AI-ENABLED SLACK TECHNICAL ASSISTANT FORMAL DOCUMENTATION

PROBLEM STATEMENT

This project is aimed at building a Slack chatbot that can provide technical assistance to Nutanix's representatives of the sales team and its 14,000+ customers. The chatbot will be developed using artificial intelligence methods like Natural Language Processing and Machine Learning and will be integrated with Slack to provide assistance over Nutanix's widely used slack channel. This AI-enabled chatbot will provide 24/7 instant human-like conversational capability and specific measures will be taken during the programming of the chatbot to make it intelligent enough to figure out when to answer by recognizing the general intent of the question. All scenarios which require human intervention will train the chatbot to answer similar questions in the future.

CURRENT STATE-OF-ART CONVERSATIONAL AI BOTS

Name of the paper: Recipe for building an open-domain chatbot
Company: Facebook
Link: https://arxiv.org/pdf/2004.13637.pdf
About: Facebook AI launched an open source Conversational-AI bot for open domain called Blender Bot in April 2020. This was trained on a text corpus of 9.4 billion language model and is able to acquire the following skills and combine all of them :

- Engaging Use of personality
- Engaging use of knowledge
- Display of emotion

Name of the paper: Towards a Human-like Open-Domain Chatbot
Company: Google

Link: https://arxiv.org/abs/2001.09977

About: Google launched an open domain conversational AI bot called Meena in February 2020, trained on a corpus of 2.6 billion parameters. It was built on a text corpus of social media interactions and has the ability to engage in conversations about a wide range of topics including humane features like emotions. It put forth a metric called Sensibleness and Specificity Average (SSA), and scored a value of 79 percent, which is very close to human interactions.

INITIAL THOUGHTS

Who, Why, What and How about a technical assistant?

1. Who needs technical assistance?
2. Why do they need technical assistance?
3. What technical assistant will do?
4. How will a technical assistant work?

Technical components

Knowledge Engine
Polling
Decision Maker
Slack Bot
Take feedback and update knowledge document if necessary, to improve over time

Version 1 functionalities and action items

Chatbot implementation in the Slack Channel which responds to technical questions when it is being addressed with a wake word

Version 2 functionalities and action items

Chatbot has been equipped with the intelligence to understand when to reply and when not to reply and emulates human - like responses and contains a feedback loop and learns from feedback

How to deal with missing information?

Questions that cannot be answered by the chatbot can be flagged as unanswered and the chatbot can learn those for future, based on responses from other users.

Feedback:

Users are asked if the response given by the chatbot met their objective. Users respond using a thumbs up or a thumbs down accordingly. In case of a thumbs down response, users are asked to elaborate more on which aspect of the response did not meet their expectations and their feedback is noted down accordingly. The feedback is associated with the category and the necessary update is made in the bot's knowledge base. The knowledge base is reviewed periodically based on the number of responses about a particular category and the number of times the bot is not able to answer a particular query for a user. Pseudo Relevance Feedback can also be implemented in order to not rely on manual feedback completely.

Several Key Performance indicators and metrics are collected from the bot periodically and its performance is measured and constantly maintained against certain thresholds.

Ranking:

Documents that are being retrieved must be ranked using a ranking function or algorithm and the highest ranked article / document must be displayed to the user. In order to assess the ranking, metrics like Precision, Recall and Mean Average Precision are used.

Precision is the proportion of retrieved documents that are relevant and recall is the proportion of relevant documents that are retrieved

Query Likelihood model can be implemented in order to measure likelihood of relevant documents getting retrieved in relation to the query.

CHATBOT KNOWLEDGE BASE

A knowledge base is a kind of repository with the information a chatbot requires to respond accurately. It helps the chatbot in rapid searching, retrieval, and reuse of information.

In our context, the knowledge base is Nutanix documentation which provides information regarding Nutanix architecture, how the software and features work, and how to leverage it for maximum performance.

