

“Emotionally Intelligent Chatbots”

***By***

**JAY JOSHI (200440993) KAJAL SHAH(200440004)** [**jjm376@uregina.ca**](mailto:jjm376@uregina.ca)[**kms613@uregina.ca**](mailto:kms613@uregina.ca)

***Of***

**Department of Computer Science University of Regina**

**Instructor: Dr. Alireza Manashty**

**Table of Contents**

1. [Introduction 1](#_TOC_250011)
2. [Team Members and Roles 2](#_TOC_250007)
3. [Problem Statement 3](#_TOC_250003)
4. [Project Context 3](#_TOC_250002)
5. [Solution Overview 3](#_TOC_250001)
6. Data Set 4
7. [Tools 4](#_TOC_250004)
8. [Expected Outcomes 5](#_TOC_250000)
9. Expected Outcome 6
10. [Data Science Life cycle phases 6](#_TOC_250005)

10.1 Data Acquisition and Preprocessing 7

10.2 Feature Engineering ,Model Planning and Evaluation 3

* 1. Deployment and Communicating Results………………………………………….

11 [Conclusion and Future work 4](#_TOC_250006)

12 References 8

**1 Introduction**

A chatbot is an intelligent piece of software able to interact and execute identical acts to a human being.

Chatbots are used widely in consumer service, ads on social network platforms and direct online messaging [1]. In business, the word "chatbot" is also used for conversational agents which can be implemented with any online messaging program. These conversational agents are designed to communicate in human language. Chatbots are very useful for the businesses, several customers prefer to inquire through chatbots in spite of using the calling center services. It also handles customer support as a result it reduces the waiting time and costs of the customers and improves the availability [2].

So, basically Chatbots are like a virtual assistant that creates a better customer experience across every touchpoint of the consumer.

Chatbots are generally of three types:

* **Rule based or Retrieval based chatbots:** These types of chatbots are based on predefined input and patterns. Such types are generally used for business purpose where rules are set and chatbot operate in a constrained manner [3]. It is built with the help of Chat Marketing Platform where the Chatbot will guide as per the response selected by the user. E.g. ELIZA and PARRY systems [2].
* **Self-Learned or Corpus Based Systems:** For such version, Deep Neural Networks are used to search a huge dataset to get a relevant answer. It requires large amount of data as input is transformed into an output through sequence to sequence neural networks. E.g. Microsoft Xiaolce and Microsoft Tay [2].
* **A.I. Chatbots:** In this type, user inputs text, Bot analyzes the request, identifies intent and entities and then replies to user.
* **Live Chat:** In live chat, the support team sees the question and replies to the customer’s question in real time.

The Chatbots are designed based on Natural Language Processing Concepts which consist of Natural Language Understanding (Machines' ability to understand the human language, such as English.) and Natural Language Generation (A machine 's capacity to produce text identical to written texts created by humans [1]

The idea of understanding and understanding emotions is emotional intelligence, and being able to label them as a specific feeling — for example, excitement. EI with AI allows sophisticated systems (such as chatbots) capable of sensing and reacting to emotions from a conversation [5]. You may think of it as an emoji generator, where EI maps a certain emoji for any thinking or feeling [4]. Chatbot in healthcare has been developed and certain advancements are required in that. This type of chatbots support pain relief for the users. Users should pose questions related to health and include their signs to know about illness and future remedies [2]. Additionally, for example, a virtual personal assistant can help support recovery and self-care by reminding the user that it is time to stretch if they were working on their computer, or with a reminder of medication.

In the coronavirus pandemic, when we face a shortage of health care provides, chatbots could ease the medical professionals' jobs. Chatbots can help in drug administration, in emergency cases and in-patient triage. Also, they could propose treatments for simple medical problems. [2]

**2 Role of Team Members**:

**KAJAL SHAH: JAY JOSHI:**

Defining Problem and Analyzing Concepts Defining Problem and Analyzing Concepts

Preparing the Overview and Report Preparing the Overview and Report

PowerPoint Presentation and Diagrams PowerPoint Presentation and Diagrams

Importing and Loading the Data Importing and Loading the Data

Preprocessing the Data Preprocessing the Data

Create Testing and Training Data Create Testing and Training Data

Training the Model Training the Model

Interacting with the Chatbot Interacting with the Chatbot

Each member is capable of performing each and every role, so there is no specific role defined for a particular member. It will be a combine effort of both the members in completing all the tasks.

**3 Problem Statement**:

Under the General Data Protection Regulation (GDPR), the usage of personal data must be limited, including all details that applies to a known or recognizable entity. This recommends that all sensitive data will be kept in encrypted form and isolated from all data. In fact, users will be able to neglect all their personal details during a chatbot talk.[3] The anonymity of the users will be embedded in chatbots architecture and operation. Ask the user 's permission, for example, and explain what the chatbot is doing, like accessing the user's calendar. So, privacy is an issue with certain chatbots.

Whether you've ever used a chatbot, you may have come away from the encounter with the impression that the system seems a little disconnected. Siri, Cortana, and Alexa are the most commonly used and developed Chatbots nowadays, but it doesn’t provide any emotional support or help to the users. Maybe it helped you understand the procedure, but it became obvious that the bot couldn't comprehend your discomfort or pick up on your annoyance given their friendly answers. Several people want a digital companion with whom they can share their feelings, ask about their personal problems, celebrate their victory and many more things [6].

Researchers are still trying to develop chatbot that could recognize emotions of a person, that could understand the human emotion and give reply accordingly. Along with that, in this COVID-19 pandemic, as the number of COVID -19 patients are increasing people are panicking and need assistance regarding that. This type of chatbot will use neural networks to perform an active, one-to-one interaction with its owner and learn how to communicate like them over time.

**4 Project Context**:

This project will develop an emotionally intelligent chatbot, they will detect the emotion and the tone. They will be able to respond appropriately to human emotions, even though they can’t replicate the proper processes of those emotions. It could relieve the stress and loneliness of the person. If a person is older or housebound due to illness, such a chatbot could give help. Such a situation may lead even emotionally healthy people to become agonized by a loss of interaction with others and chatbot can provide immediate assistance. Apart from that, it could also answer questions related to the COVID-19.

**4.1 Project Overview**:

This section defines the objectives, project scope, and target deliverables.

**Objectives:**

* The main aim of this project is to develop a chatbot that is emotionally intelligent, who is like a virtual friend to the user.
* This type of chatbot learn more and more about the person’s nature, interest after each and every conversation and this is how it make maximum authentic conversation.
* Improving user experience by providing assisting them in answering the questions related COVID-19, its causes, precautions and many more.
* Deliverables:
* Implementation of machine learning and Artificial Intelligence techniques/algorithms to develop an emotionally intelligent chatbot.
* Designing a Chatbot that gives knowledge about Covid-19 pandemic and answers some general questions asked by the users like its precautions, safety measures, causes and symptoms.

