CANDIDATE'S DECLARATION

We declare that pre-final semester report entitled "AI THERAPIST" is our own work conducted under the supervision of the guide Prof.(Dr.)Harshad Prajapati. We further declare that to the best of our knowledge the report for B. Tech. VII semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate's Signature
Candidate's Name: Shubham Patadiya
Student ID: 16ITUON025

Candidate's Signature
Candidate's Name: Preet Nagadia
Student ID: 16ITUOS099

Submitted To: Prof.(Dr.)Harshad Prajapati

Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Gujarat

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project carried out in the subject of Software Design Project entitled "AI Therapist" and recorded in this report. This a bona fide report of work of

1) Shubham Patadiya **Roll No. IT070** **ID No: 16ITUONO25**

2) Preet Nagadia **Roll No.: IT061** **ID No: 16ITUOS099**

of Department of Information Technology, semester VII, . They were involved in Project work during academic year 2019 -2020.

Prof. (Dr.) Harshad Prajapati
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. (Dr.) V . K. Dabhi,
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad.
Date:

ACKNOWLEDGMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them.

We are highly indebted to Dharmsinh Desai University , Faculty Of Technology , Department of IT for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards member of Dharmsinh Desai University for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to our project guide for giving me such attention and time.

Special thanks to Deepak Vegada sir, Anand Dave sir, Mukesh Goswami sir without whom the project could not have been better.

Shubham Patadiya
Preet Nagadia

ABSTRACT

Machine Learning has opened plethora of initiatives which has created a digital ecosystem with user as its cynosure. Consider the never ending Tweets, feeds, thousands of opinions being shared by myriads on Twitter, User has been part of this ecosystem since morning till time its night retirement. However, it has created a façade on mind and psychological state of user which is severe for some users. At present there are no metrics available to discern user's state of mind.

AI Therapist is designed to target this issue. It learns and understands all user's past Tweets. Using Machine Learning Algorithms, it creates a Neural Network and understands Text Message to analyse and Classify the Sentiment towards a subject, Personal Posts or Opinion on any aspect. It then matches polarity of analysed data and recommends Overall sentiment to user. The AI Therapist – Sentiment Analysis is a prototype of the system which provides the whole functionality with Web developing portal to analyse and predict the end result by the Sentiment of the relevant tweets.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Detecting emotional state of a person by analysing a text message written by him/her appear challenging but also essential many times due to the fact that most of the times textual expressions are not only direct using emotion words but also result from the interpretation of the meaning of concepts and interaction of concepts which are described in the text document. Recognizing the emotion of the text plays a key role in the human-computer interaction. Emotions may be expressed by a person's speech, face expression and written text known as speech, facial and text based emotion respectively. Sufficient amount of work has been done regarding to speech and facial emotion recognition but text based emotion recognition system still needs attraction of researchers. In computational linguistics, the detection of human emotions in text is becoming increasingly important from an applicative point of view.

Emotion can be expressed as happiness, sadness, anger, worry, fear, surprise and so forth. While board topic of emotion has been studied in psychology for decades, very little effort has been spent on attempting to detect emotion from text. In this work, we assume that emotion reaction of an input sentence is essentially represented by its word appearance. Automatic emotion detection from text has attracted growing attention due to its potentially useful applications. For examples, psychologists can better assist their patients by analysing their session transcripts for any subtle emotions. Reliable emotion detection can help develop powerful human-computer interaction devices and deep emotional analysis of public data such as tweets and blogs could reveal interesting insights into human nature and behaviour.

1.1 Project Definition

This Application is basically use for detecting emotion from input given by the user. Using natural language processing, our program detects how user is feeling. He may be feeling sad, happy, angry etc. Then a recommender system is used to show him content based on his current emotions. For example, if a user is feeling sad, program will suggest him motivational videos, If a user is already happy then program will suggest him comedy videos.

1.2 Motivation And Purpose

Machine Learning using NLP is an emerging and attractive branch. Emotion Detection using face (Image Processing) and from speech has been accomplished with sufficient accuracy. On the other hand, Emotion Detection on the speech has not reached up to the mark. Big multinational Companies such as Microsoft, Google, Amazon are trying to achieve this task. We are machine Learning enthusiasts and want to contribute to this field. What better way to contribute to this field by helping people to build their mental health?

1.3 Relevant Theory

1.3.1 Implementation of AI Therapist

AI Therapist works in stages as pre-processing, feature extraction and recognition using neural network. Pre-processing includes series of operations to be carried out on text sentence to make it ready for feature Extraction. During Feature Extraction the text sentence is converted to vector of words then feature extraction technique is applied on each word of vector individually. Finally feature vector is presented to the selected algorithm for Emotion Detection. Here this extracted features are provided to Neural Network for recognition of Emotion. Finally based on the emotion detected the suggestion is provided to the user for their mental health.

1.3.2 What is Natural Language Processing

NLP is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.

"Apart from common word processor operations that treat text like a mere sequence of symbols, NLP considers the hierarchical structure of language: several words make a phrase, several phrases make a sentence and, ultimately, sentences convey ideas," John Rehling, an NLP expert at Meltwater Group, said in How Natural Language Processing Helps Uncover Social Media Sentiment. "By analyzing language for its meaning, NLP systems have long filled useful roles, such as correcting grammar, converting speech to text, detecting emotions and automatically translating between languages."

1.3.3 Support Vector Machine

Support Vector Machine(SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).

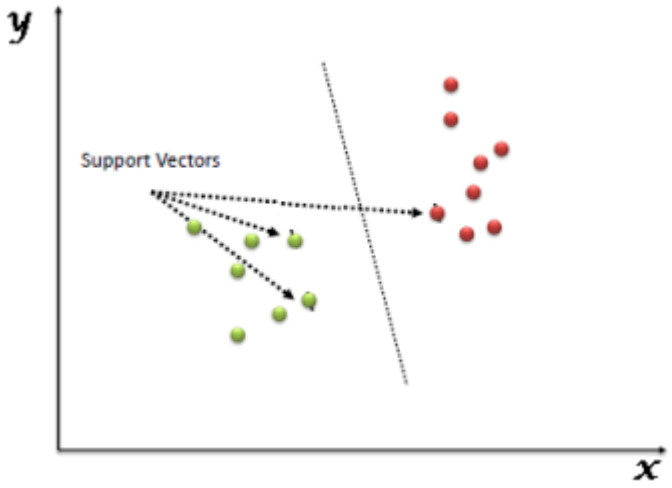


Figure 1.1 Introduction to SVM

Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

1.3.4 Deep Learning

Deep learning can be called a sub-part of Machine learning which is mostly used for learning data representations. This recently exploded technology has improved state of the art techniques used in natural language processing, object detection etc. Deep learning finds out complex structure in massive data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters used in each new layer from the representation in the previous layer. A standard neural network (NN) constitutes of many simple interconnected processors called neurons. Each neuron produces a sequence of activations. The activation of input neurons takes place through sensors that perceive the environment whereas the others get activated through weighted connections from previously active neurons.

