

-----🙞🙜🕮🙞🙜-----

BASIC PROJECT

CHATBOT

Team members:

Student’s name: Phùng Tuyền Hào

Student’s ID: 1711061809

Student’s name: Nguyễn Tấn Thành

Student’s ID: 1711066230

Supervisor: Lê Thị Ngọc Thơ

Contents

[I. INTRODUCTION: 4](#_Toc51144034)

[1. Context 4](#_Toc51144035)

[2. Definition: 5](#_Toc51144036)

[3. Reason 6](#_Toc51144037)

[4. Problem 7](#_Toc51144038)

[5. Scope of this project 7](#_Toc51144039)

[II. LITERATURE REVIEW 8](#_Toc51144040)

[1. Background 8](#_Toc51144041)

[III. Analysis and design 18](#_Toc51144042)

[1. Analysis 18](#_Toc51144043)

[1.1. NLTK & NLP 18](#_Toc51144044)

[1.2. NumPy 18](#_Toc51144045)

[1.3 Keras 18](#_Toc51144046)

[1.4 Data files 19](#_Toc51144047)

[2. Design 19](#_Toc51144048)

[3. Implementation 21](#_Toc51144049)

[3.1. Initializing Chatbot Training 21](#_Toc51144050)

[3.2. Building the Deep Learning Model 23](#_Toc51144051)

[3.3. Build simple GUI chatbot 24](#_Toc51144052)

[IV. Evaluation 29](#_Toc51144053)

[V. Conclusion 29](#_Toc51144054)

[1. Overall Result 29](#_Toc51144055)

[1.1. Achievements 29](#_Toc51144056)

[1.2. Limitations 30](#_Toc51144057)

[2. Future improvements 30](#_Toc51144058)

[VI. Reference: 31](#_Toc51144059)

[Figure 1: Number of active users in social networks 4](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156738)

[Figure 2: Chatbot Implementation Plans in organizations 4](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156739)

[Figure 3: ELIZA (by Joseph Weizenbaum)–The world’s first chatbot 8](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156740)

[Figure 5: PARRY - Kenneth Colby 9](#_Toc51156741)

[Figure 4: RACTER - William Chamberlain and Thomas Etter 9](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156742)

[Figure 6 JABBERWACKY - Rollo Carpenter which is discontinued in October 2008 10](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156743)

[Figure 7: ALICE – Richard Wallace 10](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156744)

[Figure 8 : JABBERWACKY’s successor: CLEVERBOT 11](#_Toc51156745)

[Figure 9: Classification and Regression result 15](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156746)

[Figure 10: How supervised learning machine algorithm works 15](#_Toc51156747)

[Figure 11: 3 different stages of ditting stage 16](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156748)

[Figure 12: Operating comparison between rule-based (FSM) and AI-based (Deep learning) chatbot 17](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156749)

[Figure 13 Sequence diagram 19](#_Toc51156750)

[Figure 14: Use case diagram 20](#_Toc51156751)

[Figure 15: Initializing training 21](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156752)

[Figure 16: JSON file format 22](#_Toc51156753)

[Figure 17: Tokenize the intent 22](#_Toc51156754)

[Figure 18: Tokenize the intent 23](#_Toc51156755)

[Figure 19: Building the Training model 23](#_Toc51156756)

[Figure 20: Initialize the Tranining 24](#_Toc51156757)

[Figure 21: Import infomation 24](file:///C:\Users\MSI\Destop\Do-an-co-so-Chatbot.docx#_Toc51156758)

[Figure 22: Processing for running the GUI 25](#_Toc51156759)

[Figure 23: Build simple GUI chatbot 27](#_Toc51156760)

[Figure 24: Simple interface 28](#_Toc51156761)

[Figure 25: Simple conversation 28](#_Toc51156762)

# I. INTRODUCTION:

## **1. Context**

“Digitalization, the surge of mobile and internet-connected devices has revolutionised the way people interact with one another and communicate with businesses”

(Milan van Eeuwen, 2017)

The new generation is supporting new technologies has now become a part of their daily life, this starts becoming more accepted as tech companies are mainstreaming AI into the products they offer, like Google Assistant, Amazon Alexa.

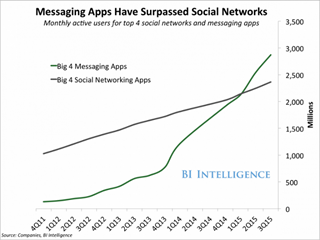
Although chatbots have existed for years without heading the mainstream, there are several times that the chatbot hits trends which highlighted in 2017:

Figure 1: Number of active users in social networks

* Mobile messenger domination.
* Support for chatbots by Facebook, Microsoft and other big tech companies.
* Dramatic reduction in chatbot development costs.

