INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Literature Survey and Freliza (Modified)

Independent Study

Project Advisor:
Prof Radhika Mamidi

Submitted By:
Akshat Gahoi
2018114012

Akshat Gahoi (2018114012)

I) User-initiated Sub-dialogues in State-of-the-art Dialogue Systems

This paper tries to test out all the state-of-the-art dialogue systems. They tried to test out user-initiated sub-dialogues.

These dialogues refer to interactions where a system question is responded to with a question or request from the user, who thus initiates a sub-dialogue.

For the experiments, five dialogue systems were compared. These were **Siri, APLAI, Houndify, Cortana, Alexa.** All these dialogue systems differ from each other due to some properties.

The experiment had two parts:-

a) User anytime jumps to a task within the app:
   1) (F1) User anytime jumps to a task within the app. Does the system respond adequately to jumps, i.e. does it shift the topic of conversation to T1?
   2) (F2) System resumes after within-app sub dialogue. After finishing T1, does the system return the dialogue to the previous (unfinished) topic T?
   3) (F3) Signal task resumption. If the system resumes T, does it also indicate this somehow?

   Note that F2 and F3 are not applicable if F1 is answered negatively and that F3 is not applicable if F2 is answered negatively.

b) User anytime jumps across apps :
   4) (F4) User anytime implicit jump across apps. After having asked a question related to a task T in a domain D, does the system respond adequately to a

request or question-related to a task T1 belonging to a domain D1 (but not mentioning D0), i.e. does it shift the topic of conversation to T1 and D1 ?

5) (F5) System resumes after other-app sub dialogue. After finishing T1 , does the system return the dialogue to the previous (unfinished) topic T?

6) • (F6) Signal app (and task) resumption. If the system resumes T, does it also indicate this somehow?

Note that F5 and F6 are not applicable if F4 is answered negatively and that F6 is not applicable if F5 is answered negatively.

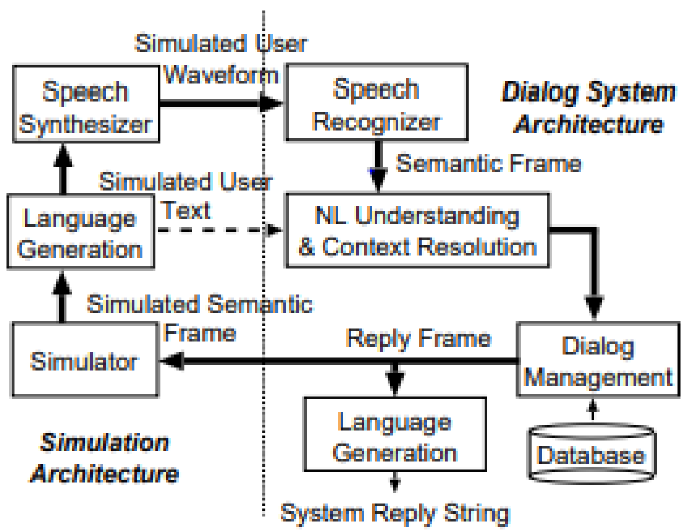
Following are the results of both of these experiments:-

	Siri	API.AI	Houndify	Cortana	Alexa
F1. User anytime jump to task within app	+	+	?	-	+
F2. System resume after within-app sub-dialogue	-	-	?	-	-
F3. Signal task resumption	N/A	N/A	?	-	N/A
F4. User anytime jump across apps	+	-	-	-	(+)
F5. System resume after other-app subdialog	-	N/A	N/A	N/A	-
F6. Signal app (and task) resumption	N/A	N/A	N/A	N/A	N/A

The overall conclusion of the tests is that none of the systems tested deal appropriately with user-initiated sub-dialogues. In light of how frequent this behavior is in human-human dialogue, this is a serious shortcoming.