**5 Solution Overview**.

**5.1 System Architecture and Components**

The below fig 5.1 shows a brief overview of session of a chat bot including all of its component. Microsoft Azure provides services like Azure web application, Azure bot service and Azure cognitive services. Cognitive service consist of Q/A maker, Emotion Analytics etc. Chat bot is created using Azure bot service. Web applications are used to host the chat bot services. System uses database to store conversation with the users. Cosmos DB data base is used to store bot conversations with the user.

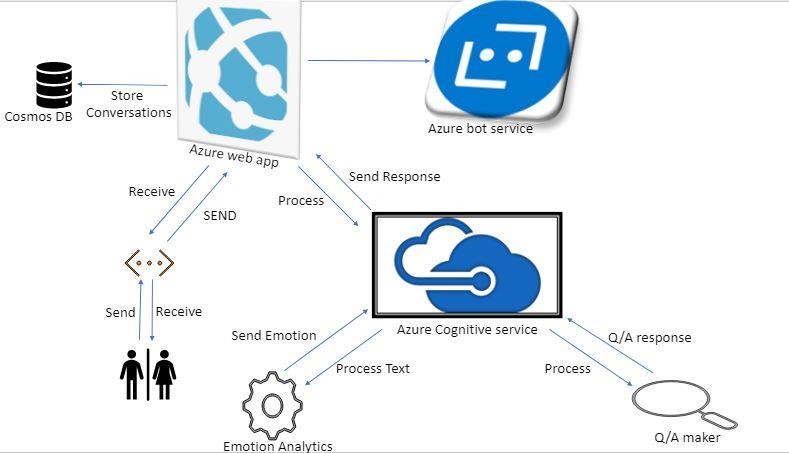


Fig 1 System Architecture and Components

**5.2 Working of a chat bot.**

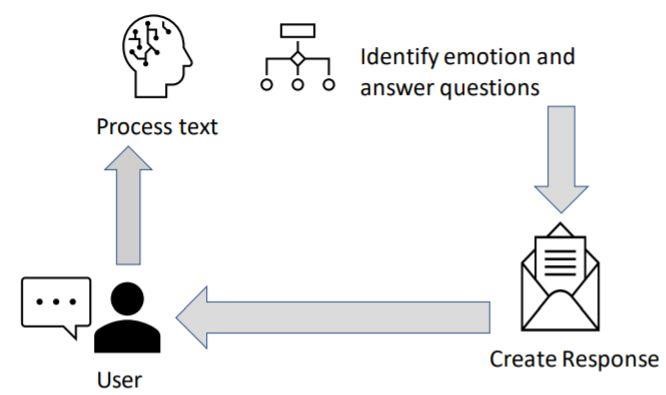


Fig 2 Solution Overview

The above fig 5.2 shows the basic working of a chat bot. Whenever a user interacts with the system, the bot system will first analyze the request, process the text and based on user responses it is intelligent to identify the emotions of the user and react accordingly. The bot system is also able to answer the user queries related to novel corona virus-19 pandemic. The bot then creates response and sends it back to the user.

**6 Data Set**

Since there is not much more research been carried out for emotion detection with conversational chat bots the idea is still novel, there are not many large efficient data sets available for training our model. Still we were able to identify few data sets which we are going to use to train our model. The source and format of data set is as listed below.

* The "Emotion Intensity in Tweets" data set from the WASSA 2017 shared task. [http://saifmohammad.com/WebPages/EmotionIntensity-SharedTask.html.](http://saifmohammad.com/WebPages/EmotionIntensity-SharedTask.html) This above data set has more than 10k sentences labeled in textual format as “anger”, “joy” etc. [6].
* ISEAR dataset consists of 7,666 sentences annotated by 1,096 participants with different cultural background who completed questionnaires about experiences and reactions for seven emotions: anger, disgust, fear, sadness, shame and guilt. The ISEAR dataset contains the emotional statements that in turn contain the emotional sentences. The data set is available in csv format [7].

Also we are planning of web scraping twitter web site to collect live tweets and tag each of tweet into different emotion for increasing efficiency of our model.

**7. Tools**

As discussed earlier we are going to use Microsoft azure frame work to create an intelligent chat bot. Some of its sub components are listed below.

* Microsoft Azure Web application
* Microsoft Azure Bot Service
* Microsoft Azure Cognitive Service
* Mongo DB
* Text Analytics
* Natural Language Understanding(LUIS)
* Q/A maker
* Beautiful Soup ( for web scrape if applicable)

**8 Project Planning and Timeline**.

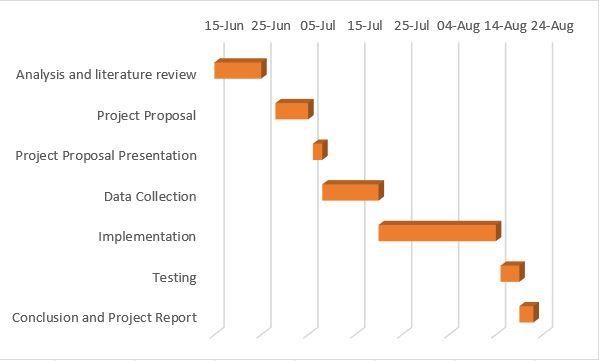


Fig 3 Project Planning and timeline through Gantt chart

Communication plan: In order to complete the project efficiently and in a given timeline, it is necessary to communicate in a team as well as with the instructor, if we are facing any roadblocks in our project development process will connect with our instructor Dr. Alireza Manashty by taking a prior appointment with him.

**9.** **Expected Outcomes**.

Our main of the system is to design an emotionally intelligent chat bot that can interact with the user, identifies the emotion of the user and react to the user accordingly. Additionally, our system also aims to answer the Frequently Asked Questions (FAQ) about the novel corona virus-19 pandemic whenever prompted. The efficiency of a chat bot to identify user emotion depends upon its training phase. Initially we are planning to train our chat bot that can identify human emotions of type including “anger”,

“happiness”, and “joy”, “sad” and “fear”. As time progresses and with more evaluation and efficient data our system will be capable to identify additional human emotions which can be entailed as future work

**10. Data Science life cycle phases:**

**10.1 Data Acquisition and Preprocessing**

We have used a dataset from ISEAR for our project as well a dataset from,

ISEAR dataset consists of 7,666 sentences annotated by 1,096 participants with different cultural background who completed questionnaires about experiences and reactions for seven emotions: anger, disgust, fear, sadness, shame and guilt. The ISEAR dataset contains the emotional statements that in turn contain the emotional sentences. The data set is available in csv format.

Using the dataset emotion dialog data set , we have trained our model in ML.net classification algorithm which classifies human emotions into several different categories.

We have used above two data set for creating two ML models respectively. Currently we have developed our model trained in ML.net classification algorithm on data set indexed 2 and deployed on azure cloud using Azure Web app service. We were successful in studying results of using LSTM recurrent neural network model on ISEAR data set to classify labels into seven different categories. So, let’s discuss in detail about different phases of two different models and their results.

**10.1.1 Data Cleaning and Visualization**:

The next task is to visualize, understand and preprocess the data.

**Data Visualization**:

It is the process of taking raw data transforming it into graphs, charts and images. That allow us to gain insights from it. It help us to discover new patterns and trends in our data. Some images that show how we have done data visualization using are shown as below. This image shows the histogram the Emotion\_value like fear, anger, sadness, shame and guilt.