1.3.5 Recurrent Neural Network

The idea behind RNNs is to make use of sequential information. In a traditional neural network we assume that all inputs (and outputs) are independent of each other. But for many tasks that's a very bad idea. If you want to predict the next word in a sentence you better know which words came before it. RNNs are called *recurrent* because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Another way to think about RNNs is that they have a "memory" which captures information about what has been calculated so far. In theory RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps (more on this later). Here is what a typical RNN looks like:

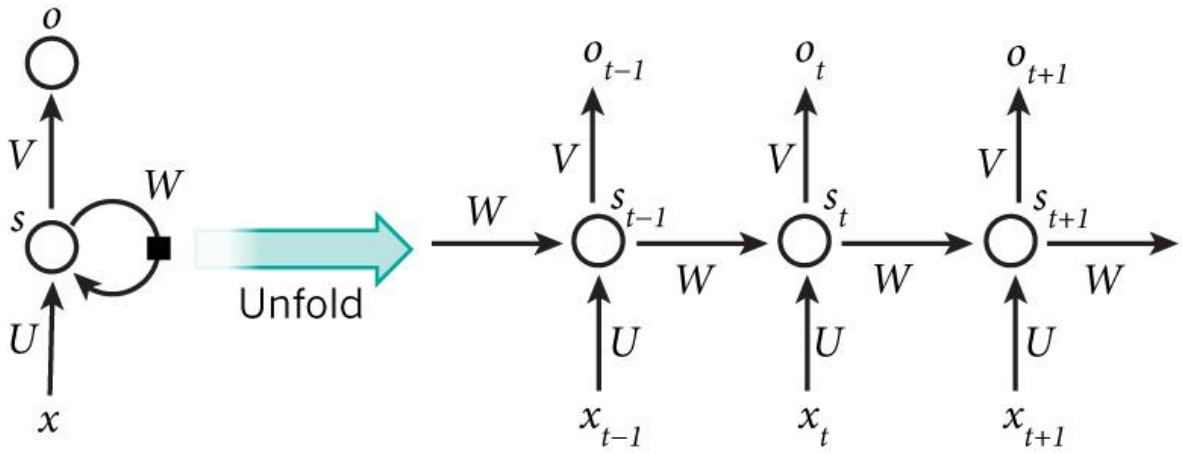


Figure 1.2 Introduction to RNN

The above diagram shows a RNN being *unrolled* (or unfolded) into a full network. By unrolling we simply mean that we write out the network for the complete sequence. For example, if the sequence we care about is a sentence of 5 words, the network would be unrolled into a 5-layer neural network, one layer for each word.

1.3.6 Long Short Term Memory

When we arrange our calendar for the day, we prioritize our appointments right? If in case we need to make some space for anything important we know which meeting could be canceled to accommodate a possible meeting.

Turns out that an RNN doesn't do so. In order to add a new information, it transforms the existing information completely by applying a function. Because of this, the entire information is modified, on the whole, i.e. there is no consideration for 'important' information and 'not so important' information.

LSTMs on the other hand, make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. The information at a particular cell state has three different dependencies.

These dependencies can be generalized to any problem as:

- The previous cell state (i.e. the information that was present in the memory after the previous time step)
- The previous hidden state (i.e. this is the same as the output of the previous cell)
- The input at the current time step (i.e. the new information that is being fed in at that moment)

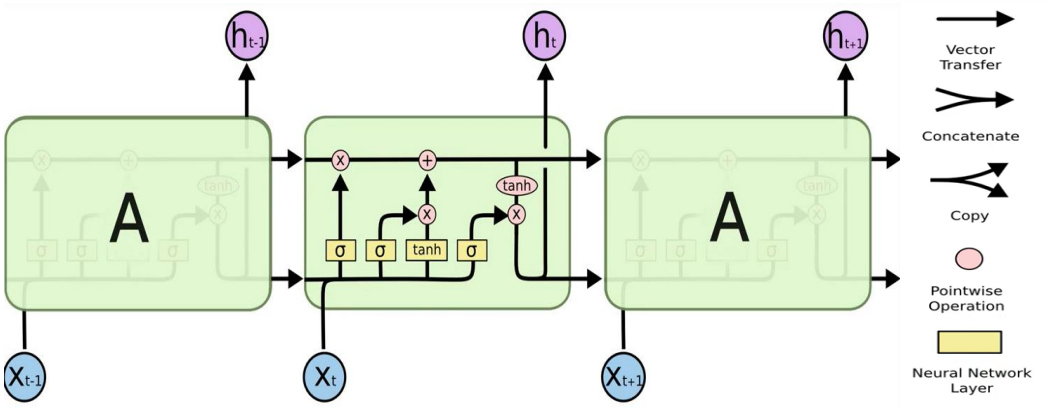


Figure 1.3 Introduction to LSTM

1.3.7 Word Embedding using GloVe

GloVe stands for global vectors for word representation. It is an unsupervised learning algorithm developed by Stanford for generating word embeddings by aggregating global word-word co-occurrence matrix from a corpus. The resulting embeddings show interesting linear substructures of the word in vector space.

1.4 SCOPE

Initially college students and colleagues. Afterwards when we achieve a significant accuracy by gaining more data, we will expand our scope.

1.5 TECHNOLOGY AND LITERATURE REVIEW

- **Python:**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scale. Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

- **Tensorflow**

- **Why is TensorFlow popular**?

TensorFlow is the best library of all because it is built to be accessible for everyone. Tensorflow library incorporates different API to built at scale deep learning architecture like CNN or RNN. TensorFlow is based on graph computation; it allows the developer to visualize the construction of the neural network with Tensorboad. This tool is helpful to debug the program. Finally, Tensorflow is built to be deployed at scale. It runs on CPU and GPU.

Tensorflow attracts the largest popularity on GitHub compare to the other deep learning framework.

- **List of Prominent Algorithms supported by TensorFlow**

Currently, TensorFlow 1.10 has a built-in API for:

- Linear regression: tf.estimator.LinearRegressor
- Classification:tf.estimator.LinearClassifier
- Deep learning classification: tf.estimator.DNNClassifier
- Deep learning wipe and deep: tf.estimator.DNNLinearCombinedClassifier
- Booster tree regression: tf.estimator.BoostedTreesRegressor
- Boosted tree classification: tf.estimator.BoostedTreesClassifier

- **Keras**

One of the most powerful and easy-to-use Python Libraries for developing and evaluating deep learing models is Keras; It wraps the efficient numerical computation libraries Theno and TensorFlow. The advantage of this is mainly that youbcan get started with neural network in an easy and fun way.

- **Tkinter**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object oriented interface to the Tk GUI toolkit.

1.6 OBJECTIVE

The main objective to develop this application is that user can get therapy according to his or her mood. i.e. If one is depressed or sad than get motivational quotes to overcome the depression and get relief.

2 PROJECT MANAGEMENT

2.1 FEASIBILITY STUDY

2.1.1 Technical feasibility

Technical analysis evaluates technical merits of the system at the same time collects additional information about performance, reliability, maintainability and productivity. The technical feasibility means that the project can be done with the current equipment, existing software technology and the current knowledge. The present system is technically feasible as it has been developed on Google Colab for professional developers and Tkinter for Frontend development. We are using the dataset of Emotions from Kaggle.

Eliza was the first program which was created based on AI in 1966. The program can talk to you on the basis of your mood and that mood was detected from the text you enter. Hence, based on this program we can say that AI Therapist is possible to implement.

Eliza: http://www.med-ai.com/models/eliza.html

2.1.2 Time Schedule Feasibility

The Project has simple working and the basic requirement can be satisfied within allotted time period so the time development feasibility is satisfied.We are trying to implement as many feature Vector as possible. More we add the feature Vector more we get the accuracy. All we needed to do is collect as much feature vector as possible and then implement it. We will create Front End and then collect data-set as fast as possible. Then we need to spend all our time on extracting feature vector.

2.1.3 Operational Feasibility

There are mainly two operations: Analysis of the data and render the data to user end. The analysis take place in python environment and render the data in the same environment using Frontend with use of trained model. Each operation has been working completely as an individual and so that the operational feasibility is satisfied.

2.1.4 Implementation Feasibility

The system can be efficiently implemented as it has been developed by using Google colab which provides high computational power. Implementation feasibility is concerned with specifying external resources and software that will successfully satisfy the requirements. We will use pre-trained model to implement the program in python Environment. This pre-trained model will be trained on google colab.