II) Developing A Flexible Spoken Dialog System Using Simulation

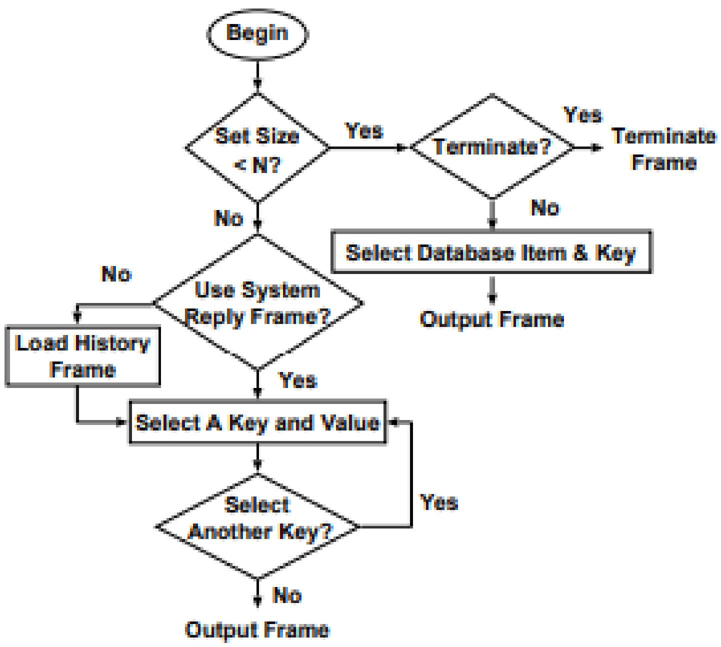
As the name suggests this paper focuses on building a better dialog system by extensive use of simulations that help to produce thousands of unique dialogs which benefit not only dialog development but also provide data to train the speech recognizer and understanding components, in preparation for real user interaction.



The figure shows us that the architecture of the dialog system that is talked about in the paper. The input is taken in the form of speech or text mode. There are different modules for the same. The dialog management system is the module where the decision-making takes place which is then provided as an output by the language generation module, This is passed on to the simulator which helps us to take this conversation forward.

We can see from the figure given below the dialog is continued until the options are reduced to less than a given threshold N (in this case no. of hotels). The dialog asks about the constraints in different ways so that it is able to reduce the options for the user and try to get exactly what a user wants.

Here dialog management plays its part by considering constraints such as geographical location, price, and type of restaurants. It will try to converge the number of possibilities that a user has through these constraints.



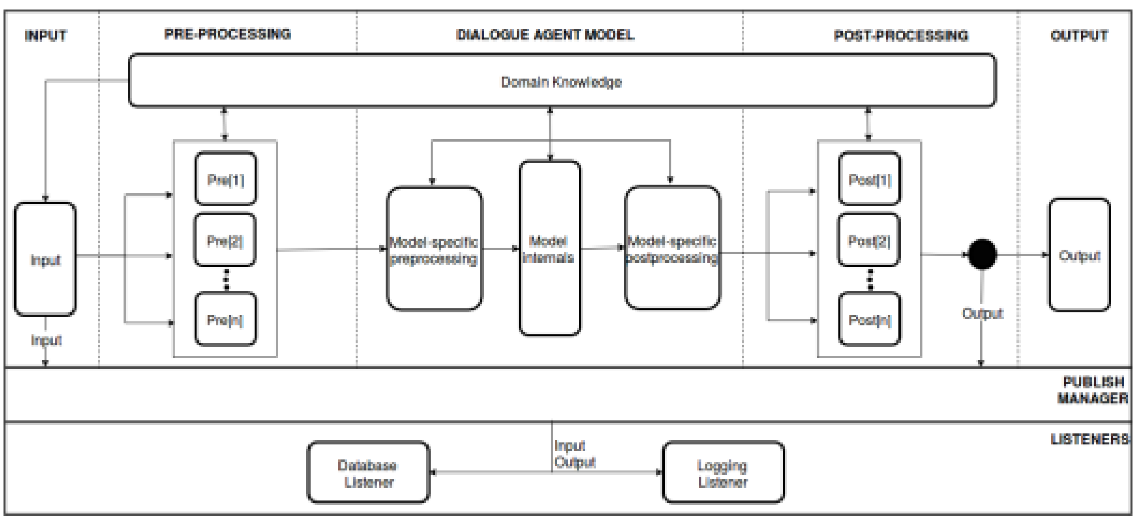
Three types of experiments were conducted for this system. Twi were the text mode and speech mode where the different forms of input made a difference as for speech mode, speech recognition becomes an essential part of the project. The third one was regarding over-constrained queries where more than one constraint was given in a single query. Initially, empty datasets were retrieved by the system but after some training, the results were better.

The hope is that using simulation runs will improve system performance to a level such that the first collection of real user data will contain a reasonable rate of task success, ultimately providing a more useful training corpus.