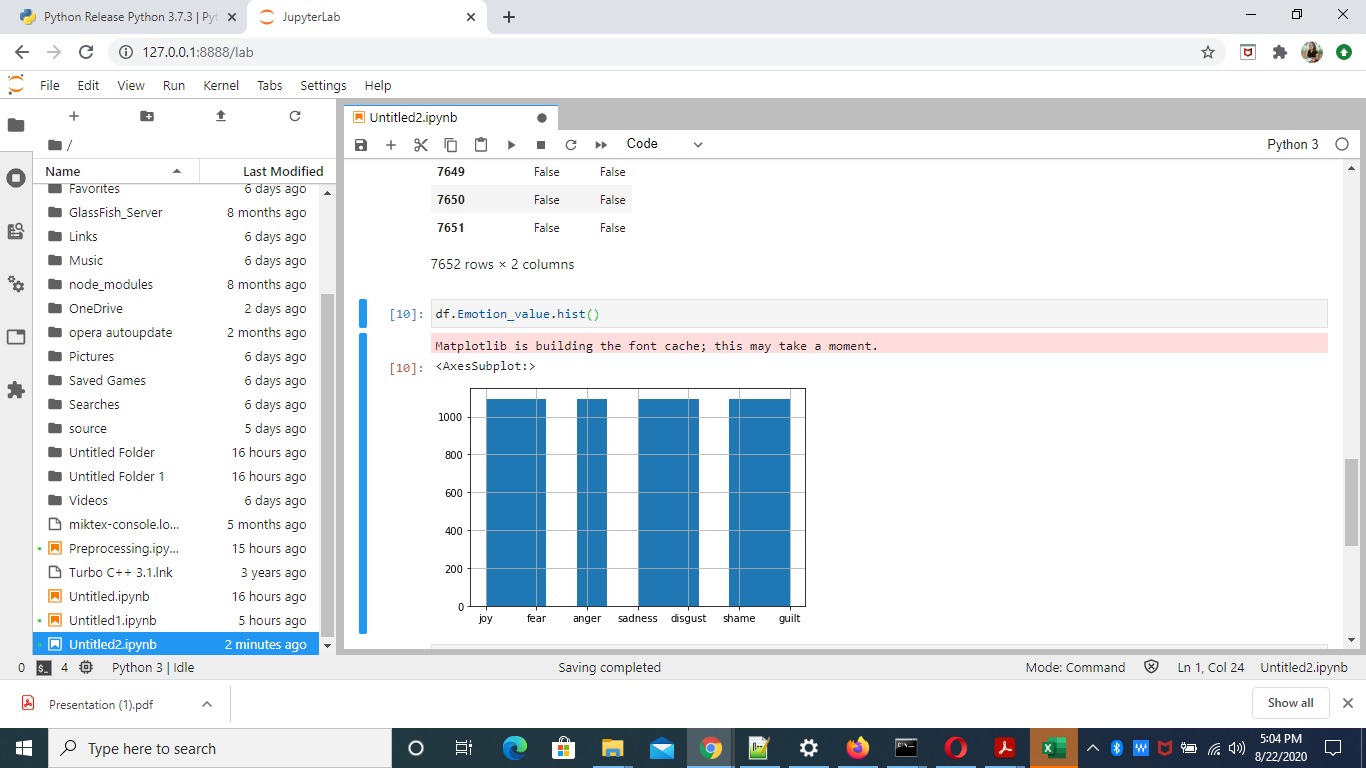


Image 1 “Data Visualization using Pandas”

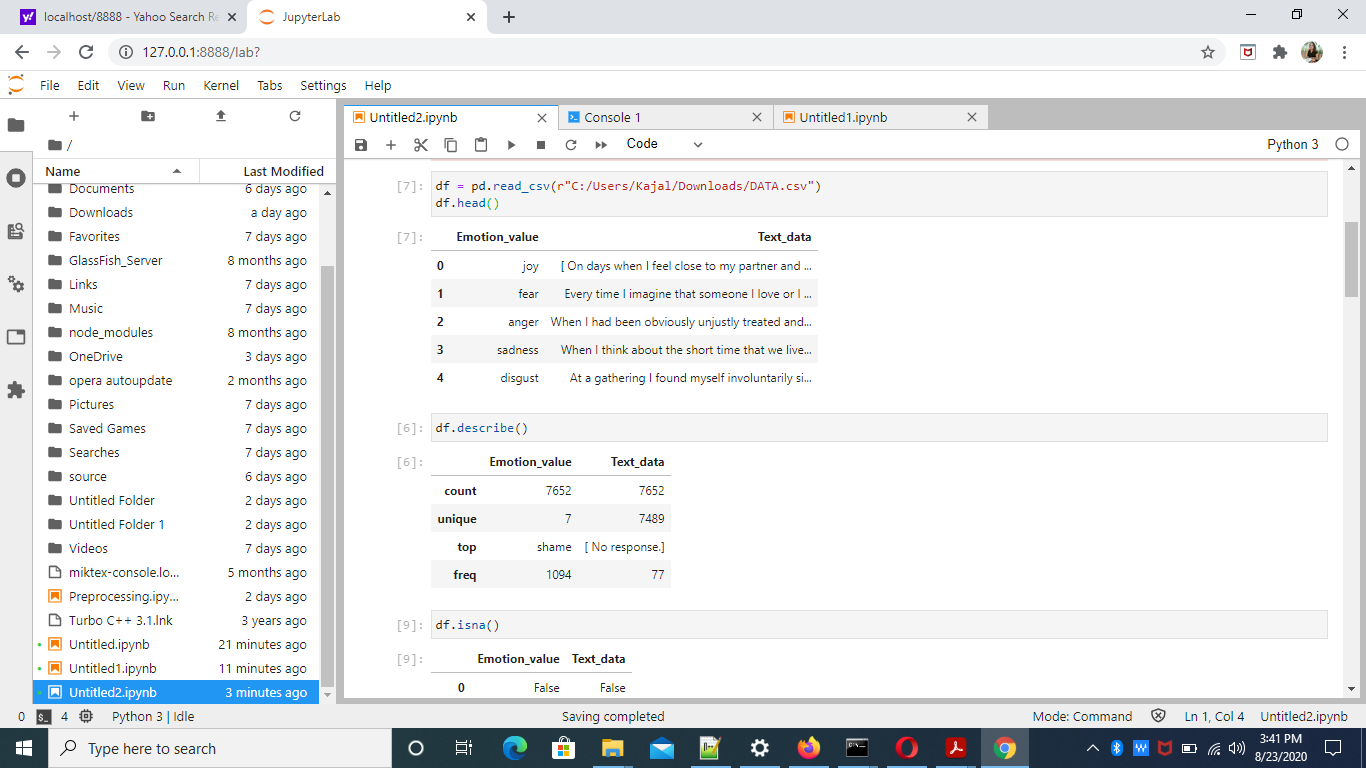


Image 2 Data Visualization using pandas

**Data Cleaning**

We have use open source Open-Refine data cleaning tool for cleaning our data. It is a very important step as if the data is not filtered, the results of the analysis will not mean anything.