The chatbot will be able to provide a textual response and a link to documentation using this knowledge base.

The knowledge base will help this AI-powered chatbot respond appropriately. The knowledge base is constantly expanding as it receives feedback from the users and updates it accordingly.

STEPS TO BE FOLLOWED IN AI CHATBOT DEVELOPMENT

The chatbot aims to reduce human intervention and improve response time for technical queries and issues faced by employees. The AI powered chatbot stores the conversations with users and the feedback it receives from users and expands its knowledge base to improve its responses over time. The development of the chatbot has the following stages:

1. **Role:** The chatbot interacts with users to provide technical assistance for several issues related to Licensing over the Nutanix Slack channel.
2. **Channel:** The chatbot resides in the Slack channel of the company.
3. **Data:** The chatbot is trained on data from the public documentation of Nutanix. This data is constantly augmented by users' feedback and stored in the database
4. **Chatbot engine:** This is the heart of the chatbot architecture which is powered by Natural Language Processing for understanding user

queries and providing appropriate responses. It comprises a Natural Language Understanding(NLU) unit, a processing unit and response generation unit.

5. **Integration system:** The chatbot consists of a front end integration component with the Slack channel of the company, which integrates the NLP component with the Slack channel and is developed using the Flask framework
6. **Training:** Train the chatbot with Nutanix documentation by altering the parameters to achieve good results. The AI-enabled chatbot requires a lot of training data to provide good accuracy. It should be trained to extract the intent and relevant information.
7. **Testing:** Test the chatbot for different scenarios. A lot of testing will be done to make sure the behavior of the chatbot is consistent.
8. **Analytics system:** A good analytics system will help measure each and every event or action of the chatbot. By understanding what the user is expecting, we would be able to optimize the chat in the best possible way to satisfy the users.
9. **User testing:** User testing will help in making us aware of the critical issues in the chatbot and optimize it. Multiple rounds of testing will help in improving the chatbot conversational quality and maximize user satisfaction.