2.2 PROJECT PLANNING

2.2.1 Project Development approach and justification.

For Project Development Iterative Waterfall Model is used. Iterative waterfall model:

The Iterative water fall model approach overcomes the problems associated with the waterfall model approach. If any difficulty or problem encounter in any phase may require going back to the previous phase and performing the required modifications and proceeds sequentially. This backtracking allows modifying any corrections or modifications required in the previous phase.

As illustrated in Fig 2.1, this model divides the cycle into the phases mentioned below:

1. Feasibility Study.
2. Requirements analysis and specification.
3. Design.
4. Coding and Unit Testing.
5. Integration and System Testing.
6. Maintenance.

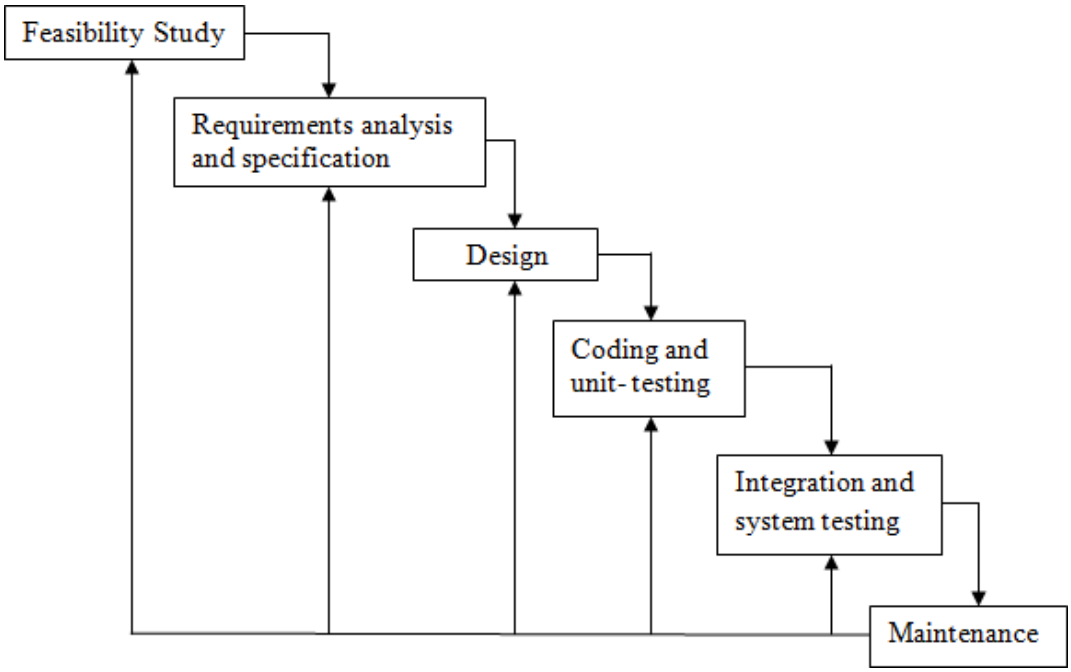


Figure 2.1 Iterative Waterfall Model

2.2.1.1 Advantages of using Iterative Waterfall Model:

- You are provided the chance to see the potential outcomes of every stage and make changes to areas of concern if necessary. This is
- one of the reasons that make the iterative model useful.
- Iterative development is more adjustable to changes as it considers each stage like a vital portion of the development cycle.
- The time spent on each successive interval may be lessened depending on how the last stage went and what knowledge was gained from past stages. The system therefore grows through adding new functionalities in the development part of all iterations.

2.2.1.2 Disadvantages of using Iterative Waterfall Model:

- When using the iterative model people working on the project can get stuck in a loop. Always finding problems than having to go back and design a fix, implement it, then test the system again and finding another problem can mean that the project can run over time and budget.
- Informal requests for improvement after each phase may lead to confusion and may also create scope creep, since user feedback following each phase may lead to increased customer demands. As users see the system develop, they may realize the potential of other system capabilities which would enhance their work, this can be an advantage as much as it can be a disadvantage.

2.2.1.3 Justification:

After feasibility study as the functional requirements were almost clear. Here we have decomposed the system into modules. That is why we decided to use iterative waterfall model which is most suitable model here i.e. if we find any difficulty in coding and testing a modification in design can be done easily.

2.3 Project Plan

After feasibility study as the functional requirements were almost clear which were decided by our project lead. After analyzing and thoroughly understanding the requirements of the application we planned the project.

The project had been planned in mainly two phases: Designing and Implementation. Designing phase consists of the road map of the system. It has been determined which frameworks and language gives optimal solution for the system and then in Implementation, it has been used to successfully accomplish the project.

2.3.1 Milestones and Deliverables

Timely directions are always required to run a project successfully. Milestones tell the developers how far he has reached and also tell him what things are still left and how to fulfil them. Milestones may be the short report of achievement in project activity that are used by the project manager to check project progress but which

are not delivered to the Clients. The deliverables are the project results that are provided to the customer. It is usually delivered at the end of some major project phases.

Table 1 Milestones and Deliverables

MILESTONES	DELIVERABLES	PURPOSE
Software Installation and Understanding of Technology.	Had complete knowledge of Python, Google colab and its features.	To be familiar with Technology.
System feasibility study, Requirement and Analysis.	Functional Specifications. Non Functional Specifications.	It gives exact understanding of the User's requirements.
System Design.	Dataset Design Sequence Diagram Use Case Diagram Form Layouts	It gives the logical Structure that describes the system.
Coding and Unit Testing and corrections if any.	Individually Tested and Functional Modules. Individual Modules for Sentiment Analysis	It gives the required Module.
Integration and System Testing.	The output obtained for the required functionality after implementing and doing various types of testing.	Integrated System is Ready.

2.3.2 Roles and Responsibilities

As only two members were involved in the whole team each of them had to perform all the tasks as the project proceeded through its different phases. This helped each one to develop skills in all the phases.

Table 2 Roles and Responsibilities

Name	Role				
	Analysis	Designing	Coding	Testing	Documentation
Shubham Patadiya	✓	✓	✓	✓	✓
Preet Nagadia	✓	✓	✓	✓	✓

2.4 PROJECT SCHEDULING

Scheduling the project tasks is an important project planning activity. It involves deciding which tasks should be taken up and when. In order to schedule the project activities, a software project manager needs to do the following:

- Identify all the tasks needed to complete the project.
- Break down large tasks into small activities.
- Determine dependencies amongst different activities.
- Establish most likely estimates for the time durations necessary to complete the activities.
- Allocate resources to activities.
- Plan the starting and ending dates for various activities.

Table 3 Task Duration

ID	Task Name	Start	Finish	Duration
1	Feasibility Study	13/07/19	16/07/19	4d
2	Requirements Gathering	16/07/19	17/07/19	2d
3	Analysis	17/07/19	19/07/19	3d
4	Design	28/07/19	17/08/19	3w
5	Coding	18/08/19	31/08/19	3w
6	Testing	02/09/19	28/09/19	3w

3 SYSTEM REQUIREMENTS STUDY

3.1 STUDY OF CURRENT SYSTEM

The current systems mainly work in backend model in which feature extraction is performed on the text sentence and based on that features emotion is detected. There is no system that provides suggestion based on your emotion will be shown to the user.

3.2 PROBLEMS AND WEAKNESSES OF CURRENT SYSTEM

System incorporates the basic analysis of ongoing topics by using machine learning algorithms. The classifiers produce around 79% accuracy for tweets across all domains. But there is extensibility in program that we have used. Program can suggest some youtube videos based on your moods.

3.3 USER CHARACTERISTICS

The targeted users can be any individual who wants to know his/her personality traits. Also, user can know about sentiment of any ongoing topic.

3.4 HARDWARE AND SOFTWARE REQUIREMENTS

3.4.1 Software requirements

The user should have Python friendly environment in which the libraries that are to imported to execute the program can be done.

3.4.2 Hardware requirements

Any processor will be compatible with the system, better the processor better will be the responsiveness of the system. No need of high graphic card.