A screenshot of that is attached below:

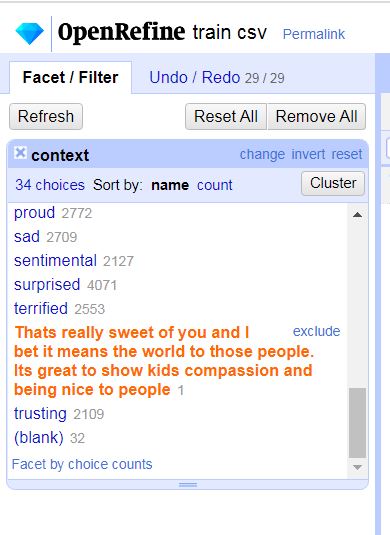


Image 3 Data Cleaning using open-refine

We have analyzed the text with Natural Language ToolkKit. In the process of learning how to do sentiment and emotion analysis, Natural Language Processing is quite important.

**10.1.2 Data Preprocessing**:

A lot of time goes into data cleaning and preprocessing phase when you are dealing with data that contains natural language. Since, input to our LSTM model is are English language sentences we need to apply some preprocessing steps before we actually consider it as input to our model. So we have autocorrect all the spellings using python autocorrect library. We have created tokens of our word in each sentences, removed any non-alphabetic characters and punctuations, if they were present in data.

Results of applying text preprocessing.

* Firstly, our corpus size i.e. total number of words in our data get reduced to 8656 instead of 9105 words present before text preprocessing.
* We have tokens of each word in input sentences, and removed any non-alphabetic characters and punctuations if they were present.
* Also, initially we have represent each output classes with sequence of integer numbers.

Below images represents a snippet of code and result of applying text preprocessing to our data.

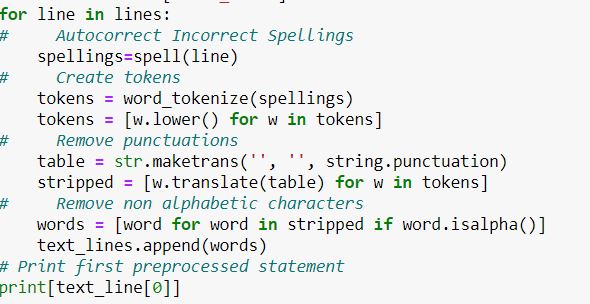


Image 4 Text Preprocessing code snippet

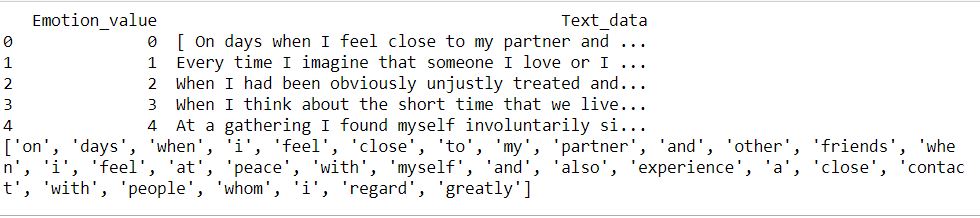


Image 5 Text Preprocessing output

**10.2 Feature Engineering, Model Planning and Model evaluation:**

**10.2.1 Using LSTM layer**.

Since LSTM model of Recurrent neural network works best for input data which are in sequential form, we decided to train our data using LSTM model. Before actually initializing with training of our model we need to follow some of the steps which are essential to perform before model training.

The human brain is not persistent enough to start thinking from scratch whenever required. Traditional neural networks are unable to perform this task, which is one of major limitations. Recurrent neural network are capable to address this issue, they work in loops which allows them to persist the previous information. This loops make recurrent neural network seemed to be altogether different thing, but they are similar to traditional neural network. As shown in figure below, recurrent neural network can be considered as multiple instances of same network passing message to successor. This chain like behavior natural reveals that a recurrent neural network is related to chain like sequence of inputs and data.

Usually we only need to look at recent information to do present task. For example, suppose RNN is designed to predict next word in sentences, on the basis of previous inputs like say, we need to predict last word in the sentence “The stars are in the sky”, we don’t need to look at previous information its obvious that last word will be sky. Thus in such examples were gaps between previous information are less RNN’s are best to learn past information. But as the gap between previous and present information starts increasing RNN’s become unable to connect the information. Hence, for handling long term dependencies LSTM’s are used.

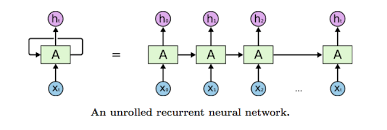


Image 6

The maximum input size of sentence for our lstm is set to be 50, hence it accepts input sentences having total maximum number of words equal to 50. Before we move ahead, we have to represent each labeled sentence in our data as (1,50) matrix were each cell indicates, the index of that number in word index, extra padding of zeros is done in matrix for sentences with size less than 50. Each output label is converted into one hot binary vector for effective representation. The below code image shows snippet of performing all above mentioned task.

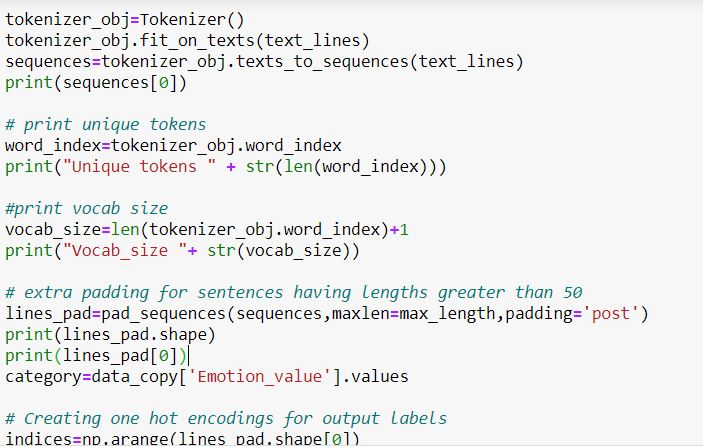


Image 7 Initialising and one hot encoding.

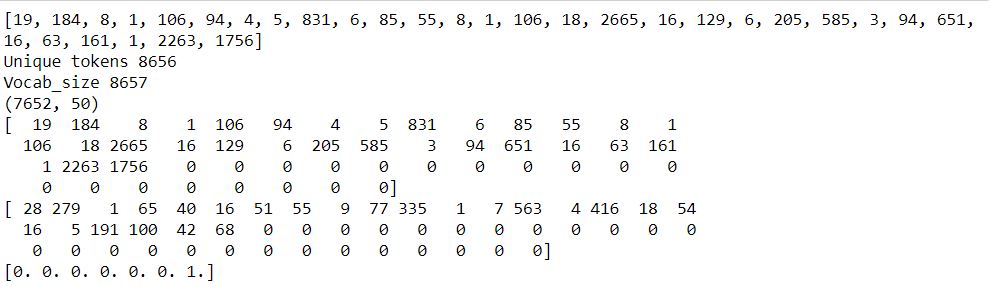


Image 8 Initialising and one hot encoding output.