CHATBOT NAME

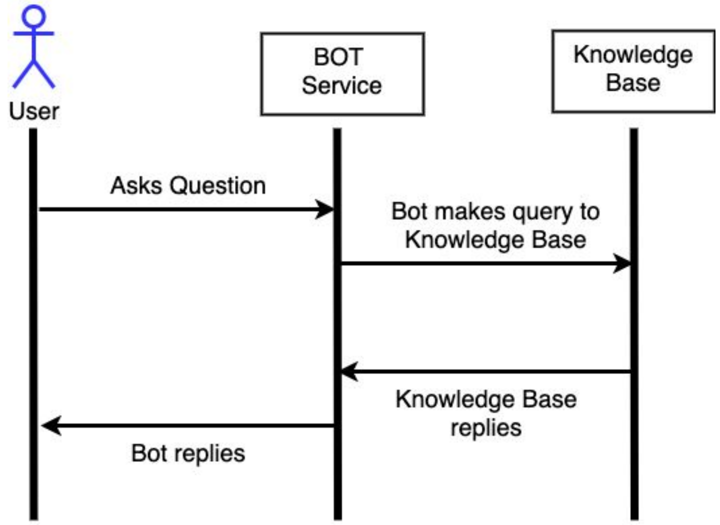
MissionX

DIAGRAMS

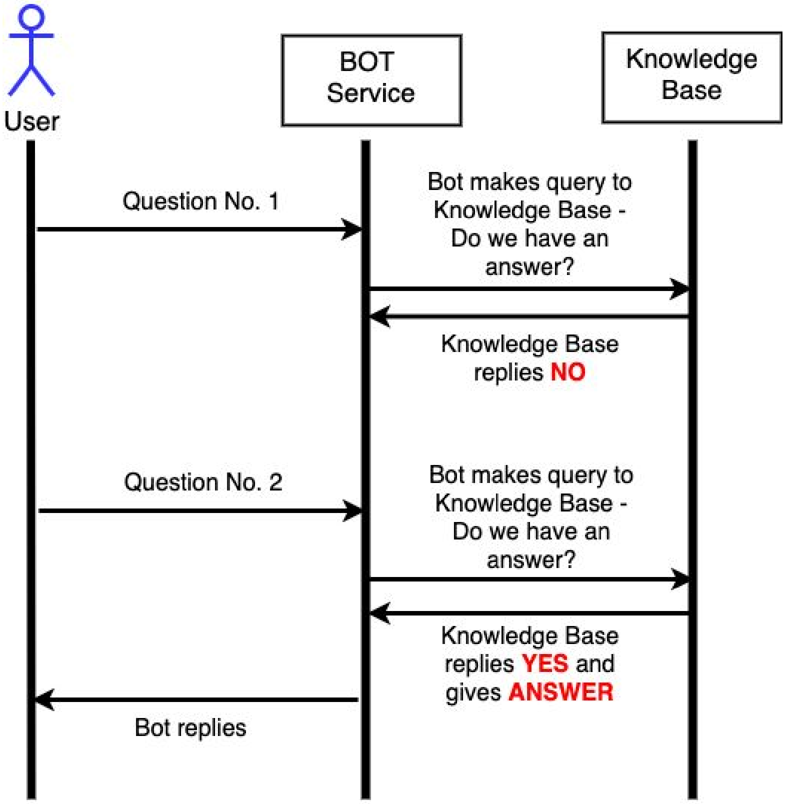
The below sequence diagrams illustrate the basic functionality of the chatbot.