III) MACA: A Modular Architecture for Conversational Agents

This paper tries to propose a modular architecture for dialogue systems. It tries to help the user to implement quick prototyping. The modular architecture will help the developer to change different things according to their use (domains etc.) easily. MACA provides tools to host dialogue agents on Amazon Mechanical Turk (mTurk) for data collection and allows the processing of other sources of training data. The current version of the framework already incorporates several domains and existing dialogue strategies from the recent literature.

	MACA	TCP	Ravenclaw
Multi Domain Support	✓	✓	✓
Plug-and-Play	✓	✓	✗
Adaptation for FCA	✓	✗	✗
Agent Abstraction	✓	✗	✗
Integration with mTurk	✓	✗	✗

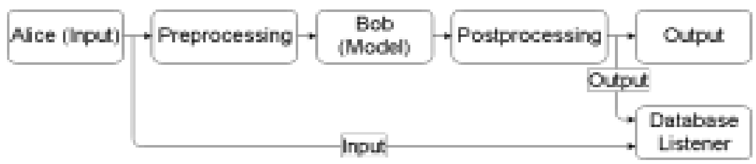


Components are divided into- **Domain Knowledge, Input, Preprocessing, Dialogue Model, Postprocessing, Output, Pubsub system/Listeners.**

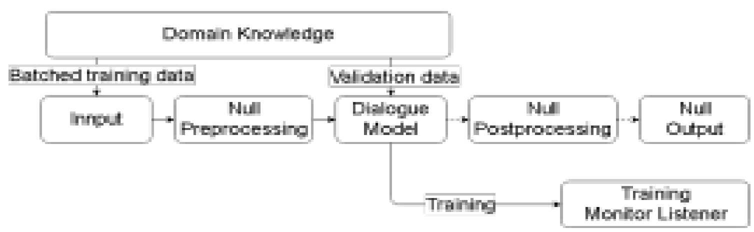
These components function differently according to the use of the developer. This also includes Listeners which keeps a log of each and every data flow that is happening inside this architecture.

The architecture work differently for:-

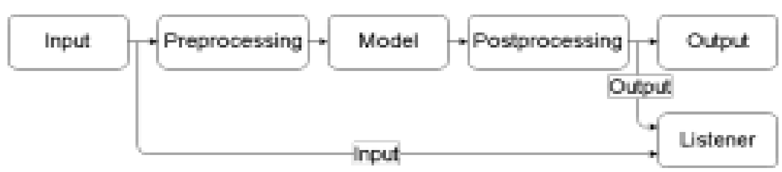
a) Data Collection Mode



b) Training Mode



c) Execution Mode



Component		Description	Note
Domain Knowledge		GoalOriented DomainKnowledge	Specifying slots information for known domains.
Input		StdInputDevice	Inputs from stdin.
Preprocessing		VoidPreprocessor	None.
Model	Preprocessing	VoidProcessing	None.
	Postprocessing	Model specific	None.
	Internal	PersonalInformation AskingModel	Intent disambiguation and execution policies.
Postprocessing		VoidProcessing	None.
Output		FileOutputDevice	Output to a file.
Listeners		LoggingListener	Log all pubsub notifications to file.

An example of a goal-oriented system in execution mode is given in the paper which helps us to understand the function of all these modules.

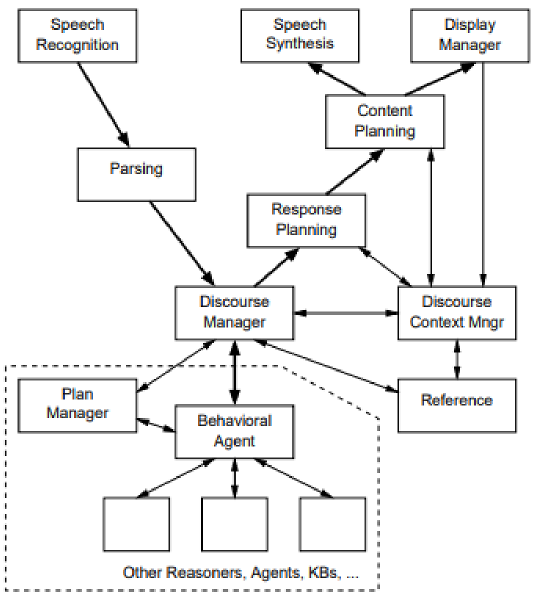
IV) An Architecture for a Generic Dialogue Shell

This paper describes the functioning of a generic dialogue shell. The paper starts by describing what is dialog shell is and its importance in the modern world. The goal of the dialog shell described by this paper is to produce machines that can mimic human conversation. The paper is built on two hypotheses:-

a) The Practical Dialogue Hypothesis: The conversational competence required for practical dialogues, while still complex, is significantly simpler to achieve than general human conversational competence.
b) The Domain-independence Hypothesis: Within the genre of practical dialogue, the bulk of the complexity in the language interpretation and dialogue management is independent of the task being performed.