We are using glove vectors which is a large pre trained set of word vectors to represent each of word in our corpus as a word vector of 300 dimensions. The concept of glove vectors is, “Judge the word by the company it keeps”, Glove basically is unsupervised algorithm for creating vector representation of words. Training is performed on global co-word occurrence statistics from large corpus which represents interesting sub structures in word vector space.

The cosine similarity between two word vectors provides effective method for measuring linguistic and semantic similarity between two words. This similarity metric used for nearest neighbor evaluations gives non scalar number which quantifies the relatedness of the words. This is mainly helpful because they are so many pairs of words which represent same level of information like man and woman, girl and boy etc. For more information you can visit <https://nlp.stanford.edu/projects/glove/> this website.

Since glove vectors is effective way to capture semantic and linguistic information between two words, we are using pre-trained glove word vectors as an embedding layer to our LSTM model. Below images show to snipped of how to create a embedding matrix from pre-trained glove vectors.



Image 9 Loading glove vectors

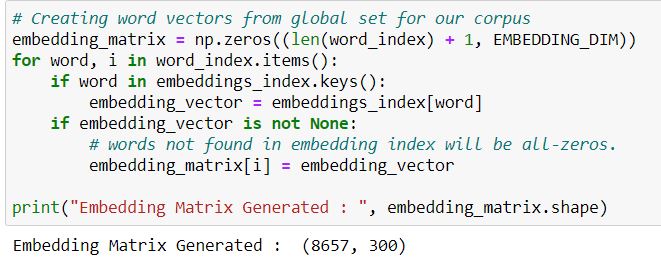


Image 10 Creating Embedding matrix.

Since the size of our word index is of 8657 in our corpus the dimension of embedding matrix is (8657,300).

Next we split our data set into training and testing batches. We have used different percent of validation data in each training of our model to evaluate the results. The below code snippet shows splitting of trainng and testing data set.

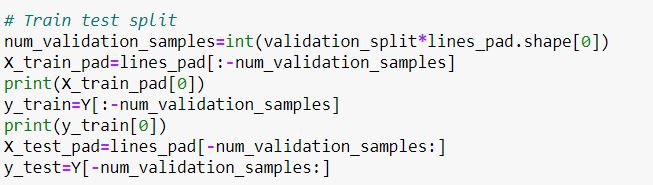


Image 11 Train test split

In above code snippet, validation split varies in each training period for better test efficiency of the model.

Let’s define input and output for our LSTM layer. First we generate an embedding layer for our LSTM model. The below code snippet shows defining an embedding layer for our LSTM model.

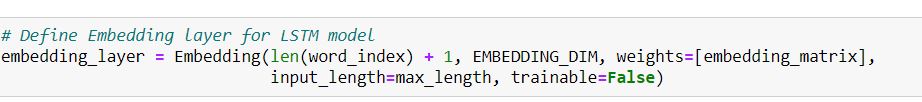


Image 12 Define Embedding layer

Since global word vectors are of 300 dimensions embedding\_dim is 300 and word index is around 8600. Hence we have defined an embedding layer which represents each word with vector dimension equal to 300 for our lstm model. Let’s declare our model. The below code shows snippet of declaring our lstm model.

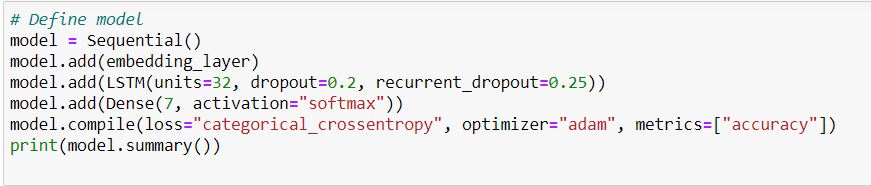


Image 13 Declare model

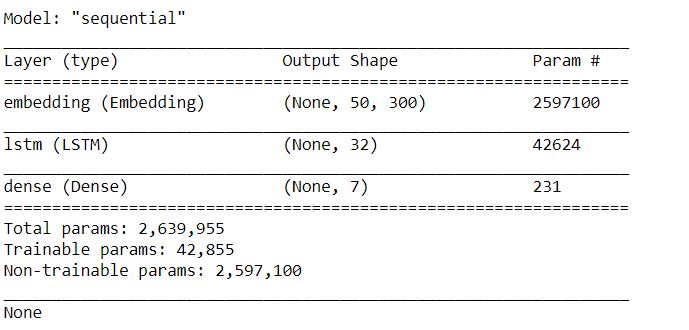


Image 14 Model summary.

As it is shown above, embedding layer derived from global word vectors is added to our lstm model of 32 cells units with drop out rate equal to 20% to avoid overfitting of our model, since RNN’s are naturally prone to overfitting because of their complexity. The 7 dense layer added are for seven distinct categories of emotion, each layer for one output emotion with activation function “softmax”, since our problem is of multiple classification, we are using softmax activation function.

Since we are done with training of our model lets evaluate testing results of our lstm model. We have trained our model for 15 epochs on different percentage of validation split equal to 10%, 5%, and 20% respectively.

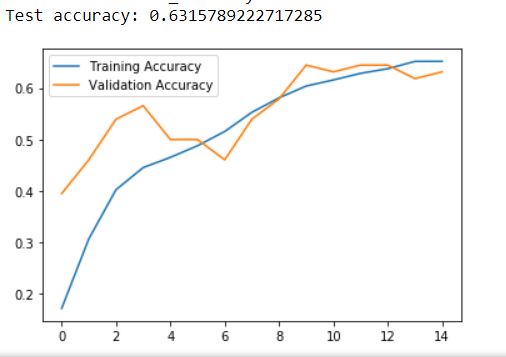


Image 15 Model evaluation with validation split=5%

Using validation split of 5% increases validation accuracy along with training accuracy with number of epochs. At the end of 15 epochs both the values seems to be equal.

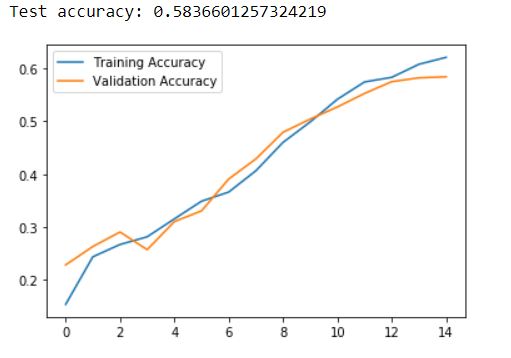


Image 16 Model Evaluation with validation split near 10%

Using validation split of 10% increases validation accuracy along with training accuracy gradually with number of epochs. At the end of 15 epochs both the values seems to be equal.