The gradual trend is constantly changing in a positive direction through several surveys like SpiceWork as more big companies gradually follow the trend of chatbot rather than developing the phone industry with Microsoft Cortana is the most commonly used intelligent assistant in the workplace and Apple Siri comes second.

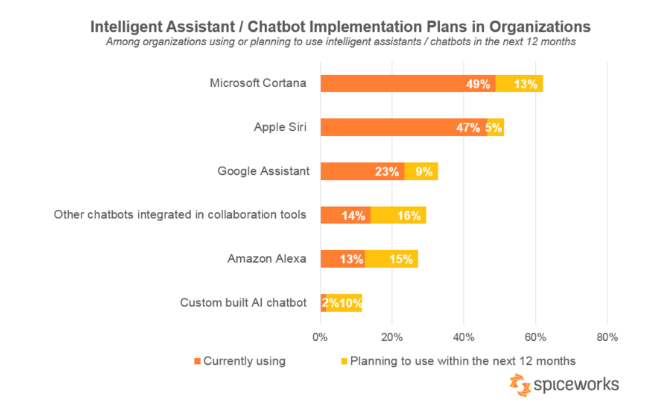


Figure 2: Chatbot Implementation Plans in organizations

These above topics simply state that chatbots are become popular, and used on many different platforms.

## **2. Definition:**

Then what is a chatbot?

A chatbot is a computer program that can stimulate conversation with users through its natural language. There are a variety of methods you can build a chatbot depending on whether it can satisfy your goals. There are 3 types of chatbots most of us already seen:

* Rule-based chatbots:

Following pre-designed rules, often build a graphical user interface where designers will design paths using a decision tree.

* Live chatbots:

Mainly used in a Customer Support organization is to answer questions in real-time.

* AI-based chatbots:

They will automatically learn after initial training by a bot developer.

## **3. Reason**

As for the business side:

According to BI Intelligence, people are using messenger apps more than they are using social networks, which is:

“People are now spending more time in messaging apps than in social media and that is a huge turning point. Messaging apps are the platforms of the future and bots will be how their users access all sorts of services.”

Peter Rojas, Entrepreneur in Residence at Betaworks

Major shifts on large platforms prove that messaging apps are the new marketing channel. Chatbots can provide benefits that traditional methods can not which are:

* Improving Customer Service:

Clients may need help to understand which step they need to take next when buying a product, and chatbots can provide real-time help, speeding up the path to purchase. Plus, chatbots can gather information from consumer feedback and use that data to optimize the algorithm to how you can improve your product and service.

* Cost savings:

Implementing an intelligent chatbot is much cheaper and faster than hiring a new team of employees, and allows to handle interact many numbers of customer’s interaction at once. The accuracy of chatbots can reduce human errors.

* Automate repetitive tasks:

Chatbots can take over some of the repetitive tasks with the call centre, which employees don’t have to. Or using them to on hold while trying to connect you to the call centre executive, they will ensure that staff can spend more of their time on crucial tasks.

As for personal use:

* A starting point for beginners in AI majors:

This is a good start for those who want to learn AI because making chatbots are easy to pick up and follow.

* Helping users with easier life management:

There are a lot of benefits that can help your daily life easier, such as scheduling a meeting, finding interesting news, managing money,…

* Helping users who require mental support:

A chatbot can provide support to people with mentally struggling, especially students. According to collegestats.org, students will probably experience some mild depression at the very least some time during their college years, which can be normal. But sometimes, students can experience a severe one and need bits of help. The chatbot can act as a counsellor and confidant to records questions and find answers that match the resources students seek for. And using this data to access information and resources, which regarding sensitive topics to staffs, which can create or improve the meaningful resources to help students.

## **4. Problem**

There is a problem we have encountered:

There are some developers, including students whose major AI found intimidated and confused to develop the chatbot due to overloading amount of stuff that they struggle to understand.

“To do this, someone has to know how to build a chatbot which can confuse for non-tech-savvy people. Even companies who have in-house developers might not have someone skilled at creating a chatbot on their own.”–Rebecca Reynoso

## **5. Scope of this project**

Getting started with a chatbot can feel a bit intimidating, which is why our aim is to find a fast and convenient way to build a machine learning chatbot without the difficulty. And understanding a human-machine interaction model through an automated chatbots system.