The paper talks about the initial dialog systems:

Domain	Date	Task	Goal	Status
TRAINS	1995-7	Finding efficient routes for trains	Robust performance on a very simple task	Robust performance (> 90% success rate)
PACIFICA	1997-8	Evacuating people from an island	Robust performance in a task requiring explicit planning	Demonstration system supports untrained users
CPoF	1998	Deployment of troops in a military situation	Scripted demonstration in a military relevant task	Scripted interaction only
Monroe	1999–	Coordinating responses to emergencies in Monroe County, NY	Robust performance on a dynamic, mixed-initiative task involving planning, monitoring and replanning; larger domain	In development for robustness evaluation
AMC	1999–	Planning airlifts using an airlift planning system	Demonstrate ability to use third-party planning systems; emphasis on agent technology	Initial demonstration completed, work on extensions continuing
Kitchen	Planned	Planning kitchen design	Robust performance on a significantly different task; multilingual experiments	Planned for development



Issues that are discussed in the paper are:

a) **Speech Recognition** (Issues regarding language model/grammar i.e. the very first step of a dialogue shell)

b) **Parsing** (Different word forms can take different meanings and functions in a sentence according to a domain)

c) **Reference Resolution** (All the semantic relations of a word in a sentence)

d) **Content planning and generation** (How the results are generated and in what order the system is taking on the conversation to give user the best output possible)

Module	Function
Speech Recognition (SR)	Transforming speech input into a word stream or word lattice
Parser	Transforming the SR output into interpretations, each a set of conventional speech acts, using full and robust parsing techniques
Reference Manager (REF)	Identifying the most salient referents for referring expressions such as noun phrases
Discourse Context Manager	Maintaining the global (topic flow) and local (salience with a topic) discourse context
Discourse Manager (DM)	Identifying the intended speech act, current task, current step in the current task, and system obligations arising from the dialogue
Behavioral Agent (BA)	Determines system actions (*e.g.*, answer a question, notify of a problem, request clarification); Manages the interface to the back-end systems.
Plan Manager	Constructing, modifying, evaluating, and executing plans (whether they are the subject of the conversation or the task being executed)
World KB	Maintains a description of the current state of the world under differing assumptions (*e.g.*, based on different plans or hypotheses)
Response Planner	Determining the best communicative act(s) (and their content) to accomplish the system's current goals and discourse obligations
Content Planner	Determining how to realize the planned speech acts
Display Manager	Managing the visual presentations given the available displays

Freliza (Modified V2)

Background and Model

The idea behind making Freliza V2 was to use the existing tools to detect depression. When we talk about depression we generally are not able to understand the issues behind it.

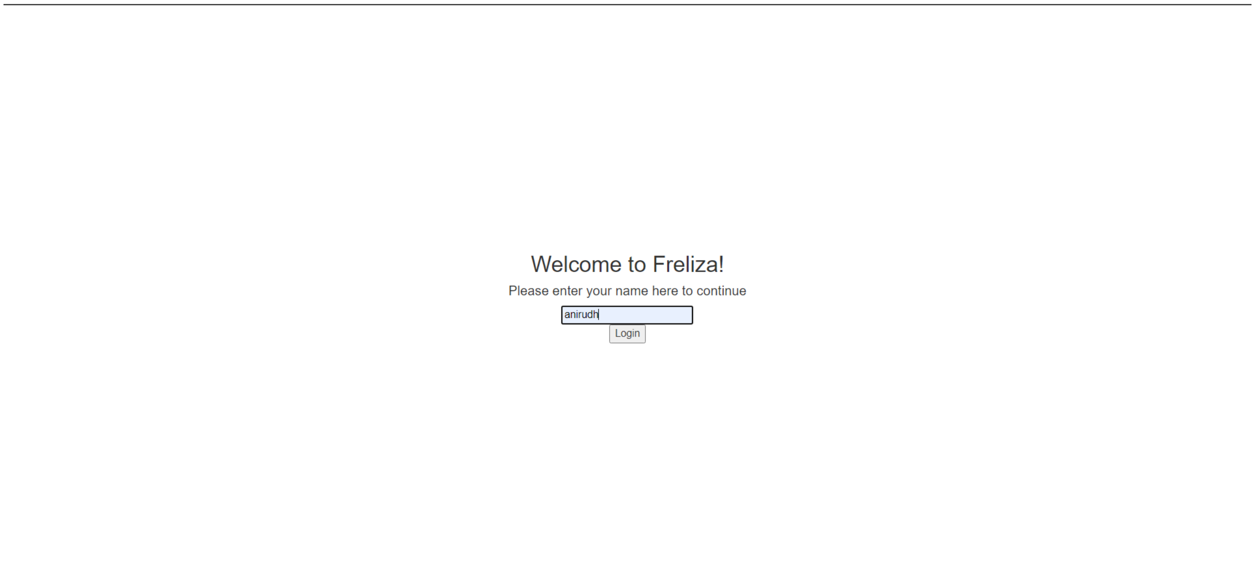
We saw that even professional therapists require at least 3 therapy sessions to detect depression. So we thought of using Freliza (created by IIIT students) to classify chats into different emotions (Anger, sad, happy, fear, neutral, etc.). For this classification we used BERT and fine tuned it on the Isear dataset (7.5k sentences).