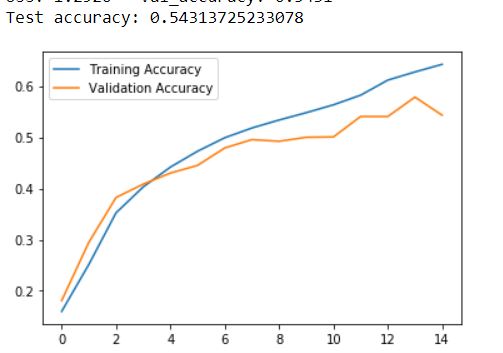


Image 17 Model Evaluation with validation split equal to 20% (Cleaned data)

Using validation split of 20% increases when performed on clean data it was seen that validation accuracy along with training accuracy gradually increases with number of epochs. At the end of 15 epochs both the values seems validation accuracy decreases with respect to training accuracy. Model starts underperforming for unseen data at last stages of epoch cycle.

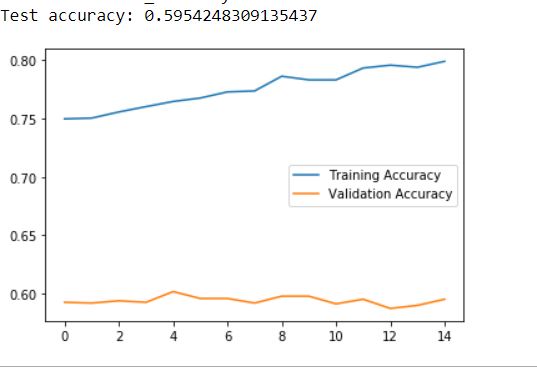
.

Image 18 Model evaluation without data cleaning #Training cycle (>3)

Using validation split of 20% when performed without data cleaning it was seen that validation accuracy was constant with respect to training accuracy being gradually increasing with number of epochs, when model was trained for more than once. At the end of 15 epochs validation accuracy does not seems to be changing its values with respect to training accuracy. Model is underperforming for unseen data from beginning of epoch cycle.

Since glove vectors are used to represent the vectors for words in our corpus words which are not identified in glove vectors (rare occurrence) has all zeros in embedding matrix. If many of words are not identified then matrix is sparse, i.e. most of entries are zero, due to which sometimes model does not perform well on unseen data. Hence, data cleaning is most important stage here to obtain high test accuracy.

Further we also need to perform model evaluation by using cross fold validation technique which can be entailed as future work.

The below code shows the output by a lstm model for sentence “when I entered in the dark room”, model identifies it has a emotion labeled “fear” with 92% result. Here, we can assume that, since lstm persists long term dependencies, model was able to perfectly predict emotion for this sentence.

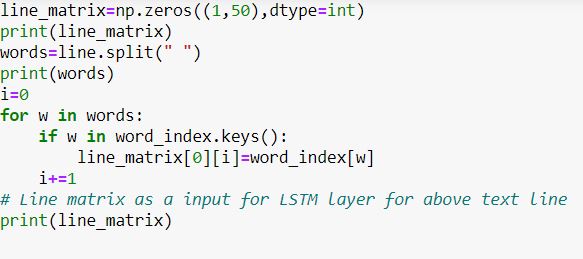


Image 19 Model tested on new input.

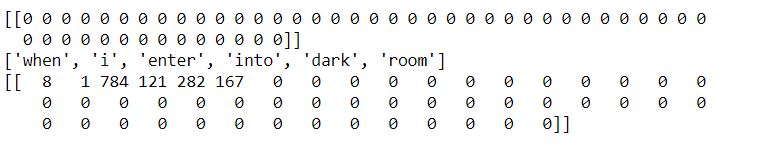


Image 20 Input to lstm layer for new input.

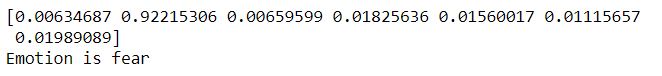


Image 21 Model prediction on new input.

As shown in above figure the output dense layer of dimension (1,7) has second cell with max value equal to 92% which identifies emotion “fear” as vector [0,1,0,0,0,0,0] represents fear in one hot encoding form.

**10.2.2 Using ML.net classification algorithm for dialogue data set**

* We have analyzed the dataset properly, cleaned the data **using open Refine** and after that we have removed 4 columns and trained our model using 3 columns best suited for text classification and emotion analysis.
* We have trained our model using ML.net and that provides readymade algorithm to classify model.
* The accuracy of our model is 76.27% and the Best Model used is AveragedPerceptronOva.
* Evaluated the model to check the efficiency of the result by entering different sentences as input and comparing the result.

Below screen shots shows evaluating model for different input statements to classify emotions. The evaluation results shows that model was quite successful to classify emotions perfectly on the basis of inputs given. This model is able to approximately classify emotions into 30 distinct labels such as terrified, joy, fear, anger etc**.**