The chatbot we build have features which are:

Training:

The chatbot is pre-trained to understand specific knowledge and terms.

Natural Conversation:

It can also identify the intent of a question–we need what–to provide an accurate first response and also propose options to confirm or clarify intent.

Customizable intents:

Users can customize the intent of chatbot to improve and match their experience.

# II. LITERATURE REVIEW

## **1. Background**

History of chatbot

Figure 3: ELIZA (by Joseph Weizenbaum)–The world’s first chatbot

Developed by Joseph Weizenbaum, a German-American computer scientist and a professor at MIT university, ELIZA considered being the very first chatbot in the history of computer science, with the function to trick users into believing that they were having a conversation with a real human.

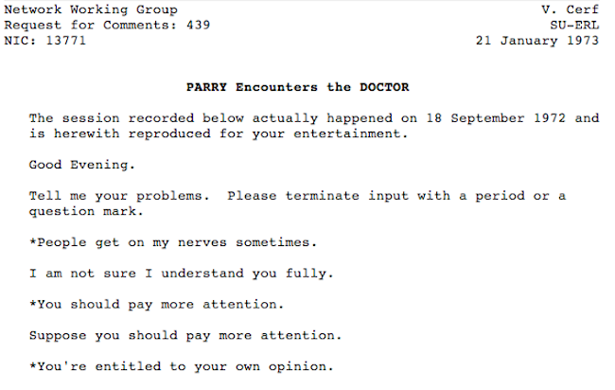
It mainly designed to act like a therapist who would ask an open-ended question and, surprisingly enough, can make a response with follow-ups.

However, Weizenbaum himself did not claim that ELIZA was genuinely intelligent, and the introduction to his paper presented it more than a debunking exercise:

“[In] artificial intelligence ... machines are made to behave in wondrous ways, often sufficient to dazzle even the most experienced observer. But once a particular program is unmasked, once its inner workings explained ... its magic crumbles away; it stands revealed as a mere collection of procedures ... The observer says to himself "I could have written that". With that thought, he moves the program in question from the shelf marked "intelligent", to that reserved for curios ... The object of this paper is to cause just such a re-evaluation of the program about to be "explained". Few programs ever needed it more.”

[Weizenbaum 1966]

After ELIZA, they made other successful chatbots, namely–PARRY  in 1972, RACTER in 1983, ALICE in 1995, then JABBERWACKY with the later successor, CLEVERBOT in 2008.