3.5 CONSTRAINTS

3.5.1 Hardware limitations

There are no hardware limitations for this system because once the complete system is developed care would be taken while deploying system so necessary pre-requisites are met.

3.5.2 Interface to other applications

There are no other systems that use this application as an interface.

3.5.3 Reliability Requirements

The application does demand much reliability and it is fully assured that the particular information about the user should be secured and flow is maintained and accessed according to the rights.

3.5.4 Criticality of the Application

The application currently run on the Client-Side Machine. No Internet is required to execute the program. So, no need to worry about criticality.

3.5.5 Security and Safety considerations

The system provides the tight security for output of the sentiment analysis and is secured at the developer end.

3.6 ASSUMPTIONS AND DEPENDENCIES

Assumptions are described as follows:

- Text that is entered by the User is formatted properly in terms of Grammar.
- User is not using sarcasm.
- User is not entering sentence that is too large to process.

4 SYSTEM ANALYSIS

4.1 REQUIREMENTS OF NEW SYSTEM

4.1.1 Functional Requirement

Name: Emotion Detection.

Input: text from textbox

Output: Emotion Emoji

Process: Detects emotion from the sentence received from user using trained model

4.1.2 Non-Functional Requirement

4.1.2.1 Performance:

System must utilize less amount of resource as well should consume less response time of the consumer request.

4.1.2.2 Availability:

The system should available from anywhere and at any time.

4.1.2.3 Reliability:

The system must will show the same output for same input for years, until the training data-set gets modified.

4.1.2.4 Supportability:

The system needs to be cost-effective to maintain. Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documentation, e.g. which test cases and test plans will accompany the system.

4.2 FEATURES OF NEW SYSTEM

The new system consists the features of user interactive functionality along with existing features. User gets suggestion based on their emotion from the system. This suggestion system is based on the emotion and generated dynamically.

4.3 NAVIGATION DIAGRAM

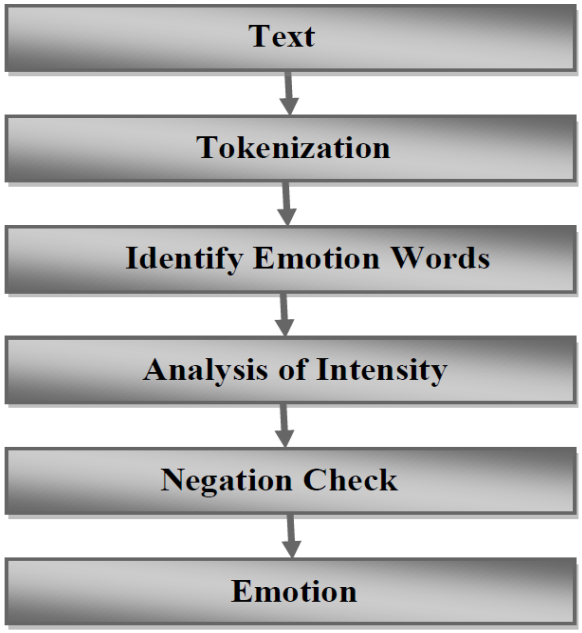


Figure 4.1Navigation Diagram

Step 1: Input Text will be sent to get Tokenize

Step 2: After Tokenizing the sentence, obtained words will be removed if they come under the stopwords criteria

Step 3: Identify the emotions words which are important and the lemmatize the word vector.

Step 4: Analysis of each word in the word vector is done based on their intensity.

Step 5: Output the emotion.

4.4 SYSTEM ACTIVITY (USE CASE OR SCENARIO DIAGRAM)

The use case view models functionality of the system as perceived by outside uses. A use case is a coherent unit of functionality expressed as a transaction among actors and the system.

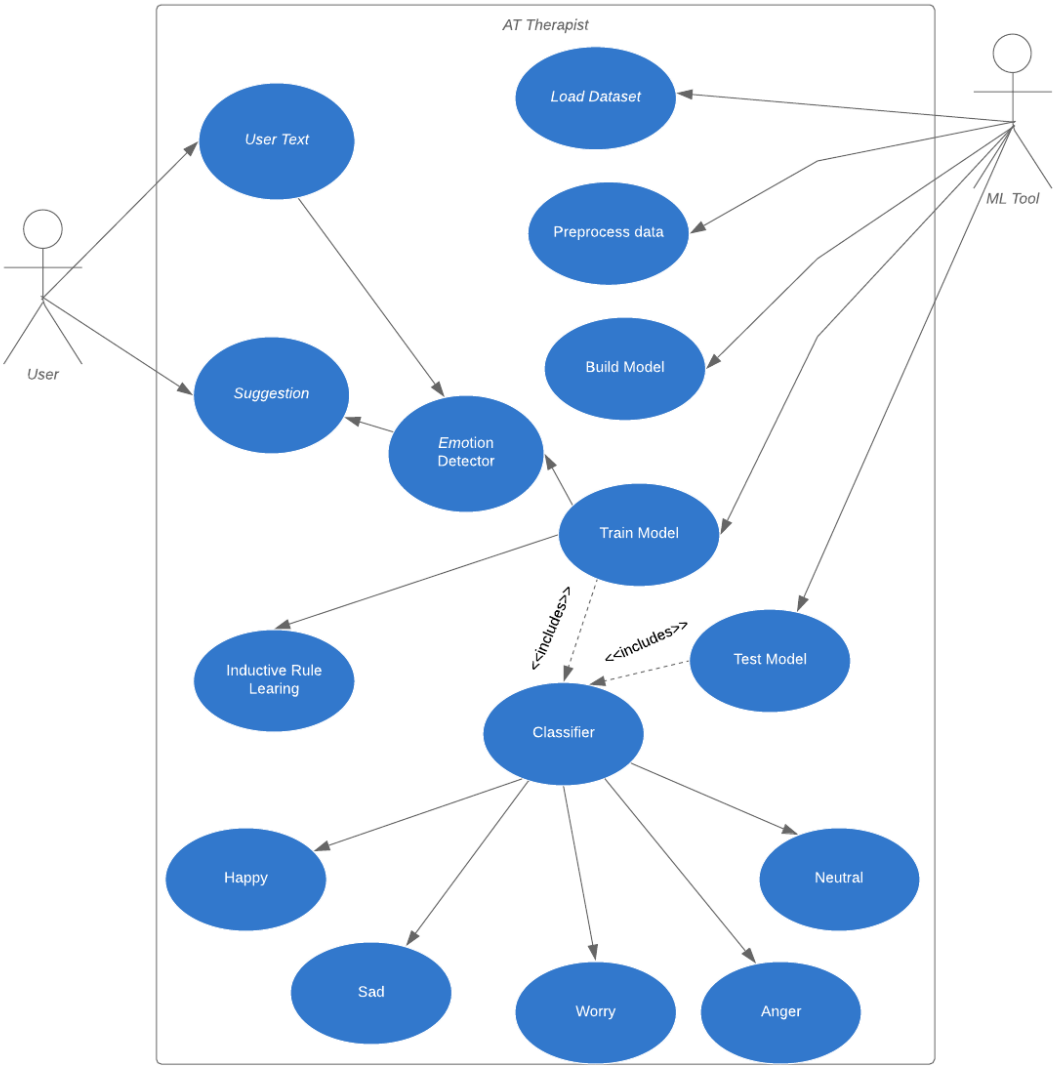


Figure 4.2 System Activity (Use Case diagram)

4.5 Sequence Diagram

A sequence diagram is a graphical view of a scenario that shows object interaction in a time- based sequence what happens first, what happens next. Sequence diagram establish the role of objects and helps provide essential information to determine class responsibilities and interfaces. This type of diagram is best used during early analysis phase in design because they are simple and easy to comprehend. Sequence diagram are normally associated with use cases.