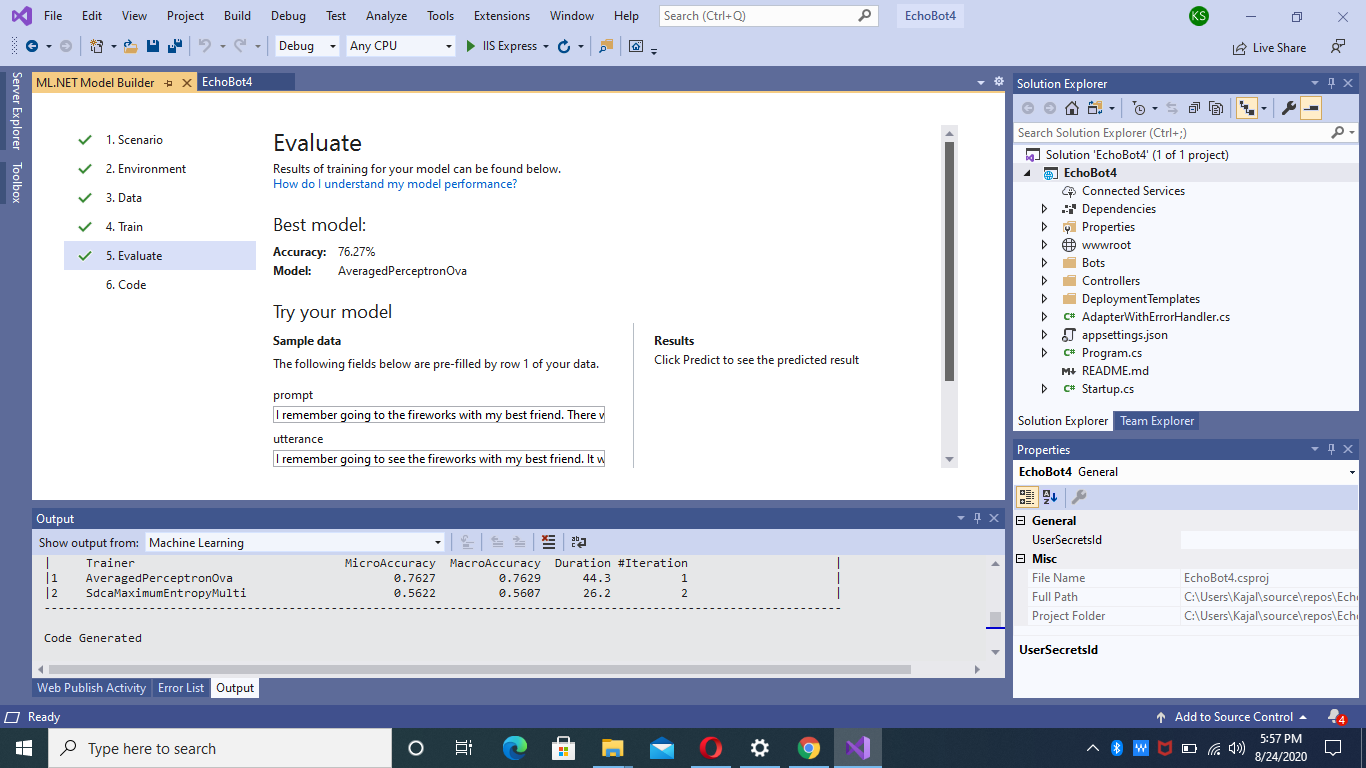
****

Image 22 Evaluating results with ML.net

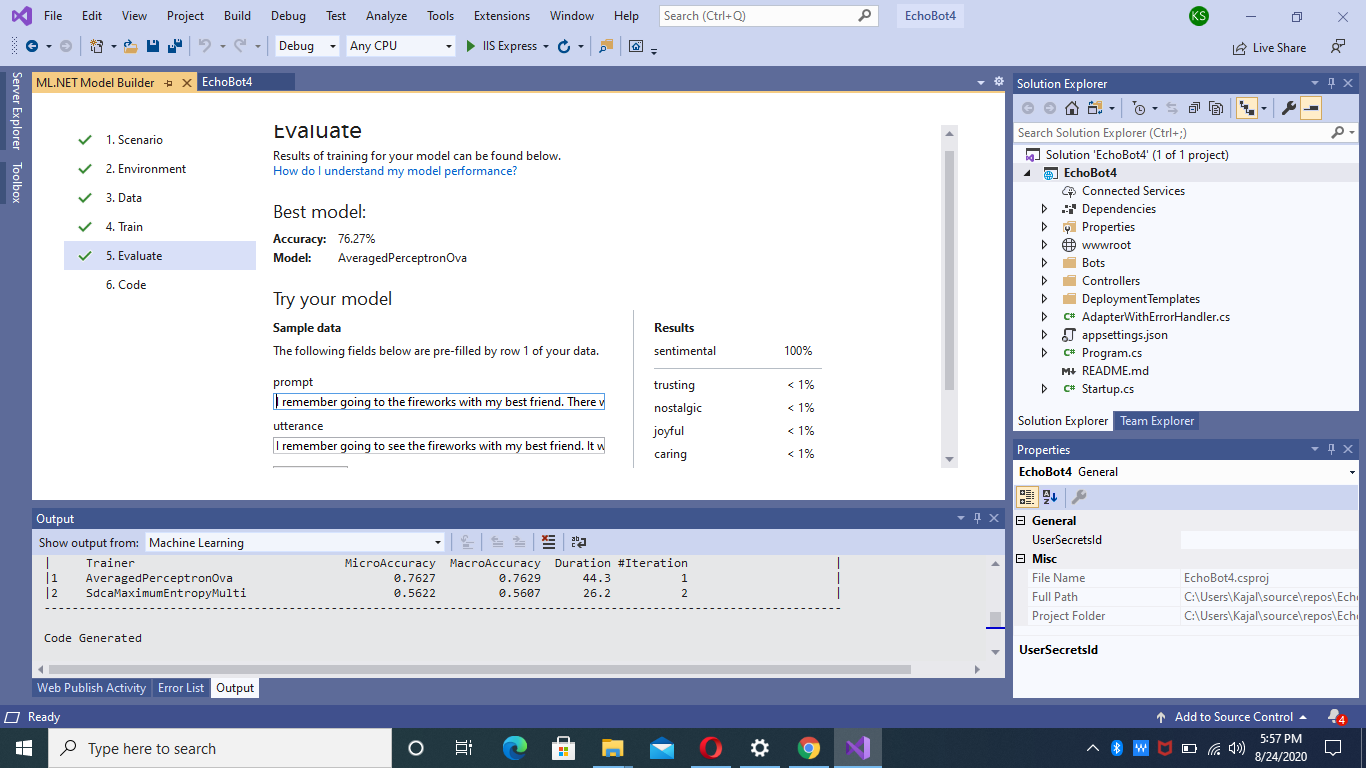
****

Image 23 Evaluating results with ML.net

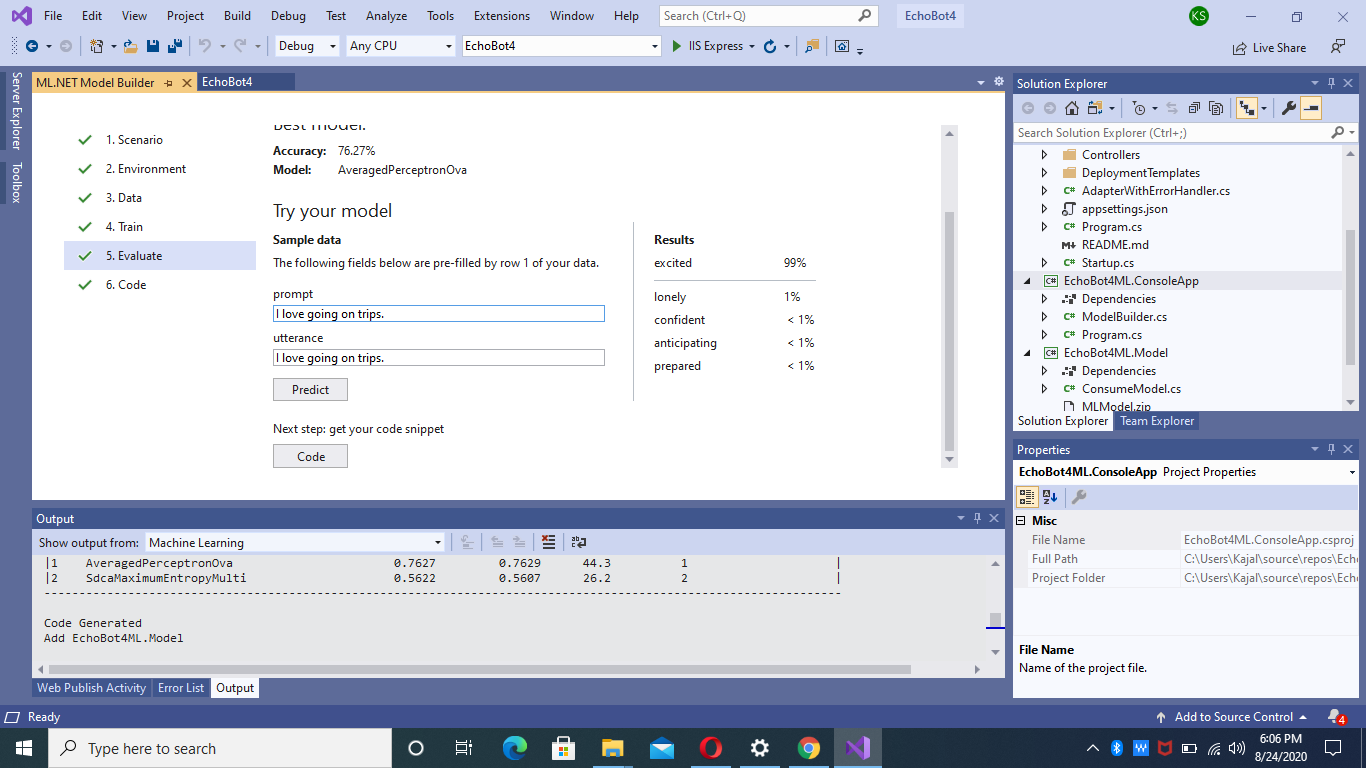
****

Image 24 Evaluating results with ML.net

**10.3**. **Deployment**.

**10.3.1 Deploying ML.net classification model.**

We have decided to publish our emotion identification chat bot built on Ml.net classification algorithm using azure web app service. We were successful in testing our chat bot on Bot framework emulator version 4.0. Some of testing snippets in bot framework emulator is shown in the figure.

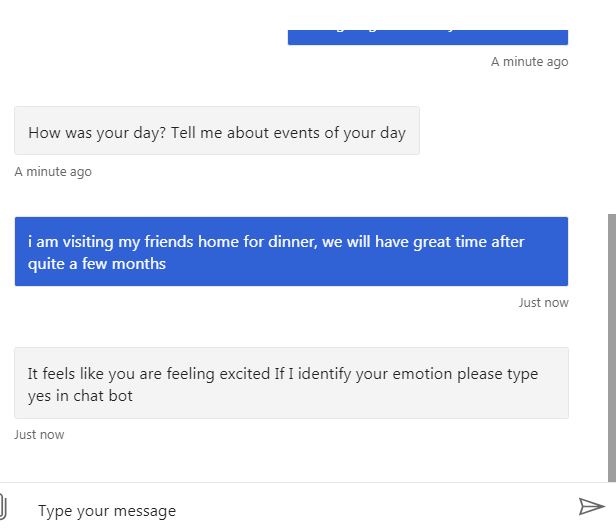


Image 25 Chat bot identifying human emotion

The chat bot also prompts user to say yes if emotion is identified correct. After testing it in a web chat you can easily deploy your bot from visual studio 19 to Microsoft azure cloud service.

Here are some steps to correctly deploy your visual studio bot to Microsoft azure framework.

After testing it on a bot framework, select publish under build project section, and you will be automatically redirected to azure login page, after successfully providing azure login information, create a web app service plan under which you want to host your bot application.

After then, you should select publish by selecting resource group. You can also create new resource group. After completing this you can easily deploy your project on azure using azure command line interface.

Below both figures shows steps of creating azure service plan and publishing your code using azure resource group.

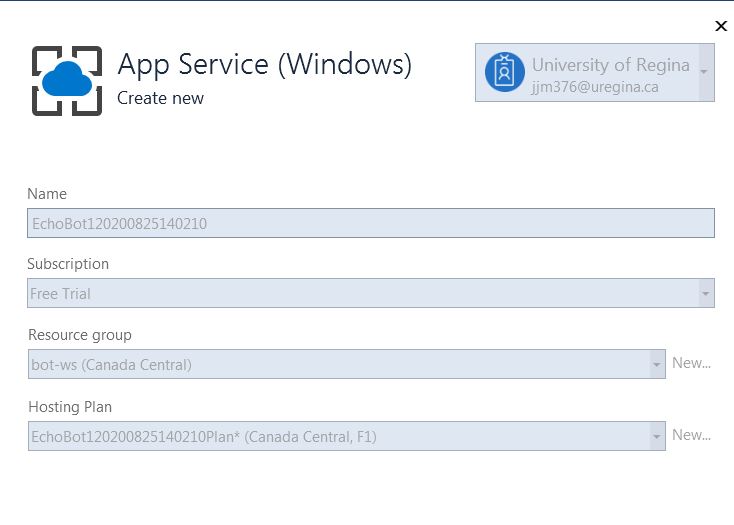


Image 26 Register for Azure App service

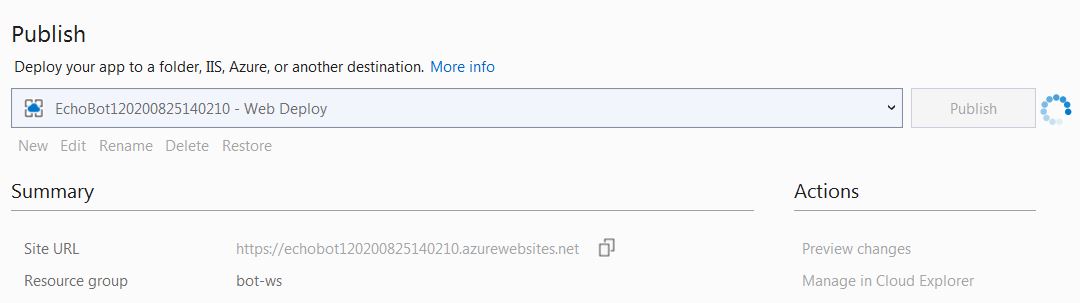