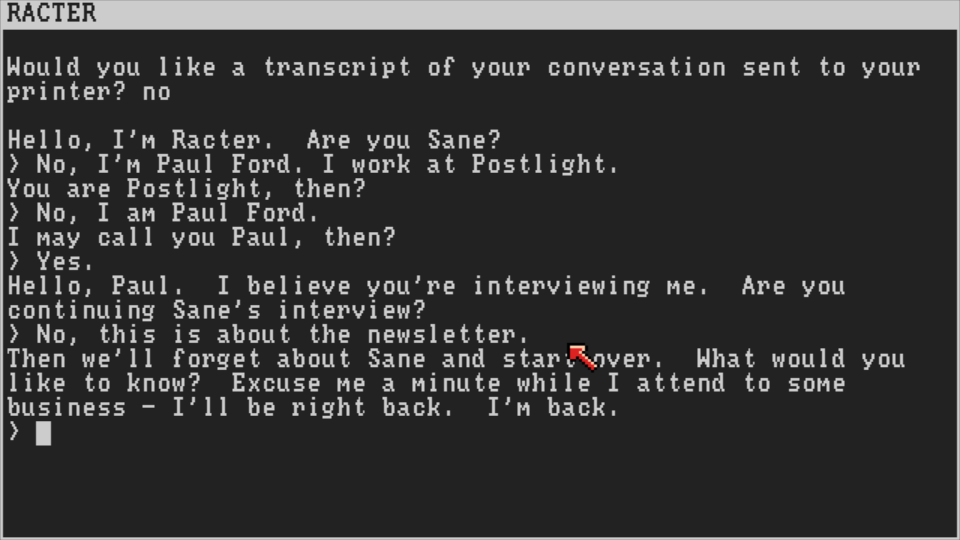
Figure 5: PARRY - Kenneth Colby

Figure 4: RACTER - William Chamberlain and Thomas Etter



Figure 6 JABBERWACKY - Rollo Carpenter which is discontinued in October 2008

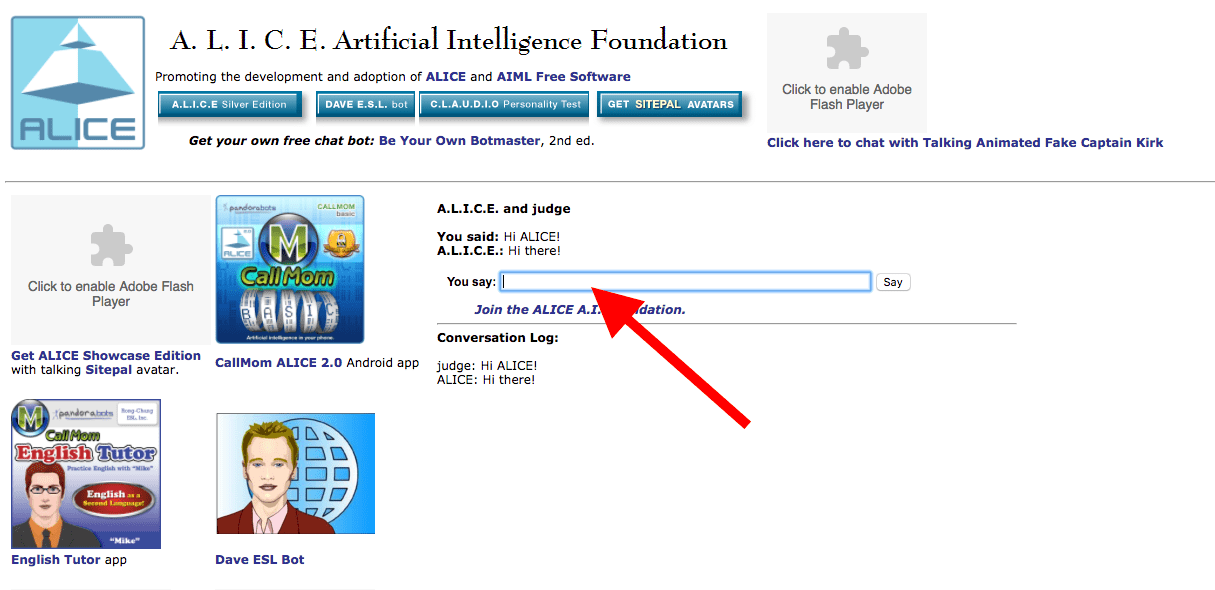


Figure 7: ALICE – Richard Wallace



Figure 8 : JABBERWACKY’s successor: CLEVERBOT

## **2. Definition of Machine Learning**

Definition of Machine Learning

Machine learning considered being a backbone of the chatbot which is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence.

It closely relates machine learning to computational statistics, which focuses on making predictions using computers.

Machine learning algorithms build a mathematical model based on sample data, known as "training data", to make predictions or decisions without being explicitly programmed to do so.

Programming Language:

### **2.1. Python**

- Definition

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1990. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting. In 2020, Python 2.x officially discontinued. Only recommended for users to learn it when:

Some certain third-party software and libraries that can’t be ported to Python 3.

If companies have legacy code written in Python 2.

Python 3.0, released in 2008, was a major revision of the language that is not completely backwards-compatible. I highly recommend it for beginners.

+ One of the biggest advantage when using Python in building a chatbot is Python is a versatile language with very simple syntax. Some changes in Python 3 have made beginners easy to read and understand.

+ Python also comes equipped with extensive libraries (include Natural language processing (NLP) toolkit, which is a key element of intelligent chatbots written by Python) to accomplish a variety of AI projects.

+ So with AI programming languages, Python is the best candidate with its dedicated community support and pre-built libraries that help AI development significantly.

+ The only limitation in using Python to build the chatbot is its slowly executing. The line-by-line execution of code often leads to slow execution. The dynamic nature of Python is also responsible for the slow speed of Python because it has to do the extra work while executing code.

+ Also Python requires an extensive amount of memory.

For this, Python will be the primary language for our project. Some alternatives can replace Python are:

### **2.2. Java**

- Definition

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform.

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

A chatbot is also a suitable choice for building chatbots for several reasons:

* As an object-oriented programming language, the portability of Java makes it ideal for Chatbot development.
* With a lean and portable size, Java also offers an in-built garbage collection facility.
* ALICE, one of the leading chatbots inspired and strongly developed by ELIZA, is written in Java.

The main limitation in building chatbot using Java is its slow execution speed. Also, it requires changes when running on different platforms.

### **2.3. Ruby**

- Definition

Ruby is an interpreted, high-level, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan.