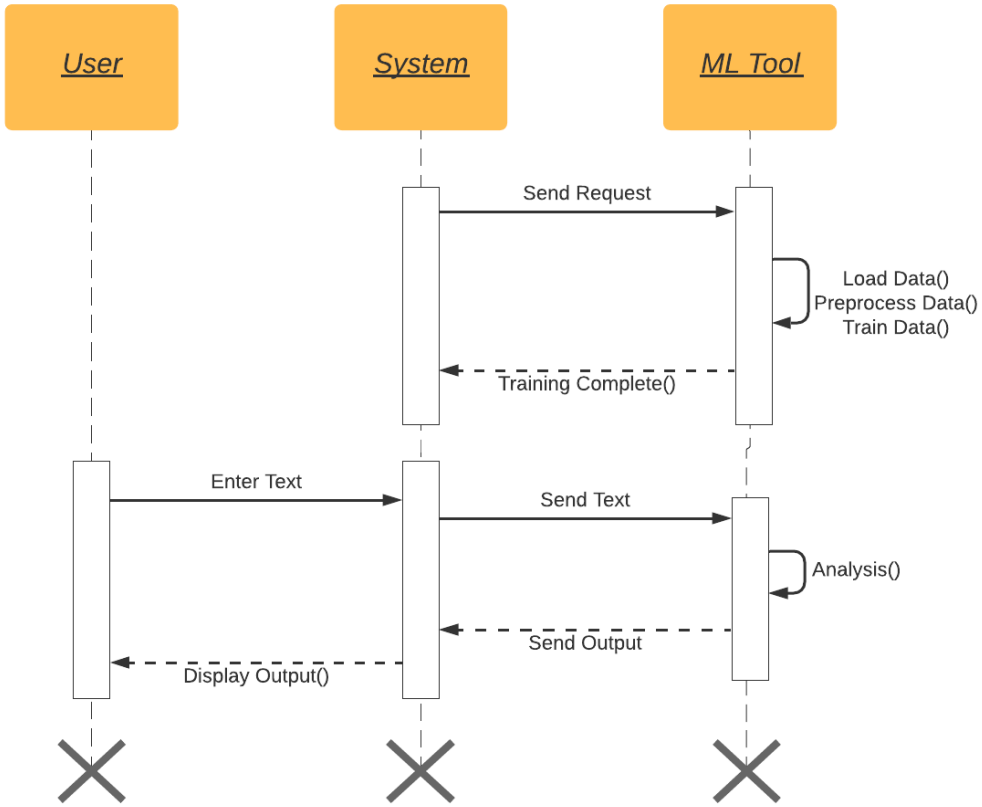


Figure 4.3 Sequence Diagram

4.6 Data Flow Diagram

Data flow diagram (DFD) is also called as Bubble Chart is a graphical technique, which is used to represent information flow, and transformers those are applied when data moves from input to output. DFD represents system requirements clearly and identify transformers those becomes programs in design. DFD may further partitioned into different levels to show detailed information flow e.g. level 0, level 1

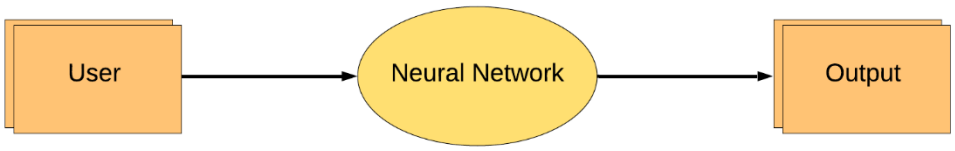


Figure 4.4 DFD(level 0) for AI Therapist

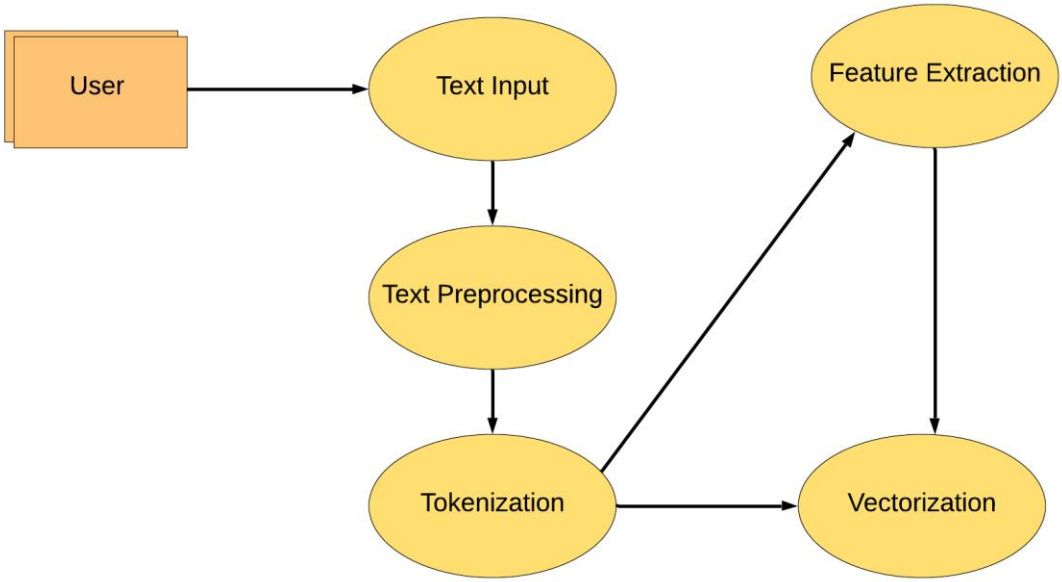


Figure 4.5 DFD(level 1) for Emotion Detection

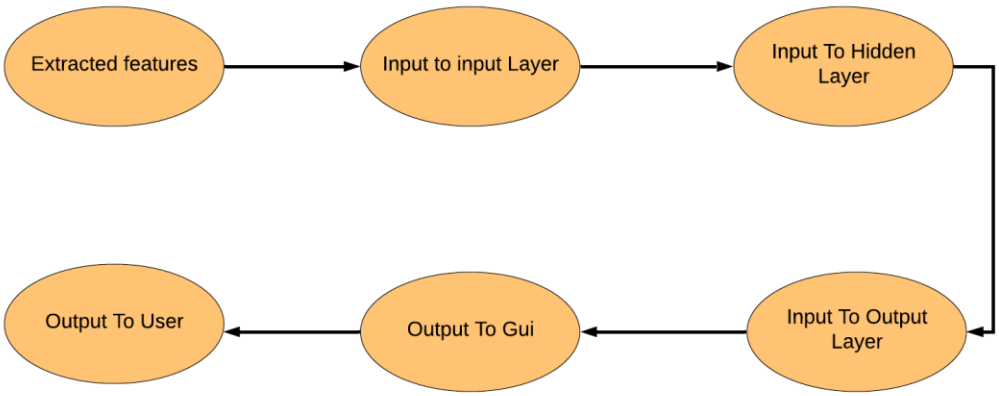


Figure 4.6 DFD(level 1) for Suggestion Based on the Emotion

5 TECHNICAL SPECIF3ICATION

5.1 Technology Details Used in Project

- **Installing Tensorflow on Ubuntu**

- Install TensorFlow with pip
- TensorFlow 3 packages are available:
- tensorflow —Latest stable release for CPU-only
- tensorflow-gpu —Latest stable release with GPU support (Ubuntu and Windows)
- tf-nightly —Preview build (unstable). Ubuntu and Windows include GPU support.

- **System Requirements**

- pip 19.0 or later
- Ubuntu 16.04 or later (64-bit)
- macOS 10.12.6 (Sierra) or later (64-bit) (no GPU support)
- Windows 7 or later (64-bit) (Python 3 only)
- Raspbian 9.0 or later

- **Hardware Requirements**

- Starting with TensorFlow 1.6, binaries use AVX instructions which may not run on older CPUs.
- Read the GPU support guide to set up a CUDA®-enabled GPU card on Ubuntu or Windows.