Sequence Diagram

Phase 1

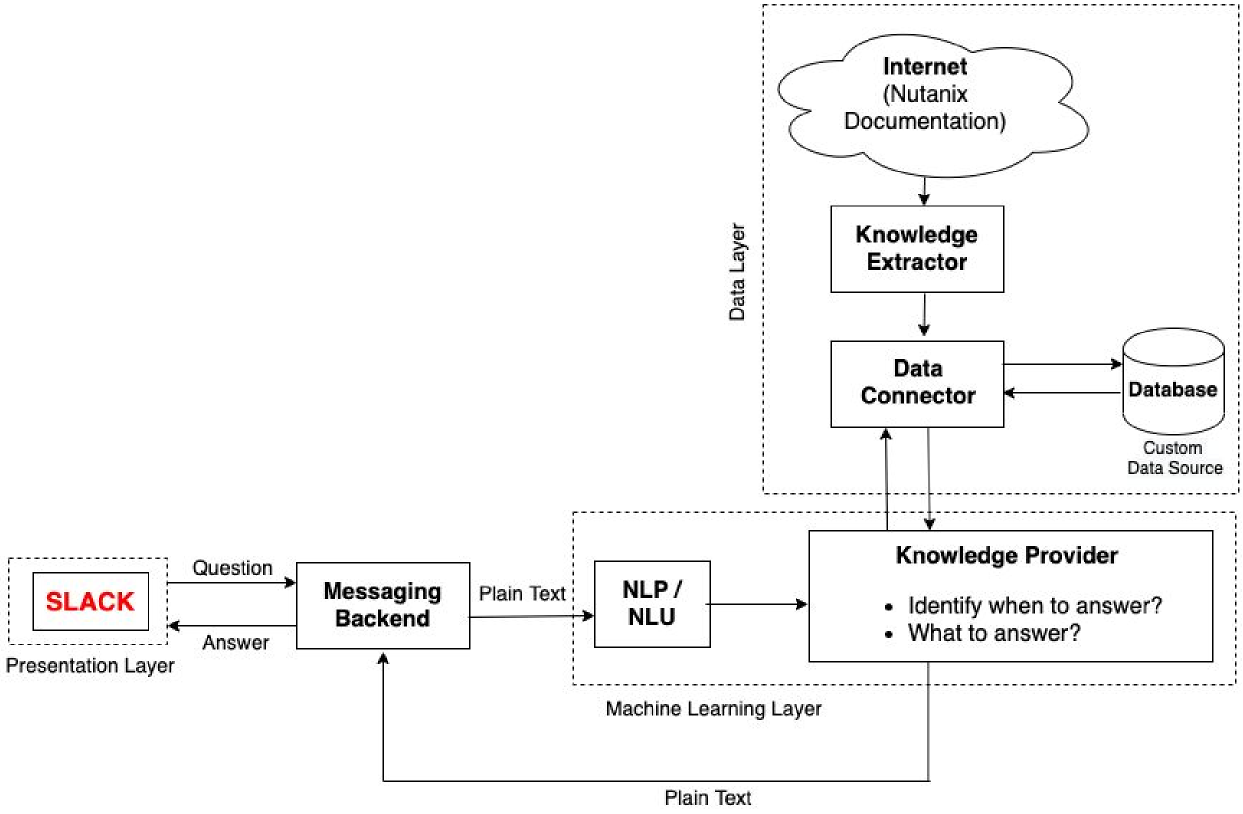


Phase 2


User
BOT Service
Knowledge Base
Question No. 1
Bot makes query to Knowledge Base - Do we have an answer?
Knowledge Base replies NO
Question No. 2
Bot makes query to Knowledge Base - Do we have an answer?
Knowledge Base replies YES and gives ANSWER
Bot replies

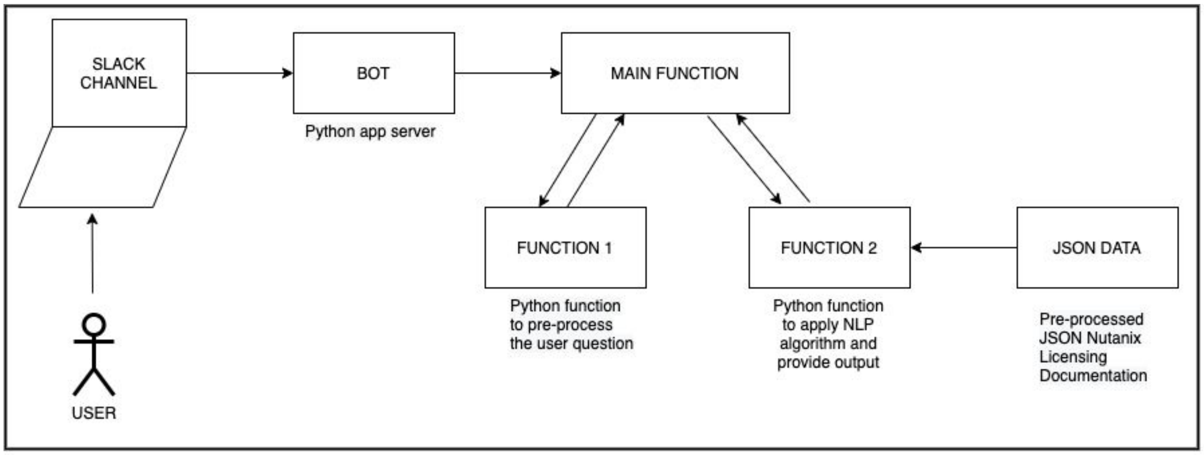
Architecture Diagram

The below architecture diagram illustrates the basic architecture of the chatbot.



Chatbot Architecture Diagram

Phase 1


SLACK CHANNEL
BOT
Python app server
MAIN FUNCTION
FUNCTION 1
Python function to pre-process the user question
FUNCTION 2
Python function to apply NLP algorithm and provide output
JSON DATA
Pre-processed JSON Nutanix Licensing Documentation
USER

STEPS:

Github Repository:

https://github.com/chetnakhanna16/MissionX

It has the complete model along with the slack integration code.