Image 28 Channel for Azure cli

* Further you should select deploy on azure option from the list of options given

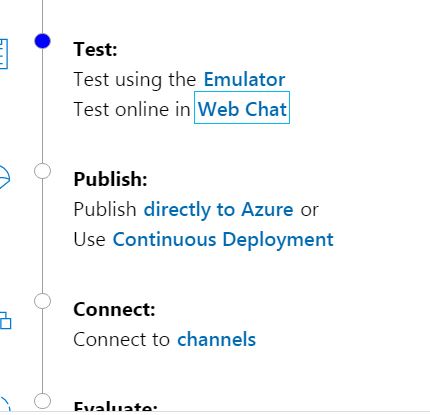


Image 29 Select to deploy on azure service

After then you should login on azure using azure command line interface and by selecting a path to templates-with-preexisting-rg-json you should be able to deploy your project.

**10.3.2 Deploying LSTM model on Azure Machine learning**.

* Workflow for deploying our lstm machine learning model on azure
* Register your model.
* Prepare an inference configuration.
* Prepare an entry script using no code deployment.
* Deploy your model to compute target.
* Test the your model called a web service
* You must call a web service from your Emotion chat bot which is able to interact with the user and make predictions.

**11. Conclusion and Future work.**

The above mentioned draft explains a brief idea of deploying a emotion recognizing chat bots form data collecting to deployment phase on cloud using azure web services. The different Machine learning models are discussed which uses to different data sets to build a emotion recognizing chat bots. The model which uses ML.net classification algorithm on a dialogue data set shows great fficiency on unseen data. Second model uses LSTM recurrent neural network algorithm which classifies input into seven distinct labeled categories which needs to be evaluated further for different interesting results. We need to address following questions while studying LSTM model for classifying emotions from sentences they are

* Does using large corpus data set along with global vectors will increase efficiency of model?

Is there any relationship between word indexing in corpus and model efficiency?