- **Installation Steps on Ububtu**

```
sudo apt update
sudo apt install python3-dev python3-pip
sudo pip3 install -U virtualenv  # system-wide install
```

- **Installing Keras on Ubuntu**

```
sudo apt-get install python3-pip python3-dev
```

- **Installing Tkinter on Ubuntu**

```
sudo apt install python3-tk
```

- **Installing Pandas on Ubuntu**

```
pip install numpy
```

- **Installing NumPy on Ubuntu**

```
pip install pandas
```

6 DATASET DESCRIPTION

The field of textual emotion detection is still very new and the literature is fragmented in many different journals of different fields. It is really hard to get a good look on what is out there.

Note that there a several emotion theories psychology. Hence there a different ways of modelling /representing emotions in computing. Most of the times "emotion" refers to a phenomena such as anger, fear or joy. Other theories state that all emotions can be represented in a multi-dimensional space (so there is an infinite number of them).

Training and test datasets are provided for four emotions: joy, sadness, fear, and anger. For example, the anger training dataset has tweets along with a label of anger class. The test data includes only the tweet text and label for correct output.

- **Data fields**

- id - id of a row
- sentiment - emotion of text
- content - the text itself

7 PYTHON CODE FOR TRAINING MODEL

- **Import Libraries.**

```
[ ]  #importing libraries
     import pandas as pd
     import tensorflow as tf
     import numpy as np
     from nltk.corpus import stopwords
     from sklearn.feature_extraction.text import CountVectorizer
     from tensorflow.keras.preprocessing.text import Tokenizer
     from tensorflow.keras.preprocessing.sequence import pad_sequences
     from tensorflow.keras import models
     from tensorflow.keras import layers
     from sklearn.model_selection import train_test_split
     from tensorflow.keras.utils import to_categorical
     import re
     from tensorflow.keras import regularizers
```

- **Import Dataset and drop other emotions.**

```
#importing Data
data = pd.read_csv('train_data.csv',encoding='ISO-8859-1')
#data = data.drop(data[data.sentiment == 'anger'].index)
#data = data.drop(data[data.sentiment == 'neutral'].index)
#data = data.drop(data[data.sentiment == 'worry'].index)
data = data.drop(data[data.sentiment == 'boredom'].index)
data = data.drop(data[data.sentiment == 'enthusiasm'].index)
data = data.drop(data[data.sentiment == 'empty'].index)
data = data.drop(data[data.sentiment == 'fun'].index)
data = data.drop(data[data.sentiment == 'relief'].index)
data = data.drop(data[data.sentiment == 'surprise'].index)
data = data.drop(data[data.sentiment == 'love'].index)
data = data.drop(data[data.sentiment == 'hate'].index)
```

- **Dataset Example**

```
[ ]  data_frame_train
```

	sentiment	content
1	sadness	Layin n bed with a headache ughhhh...waitin o...
2	sadness	Funeral ceremony...gloomy friday...
4	neutral	@dannycastillo We want to trade with someone w...
5	worry	Re-pinging @ghostridah14: why didn't you go to...
6	sadness	I should be sleep, but im not! thinking about ...
7	worry	Hmmm. http://www.djhero.com/ is down
8	sadness	@charviray Charlene my love. I miss you
9	sadness	@kelcouch I'm sorry at least it's Friday?
10	neutral	cant fall asleep

- **Data Pre-Processing:**

- Converting whole sentence into lower case
- Removing Punctuations from the sentence
- Removing Stop Words using NLTK

```
#datapreprocessing
#lowercassing letters
data_train['content'] = data_train['content'].apply(lambda x: " ".join(x.lower() for x in x.split()))
# Removing Punctuation, Symbols
data_train['content'] = data_train['content'].str.replace('[^\w\s]',' ')
# Removing Stop Words using NLTK
stop = stopwords.words('english')
```

- **Preparing data for Neural Network**

```
max_features = 10000
MAX_SEQUENCE_LENGTH = 15
tokenizer = Tokenizer(num_words=max_features, split=' ')
tokenizer.fit_on_texts(data_train['content'].values)
word_index = tokenizer.word_index
X = tokenizer.texts_to_sequences(data_train['content'].values)
X = pad_sequences(X,maxlen=MAX_SEQUENCE_LENGTH)
```

```
[ ]  X.shape
```

```
(21685, 15)
```

- **Splitting the data for training and validation**

```
[ ]  X_train,X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.1,random_state=42)
```

- **Downloading the Glove Embeddings.**

```
[ ]  !wget http://downloads.cs.stanford.edu/nlp/data/glove.twitter.27B.zip
```

```
--2019-10-18 05:53:36--  http://downloads.cs.stanford.edu/nlp/data/glove.twitter.27B.zip
Resolving downloads.cs.stanford.edu (downloads.cs.stanford.edu)... 171.64.64.22
Connecting to downloads.cs.stanford.edu (downloads.cs.stanford.edu)|171.64.64.22|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1520408563 (1.4G) [application/zip]
Saving to: 'glove.twitter.27B.zip'

glove.twitter.27B.z 100%[===================>]   1.42G  2.08MB/s    in 11m 43s

2019-10-18 06:05:20 (2.06 MB/s) - 'glove.twitter.27B.zip' saved [1520408563/1520408563]
```

```
[ ]  !unzip "glove.twitter.27B.zip"
```

```
Archive:  glove.twitter.27B.zip
  inflating: glove.twitter.27B.25d.txt
  inflating: glove.twitter.27B.50d.txt
  inflating: glove.twitter.27B.100d.txt
  inflating: glove.twitter.27B.200d.txt
```

- **Load Embeddings**

```
[ ]  def load_embeddings():
         embeddings_index = {}
         f = open('./glove.twitter.27B.100d.txt','r')
         for line in f:
             values = line.split()
             word = values[0]
             coefs = np.asarray(values[1:],dtype='float32')
             embeddings_index[word] = coefs
         f.close()
         print('Found %s word vectors' %len(embeddings_index))
         return embeddings_index
```

```
[ ]  embedding_index = load_embeddings()
```

```
Found 1193514 word vectors
```

- **Feature Extraction from embeddings.**

```
[ ]  embedding_dim = 100
     embedding_matrix = np.zeros((max_features,embedding_dim))
     for word,i in word_index.items():
         if i<max_features:
             embedding_vector = embedding_index.get(word)
             if embedding_vector is not None:
                 embedding_matrix[i] = embedding_vector
```

- **Defining the Model**

```
[ ]  embed_dim = 100
     lstm_out = 512

     model = models.Sequential()
     model.add(layers.Embedding(max_features, embed_dim,embeddings_initializer='uniform',input_length = MAX_SEQUENCE_LENGTH))
     model.add(layers.SpatialDropout1D(0.5))
     model.add(layers.LSTM(lstm_out,dropout=0.5, recurrent_dropout=0.2,kernel_regularizer=regularizers.l2(0.01)))
     model.add(layers.Dense(5,activation='softmax'))
     model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
     print(model.summary())
```

- **Training of the Neural Network**

```
[ ]  batch_size = 32
     history = model.fit(X_train, Y_train, epochs = 50, batch_size=batch_size,validation_data=(X_test,Y_test))
```

- **Epochs**

```
Epoch 45/50
19516/19516 [==============================] - 29s 1ms/sample - loss: 0.8802 - acc: 0.6636 - val_loss: 1.4563 - val_acc: 0.4449
Epoch 46/50
19516/19516 [==============================] - 29s 2ms/sample - loss: 0.8714 - acc: 0.6665 - val_loss: 1.4915 - val_acc: 0.4444
Epoch 47/50
19516/19516 [==============================] - 29s 2ms/sample - loss: 0.8645 - acc: 0.6806 - val_loss: 1.5029 - val_acc: 0.4320
Epoch 48/50
19516/19516 [==============================] - 29s 1ms/sample - loss: 0.8605 - acc: 0.6748 - val_loss: 1.4990 - val_acc: 0.4426
Epoch 49/50
19516/19516 [==============================] - 29s 1ms/sample - loss: 0.8526 - acc: 0.6786 - val_loss: 1.4868 - val_acc: 0.4463
Epoch 50/50
19516/19516 [==============================] - 29s 1ms/sample - loss: 0.8489 - acc: 0.6799 - val_loss: 1.5274 - val_acc: 0.4417
```

8 Introduction To GUI With Tkinter In Python

Tkinter is an inbuilt Python module used to create simple GUI apps. It is the most commonly used module for GUI apps in the Python. You don't need to worry about installation of the Tkinter module as it comes with Python default.