DATASET CREATION

1. **Scraping the Nutanix Documentation**
    - Storing HTML version of Nutanix Documentation to Local.
    - Converting the 'http' response to 'html' content.
    - Extracting the first 'h1' to get the name of the document.
    - Extracting all 'h1', 'h2' and 'h3' with 'a' tag.
    - Storing name of the document as 'Document_Name', href link of a tag as 'Link', div content next to h1, h2, h3 as 'Content' (the main content of the article), a tag text as 'Heading' (heading of the article) in a Python dictionary.

    Improvements:

    - The document needs to be stored in the database instead of our local machine.
2. A JSON file has been created which has this Python dictionary. This JSON file is our dataset for the chatbot.

PRE-PROCESSING

The headings from the documentation have been used to train the model. They are pre-processed using the following steps:

1. The headings are converted to lowercase.
2. ReexpTokenizer has been used to tokenize the words and all special characters have been removed.
3. English stop words have been removed.
4. Word stemming has been done using Porter Stemmer.
5. The keywords obtained from the pre-processing have been stored back in the JSON at their respective location.

6. The keywords are also stored in a list as ‘words’.
7. Similarly, all the headings have been stored in a separate list called ‘categories’.
8. A list named ‘words_categories’ has been created which is a tuple of ‘words’ and ‘categories’.
9. The lists ‘words’ and ‘categories’ have been sorted and only unique entries have been saved in them which have been stored as pickle objects.

Now ‘words’ contain the vocabulary for our chatbot and the ‘categories’ contain all the categories to classify.

TRAINING DATA

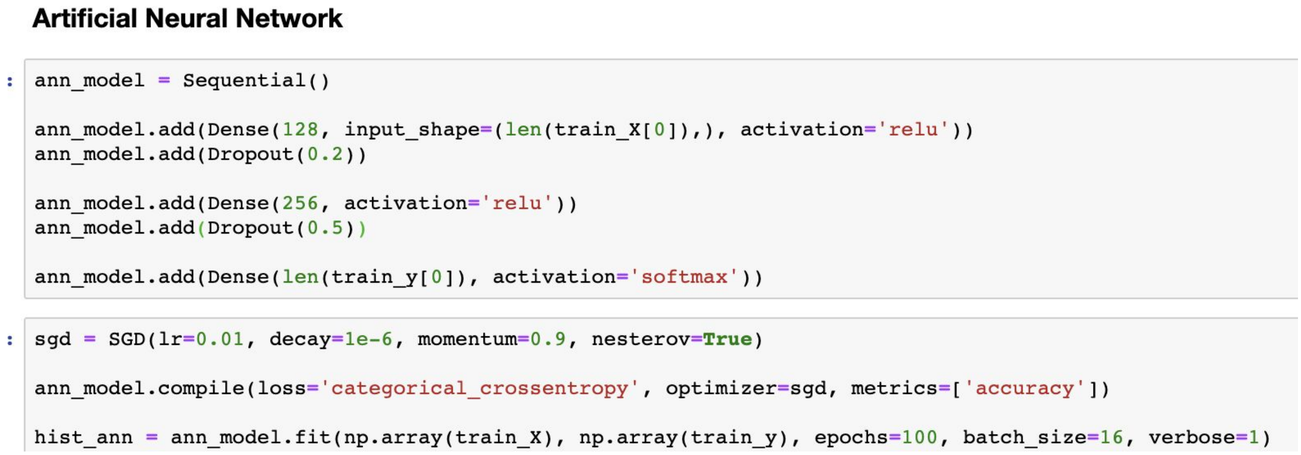
1. An array of zeros has been created which has the same length as the number of categories.
2. Each input pattern has been converted to numbers. Value 1 has been set for the indexes that contain the word in the patterns.
3. The output has been created in the same way by setting 1 to the category input the pattern belongs to.
4. Random shuffling has been done before using the training data.

BUILDING THE NETWORK

Artificial Neural Network

1. An artificial neural network model has been built with three dense layers.
2. The input layer has 128 neurons and ‘ReLU’ has been used as the activation function.
3. The hidden layer has 256 neurons and ‘ReLU’ has been used as the activation function.
4. The output layer is a multiclass classification layer with Softmax as the activation function to give probabilities.