All the comments made by a user are classified into a negative emotion (like sadness and anger) and a positive emotion to give a score of -1 and +1 respectively. These scores are averaged over the length of the chat to gain information about an individual's mental state at the time of chat.

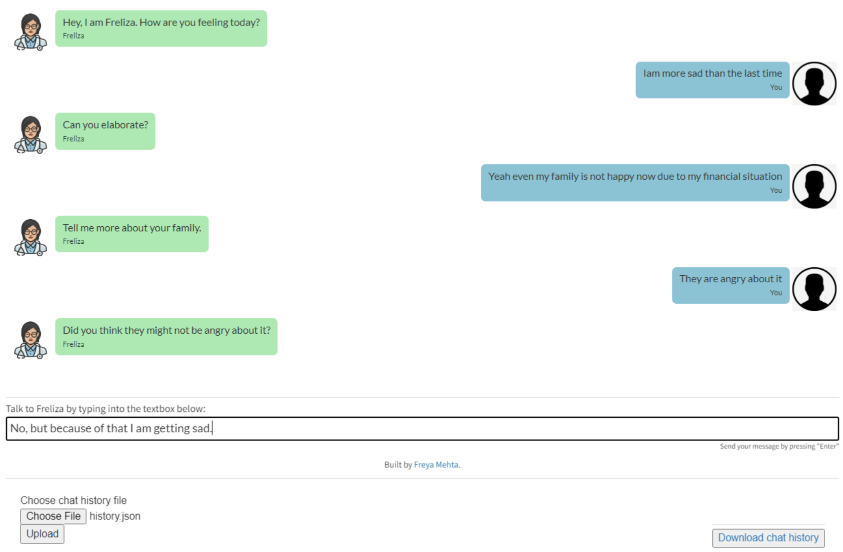
After three chats (can be increased in the code) with Freliza, we generate scores of each chat which can be seen by a therapist/individual. If these scores consistently show negative scores we can say how a person is constantly in the state of sadness and can be in depression (or will go into depression in future).

Demo

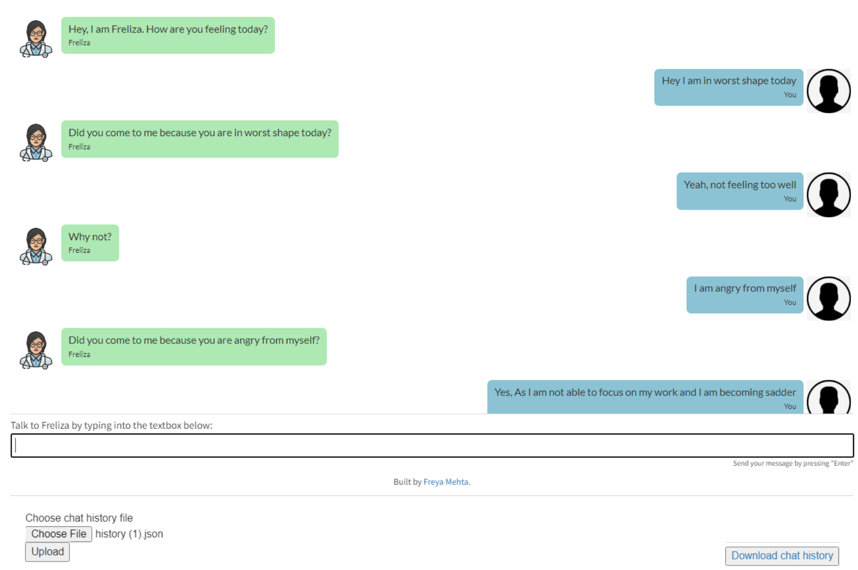
1) Open index.html and enter the name of the patient.