8.1 AI Therapist GUI

- **Import Libraries.**

```
import the module tkinter.
import PIL.
import Image Tk, Image.
import webbrowser.
Import tkFont.
```

- **Load model**

```
Import load_model from keras Library
From tensorflow.keras import models
Load .h5 file of your trained model.
```

- **Define_main_method**

- In this method we create the object of Tk.
- Rename the title of the window as you like with the master.title('AI Therapist').
- Set the geometry axis.
- Create frame object.
- Create Label for user interface.
- Call finalFunction method which will subsequently call other methods according to needs.
- Pack the Emoji Image.
- Run mainloop method.

- **Create showEmoji function**

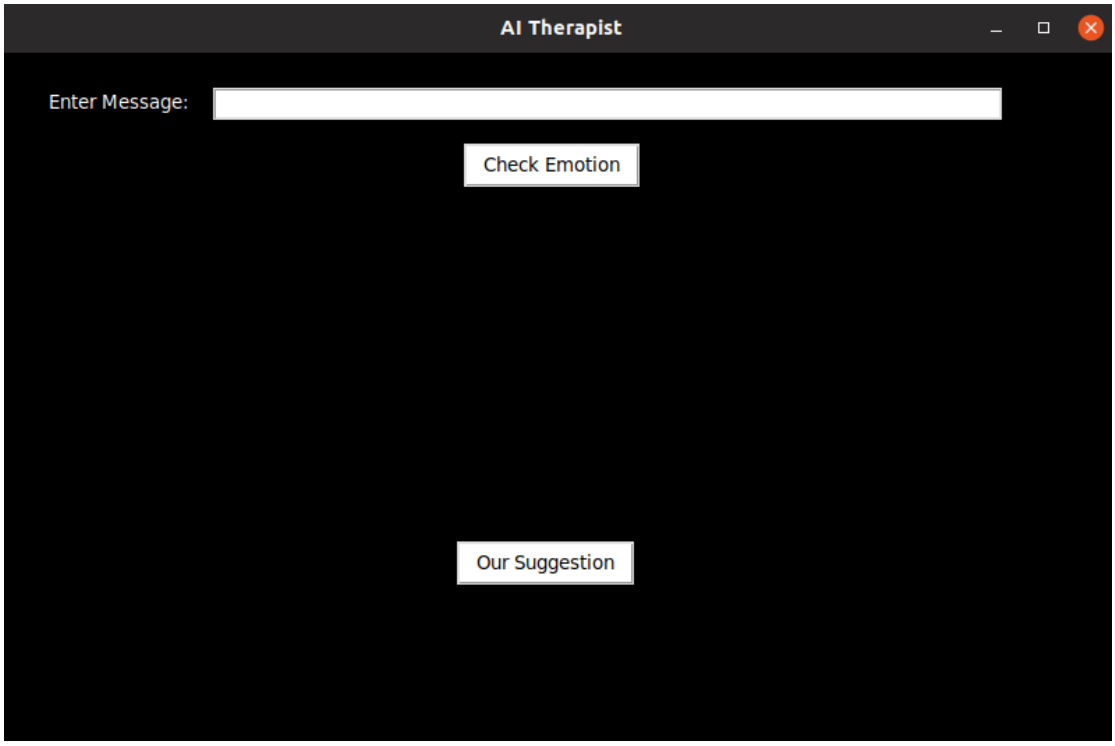
- In this function according to the finalFunction the output is passed as parameter and the emoji<name>.png file is shown on to the screen.
- This function will be called automatically after completion of finalFunction. And will always update after there is an update on the text message(emotion)
- Bind the method for button click events.

- **Define method for Predict onclick event.**

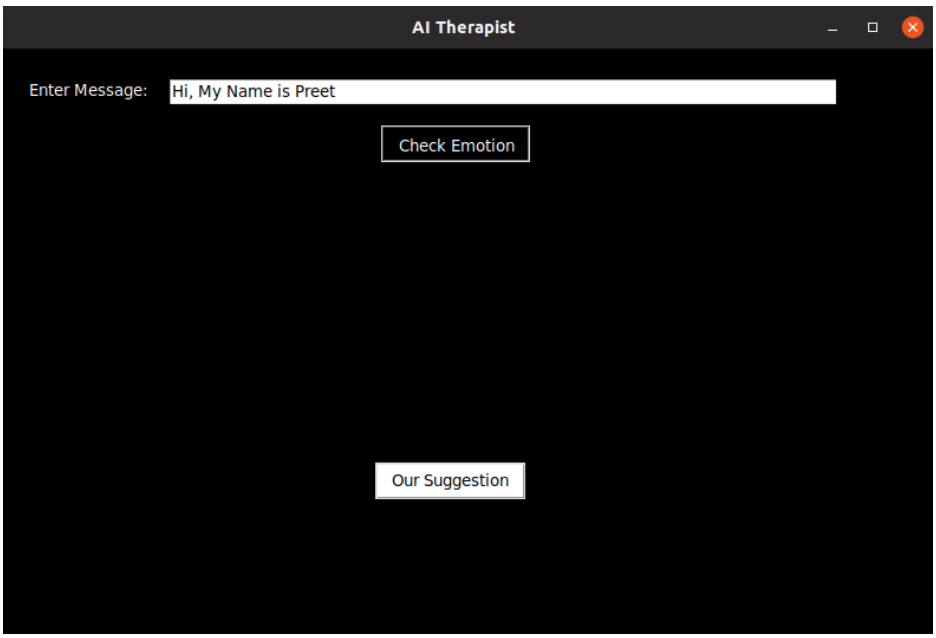
- First store the text sentence in the string which is written on the textbox.
- Preprocess that text string so that our trained model can do prediction and give output.
- Call predict method of model object.