5. Stochastic Gradient Descent with nesterov momentum has been used as the optimizer with learning rate 0.01, decay rate 1e-6, momentum 0.9.
6. Categorical cross entropy has been used as the loss function with Accuracy as the performance metrics.
7. The network has been trained for 100 epochs with a batch size of 16.
8. The model has 175,666 trainable parameters and gives an accuracy of 92%.



Artificial Neural Network

```
ann_model = Sequential()

ann_model.add(Dense(128, input_shape=(len(train_X[0]),), activation='relu'))
ann_model.add(Dropout(0.2))

ann_model.add(Dense(256, activation='relu'))
ann_model.add(Dropout(0.5))

ann_model.add(Dense(len(train_y[0]), activation='softmax'))
```

```
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)

ann_model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])

hist_ann = ann_model.fit(np.array(train_X), np.array(train_y), epochs=100, batch_size=16, verbose=1)
```

Long Short Term Memory (LSTM)

1. A basic LSTM model has been built which has one embedding layer, one LSTM layer and one dense layer.
2. The embedding layer is using the maximum possible and recommended vocabulary dim of 50000 words.
3. The LSTM layer has 256 units.
4. The output layer is a multiclass classification layer with Softmax as the activation function to give probabilities.
5. Categorical cross entropy has been used as the loss function with Accuracy as the performance metrics and Adam as the optimizer.
6. The model has 2,878,242 trainable parameters and gives an accuracy of 80%.
7. The network has been trained for 100 epochs with a batch size of 16.

PREDICTIONS

Predictions have been made by performing the following steps:

1. The question asked by the user is pre-processed first.
2. A bag of words array has been created which contains 1 for the words that exist in both question and our vocabulary ‘words’ and 0 for the ones that don’t.
3. The probability of the predictions made by the model obtained and sorted in descending order.
4. The link and content related to the highest probability is returned to the user as the response to the question.

The above two models have been used in the chatbot. Currently the bot can use any of the algorithms (ANN or LSTM) as per our specification.

Other Algorithms Tried:

BERT

A network was developed using the algorithm BERT by Google. The network is a part of the Hugging Face Transformer library which is made up of 12 layers or transformer blocks, 768 hidden layers, 12 Attention heads and a total of 110 Million parameters in its base form. This network is trained on lower case English letters. The network has been designed to understand context and English language in a way that makes it possible to perform various tasks like Question Answering, Sequence to Sequence classification etc. The network has been pretrained on the Question Answering dataset from Stanford called SQUAD (Stanford Question Answering Dataset). This pretrained model is fine tuned on the Nutanix Licensing dataset to perform question answering. The network is executed on the Google Colab platform with TPU support.

Challenges Faced:

1. The bot is able to provide answers to the user’s query but it is still not an intelligent bot which understands the context and language well.
2. Another LSTM model has been built to overcome the language and context issue by considering the fact that we just have 28 headings and their words in our vocabulary which makes a small dataset. However, the input form for this model’s training set has both heading and content. But, the test set is just a question string which is creating issues while making predictions.

SLACK INTEGRATION

The flask web framework has been used as the Python app server and the integration has been made with Slack.

CLOUD PLATFORM

To have reliable and scalable cloud computing service, we will be using Google Cloud Platform as our cloud partner.