2) Use the chatbot and download the history so that we can store the chats (Can be used by therapists and classifiers for further study).



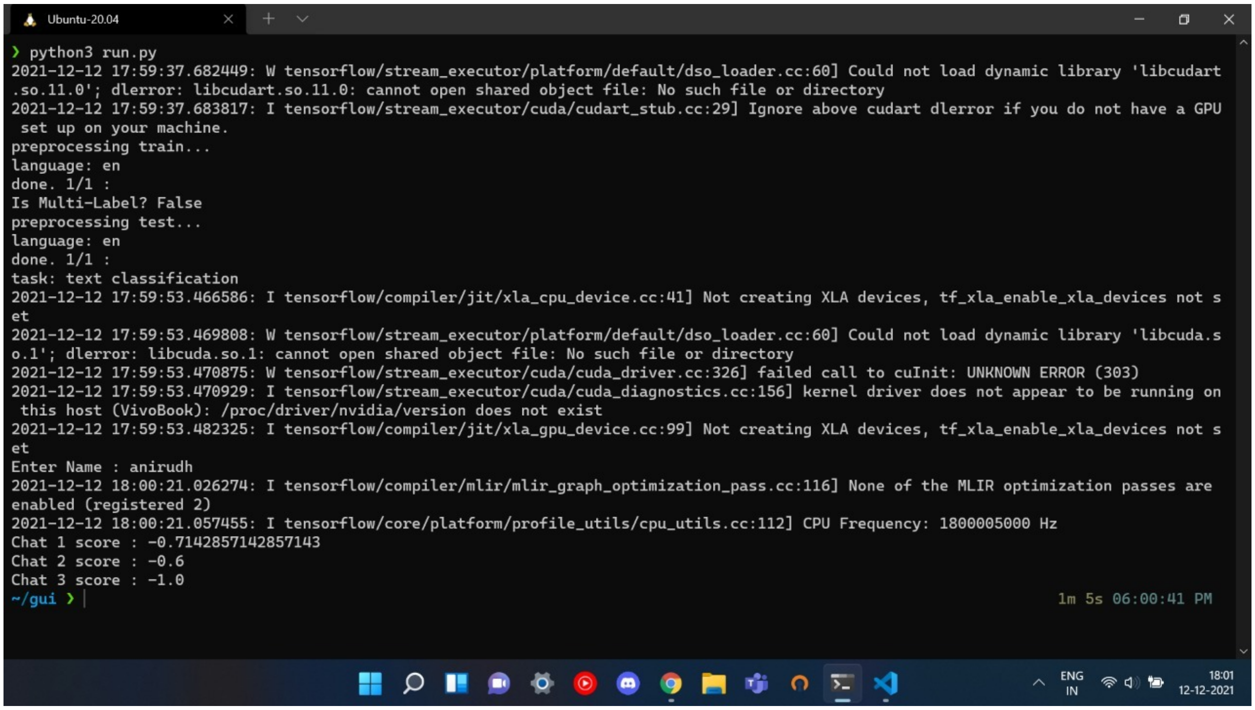
3) When a patient logs in again they have to upload their history file for the chatbot first then they can start chatting.



4) After a patient has completed 3 chats with Freliza a result screen will be shown which will show a user how to retrieve results from chatbot

Execute run.py with history file(history.json) { python3 run.py } and enter the username of patient.

5) After putting in the username the chatwise scores will be shown on the terminal.



Future Steps

As one can see we have divided the task in collecting chat data over a period of a time and classifying it on each reply. We can try to improve both of the parts individually i.e. the reply of Freliza and how good our classifier is.

We can also experiment with the scoring i.e. what negative scores should be given to sadness, anger, fear etc. and what should be the difference in scoring between them but this should be done only after taking advice from a therapist. This is a very important step as all the emotions play differently when we are dealing with depression.