Define method for Suggestion button onclick event

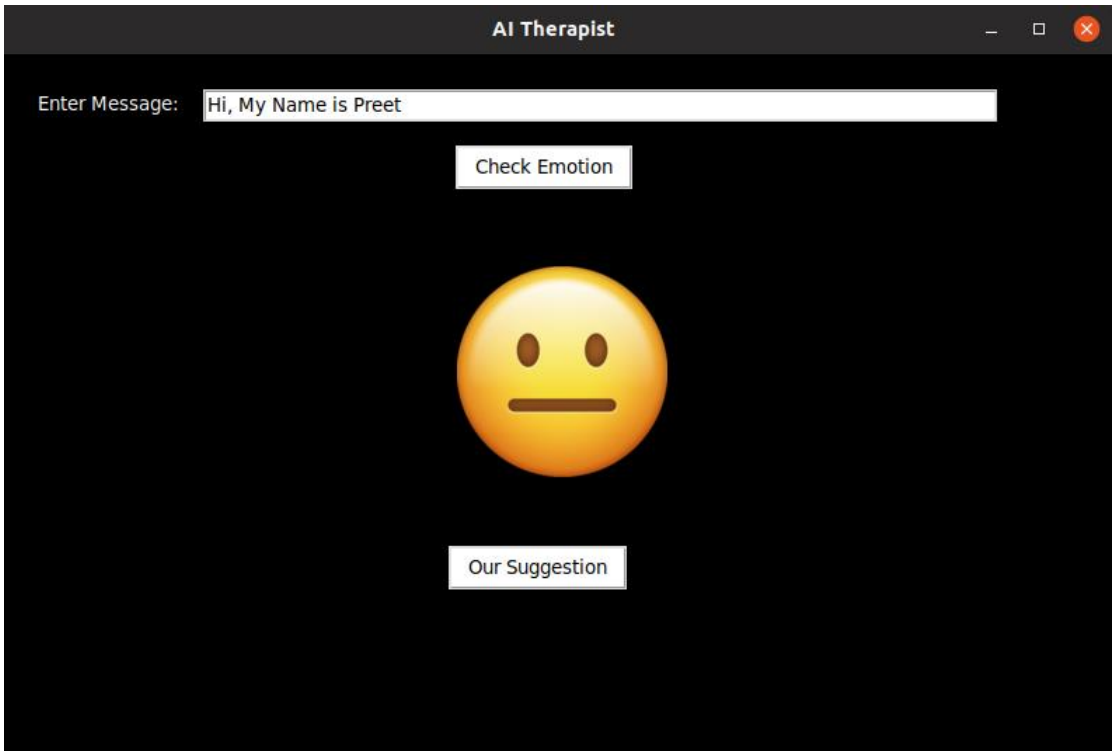
- Automatically the link is generated for YouTube videos based on user's emotion.
- Onclick event will redirect user to YouTube link in the default web-Browser.
- Opening Look of the Application



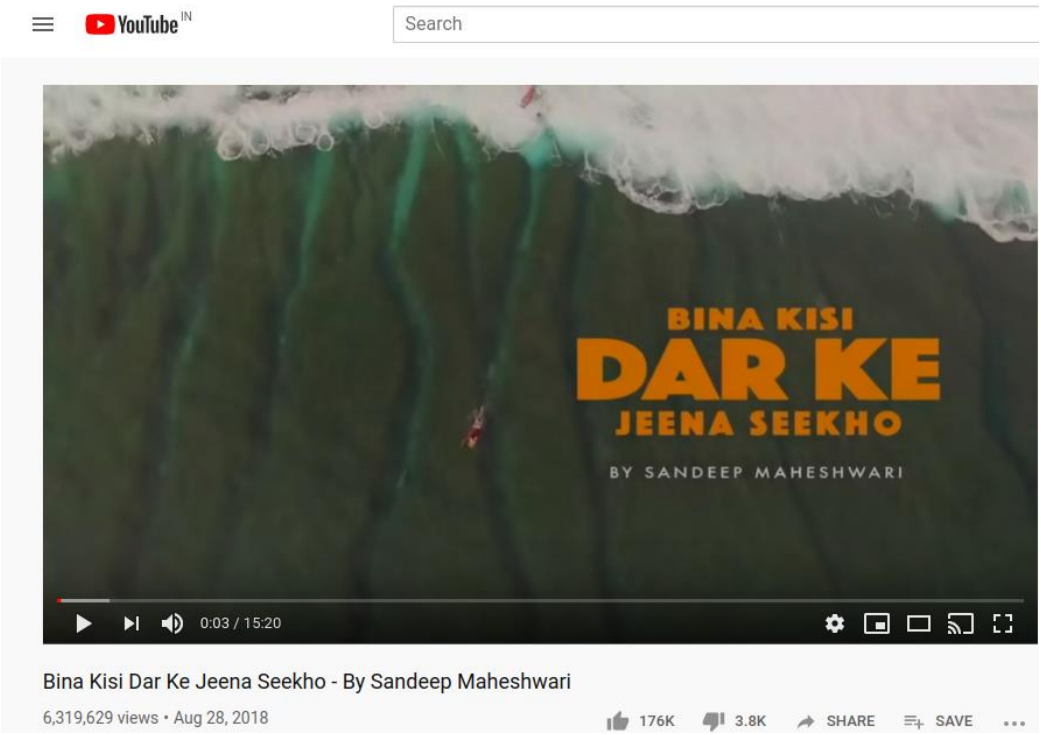
- Enter Text Message



- Then Press the “check Emotion” Button and output will be displayed.



- Press “Our Suggestion” Button and you will directed to YouTube.



- Another Example for Happy Emotion.

9 SOFTWARE TESTING

9.1 Introduction

Software Testing is the process of testing the functionality and correctness of software. Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product to the context in which it is intended to operate .This includes ,but is not limited to ,the process of executing a program or application with the intent of finding errors.

9.1.1 Unit Testing

In this each module is tested individually .Criteria selected for identifying unit test module is to identify module that has core functionality implementation .Module could be an individual or procedure. The following is a list of functions for unit testing that will tested:

- Select the scanned input image of handwritten document.
- Apply Preprocessing.
- Apply Segmentation.
- Apply Feature Extraction.
- Extract Digital character.

9.1.2 Integration testing

Integration testing integrates individual modules and tested as a group. Integration testing takes as its input modules that have been unit tested ,groups them in larger aggregates ,applies tests defined in an integration test plan to those aggregates ,and delivers as its output the integrated system for testing.

9.1.3 Validation testing

The process of evaluating software during or at the end of the development process in to determine whether it satisfies specified requirements.

9.1.4 GUI Testing

GUI testing is the process of testing a product's graphical user interface to ensure it meet its specification, ensuring the navigation between icons/buttons with source code.

10 LIMITATION AND FUTURE ENHANCEMENT

10.1 LIMITATIONS

Like all opinions, sentiment is inherently subjective from person to person, and can even be outright irrational. It's critical to mine a large and relevant sample of data when attempting to measure sentiment. No particular data point is necessarily relevant.

As to sarcasm, like any other type of natural language processing (NLP) analysis, *context matters*. Analyzing natural language data is, in our opinion, the problem of the next 2-3 decades. It's an incredibly difficult issue, and sarcasm and other types of ironic language are inherently problematic for machines to detect when looked at in isolation. It's imperative to have a sufficiently sophisticated and rigorous enough approach that relevant context can be taken into account. For example, that would require knowing that a particular user is generally sarcastic, ironic, or hyperbolic, or having a larger sample of the natural language data that provides clues to determine whether or not a phrase is ironic.

10.2 FUTURE ENHANCEMENT

The system can be further developed for analysis of specific user's tweets and then provide them personal guide for mental care if any discrepancy found.

11 CONCLUSION AND DISCUSSION

11.1 CONCLUSION

Every project work, doesn't matter software project or any other project, could not be the result of sole effort. Sentiment analyses or opinion mining is a field of study that analysis people's sentiments, attitudes, or emotions towards certain entities. This project tackles a fundamental problem of sentiment analysis, sentiment polarity categorization of Twitter data. The field of Machine learning is vast and lot more to innovate in it.

11.2 DISCUSSION

11.2.1 Self-Analysis of Project Viabilities

According to us, this project is absolutely a good start for gaining hands-on experience on projects. It is useful if it is managed according to the goal for which it is made.

11.2.2 Problems Encountered and Possible Solutions

Following are problems encountered during this project.

- Sentiment analysis of Sarcastic, irrelevant tweets
- Limitations of classifiers

11.2.3 Summary of Project Work

We have successfully completed the project. The prior knowledge of software engineering has helped immensely in overcoming the various roadblocks. We have done work with pre-planned scheduling related with time constraints and result oriented progress in project development.

REFERENCES

- https://blog.algorithmia.com/introduction-natural-language-processing-nlp
- http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/
- https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/
- https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
- https://nlp.stanford.edu/projects/glove/
- https://www.researchgate.net/publication/225045375_Emotion_Detection_from_Text
- https://www.researchgate.net/publication/220817578_Emotions_from_Text_Machine_Learning_for_Text-based_Emotion_Prediction