TASKS

1. **Intent Classifier and Entity Extractor :** The engine must be able to parse the user query and perform tasks like stemming and lemmatization and identify and extract keywords from the user query
2. **Search the knowledge base :** The engine must be able to associate the query with knowledge articles from the knowledge base
3. **Ranking :** The engine must assign ranking to all the retrieved results based on a ranking algorithm
4. **Replying to the user :** The bot must reply to the user with the highest ranked page from the knowledge base. The reply must be a short summary of the appropriate answer, and a link for the knowledge base article

5. **Feedback :** The bot must collect feedback from the user and update its knowledge base with the relevant information which forms a continuous learning loop for the bot

DATASET

Nutanix Licensing

https://portal.nutanix.com/page/documents/details/?targetId=Licensing-Guide%3ALicensing-Guide

Additional datasets for future work:

Nutanix Bible

https://nutanixbible.com/

Nutanix Quick Reference Guide

https://www.adn.de/fileadmin/user_upload/Hersteller/Nutanix/Datenblaetter/Nutanix-Licensing-Quick-Reference-Guide-1_2.pdf

STAGES OF WORK

Stage 1 : Design document and final architecture of the chatbot with all its features

Stage 2 : Implementation of the Question-Answering chatbot

Stage 3 : Integration of the bot with the Slack channel and testing, further improvements and continuous learning through comments and feedback

DELIVERABLES

We aim to deliver the following:

1. A Question-Answering AI chatbot integrated with Nutanix Slack channel which is capable of answering technical questions in the channel and aims to learn and improve over time from users' feedback and comments
2. Demo of the chatbot for different types of users queries

3. **Project presentation about accomplishments and techniques used along with challenges faced**
4. **Scope of improvement and future work**

References:

1. https://nutanixbible.com/
2. https://portal.nutanix.com/page/documents/details/?targetId=Licensing-Guide%3ALicensing-Guide
3. https://www.quora.com/What-are-the-steps-in-an-AI-Chatbot-development
4. https://searchcustomerexperience.techtarget.com/definition/chatbot
5. https://chatbotslife.com/nlp-nlu-nlg-and-how-chatbots-work-dd7861dfc9df
6. Diagrams using https://app.diagrams.net/

TIMELINE

Task	Week Number	Date
Initial Documentation Feedback and Discussion	1	June 28, 2020
Final Documentation and Architecture Understanding and Learning Technicalities and Processes	2	July 5, 2020
Code Feedback and Discussion	3	July 12, 2020
Code Feedback and Discussion	4	July 19, 2020
Code Feedback and Discussion	5	July 26, 2020
Code Feedback and Discussion	6	August 2, 2020
Integration with Slack Deployment Feedback and Discussion	7	August 9, 2020
Demo and Alpha Testing Working on critical issues for improvement Feedback and Discussion	8	August 16, 2020
Beta Testing Working on critical issues for improvement Feedback and Discussion	9	August 22, 2020
Final product Demo Presentation	10	August 30, 2020

Resources:

https://www.analyticsvidhya.com/blog/2019/10/how-to-build-knowledge-graph-text-using-spacy/

https://www.youtube.com/watch?v=mmQl6VGvX-c

https://www.youtube.com/results?search_query=Knowledge+graph

https://searchcustomerexperience.techtarget.com/definition/chatbot

https://chatbotslife.com/how-i-developed-my-own-learning-chatbot-in-python-from-scratch-and-deployed-it-on-facebook-88bc828be0a8

http://ai.intelligentonlinetools.com/ml/list-links-chatbot-examples-tutorials-code-platforms/

https://medium.com/coinmonks/how-to-make-a-q-a-chatbot-with-machine-learning-1c90207bde7b

https://dzone.com/articles/python-chatbot-project-build-your-first-python-pro

https://towardsdatascience.com/natural-language-understanding-with-sequence-to-sequence-models-e87d41ad258b

https://towardsdatascience.com/bert-for-dummies-step-by-step-tutorial-fb90890ffe03

https://arxiv.org/pdf/1809.08267.pdf

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/?utm_source=KJLinkedin&utm_medium=post&utm_campaign=21_July_new_article

https://www.em360tech.com/ai_enterprise/tech-features-featuredtech-news/top-10-customer-service-ai-companies/

https://www.intercom.com/blog/customer-service-chatbots/

https://towardsdatascience.com/how-i-built-a-deep-learning-powered-emoji-slackbot-5d3e59b76d0