doctoral dissertation, University of Pennsylvania).
4. Green, M. (1985). Report on dialogue specification tools. In User interface management systems (pp. 9-20). Springer Berlin Heidelberg.
5. Hefley, W. E., & Murray, D. (1993, February). Intelligent user interfaces. In Proceedings of the 1st international conference on Intelligent user interfaces (pp. 3-10). ACM.
6. Horzyk, A., Magierski, S., & Miklaszewski, G. (2009). An Intelligent Internet

Shop-Assistant Recognizing a Customer Personality for Improving Man-Machine

Interactions. Recent Advances in intelligent information systems, 13-26.