Ruby is designed to have a very simple syntax which allows beginners easily create a chatbot

Like Python, Ruby is a dynamic and object-oriented programming language. In this language, everything is an expression and everything is executed imperatively.

But unlike other programming languages, creating a chatbot with Ruby might be expensive. It is initially available for free, but a licence might be required at some point. It is also not very flexible for developers since it’s not independent like Java or PHP.

## **3. Method**

Finite State Machines:

During our research, we have tried to implement a previous method called FSM, is a computational model that can represent and control execution flow. It is built up of one or more states and can only be in one active state at any given time.

These are state machines that involve complex pattern matching and are about matching incoming strings, which is an input text with pre-defined patterns. The pre-defined pattern matching uses a regular expression to find patterns in the incoming text and responds it back with fixed-designed texts.

FSMs mainly designed to solve the static problems in fields like mathematics, games, linguistics, and artificial intelligence.

For example, traffic lights are an example of the FSM model that uses three colours green, amber and red as three different states. And it will activate one state after another according to the pre-designed time limit.

It mainly uses FSMs on AI as rule-based chatbots in many companies because of flexible implementation and low cost.

FSMs remain widely used in different industries. However, we can not recommend this method for its limitations can not be overlooked::

The major limitation of pre-designed chatbots that keeps persists is mainly designed for repetitive work from small to complex tasks. When the user inputs string texts outside of its pre-designed database, the chatbot will be immediately unresponsive and then require human intervention to that conversation. Therefore, rule-based bots cannot operate standalone.

Building FSM to a chatbot that itself is a complex system, hard for managing with no idea of design.

Deep Learning:

Deep learning is a subset of machine learning that is more sophisticated methods than FSM. The process, input, and understand the intention in a broader context. They use various types of neural networks for this purpose. This way they are more capable and effective in managing dialogue with humans than FSM are. The chatbot applications within deep learning can handle a broader context that can cover multi-domain or multi-service business context.

There are 3 types of learning is supervised, semi-supervised, unsupervised learning. We choose supervised learning to apply for our project.

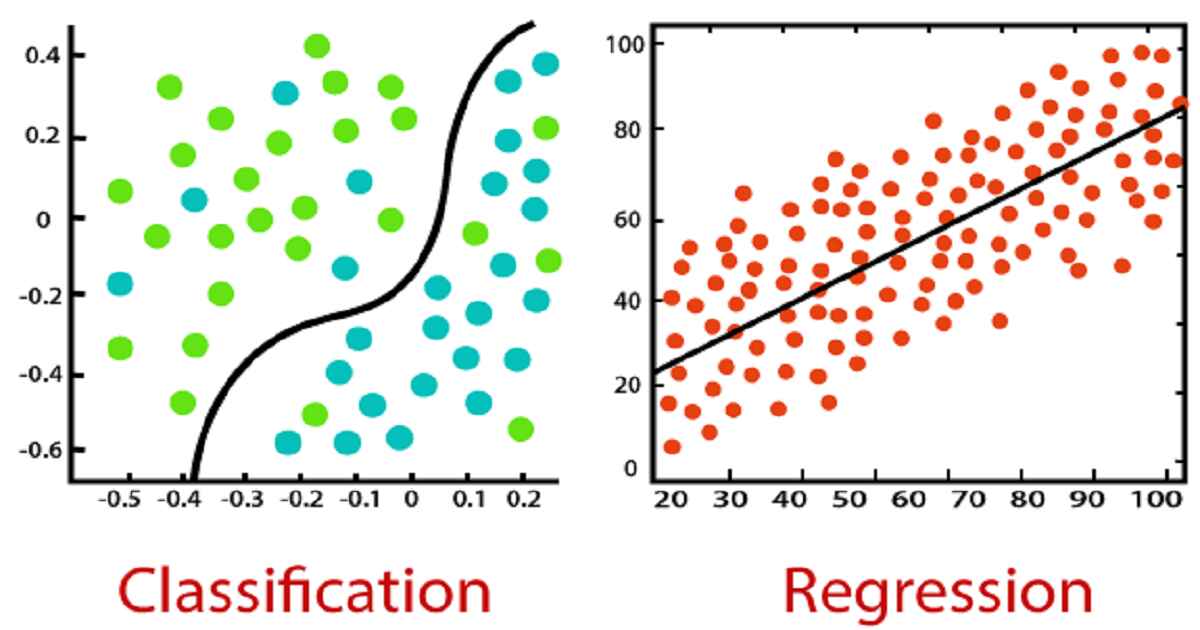
There are two common usages is Classification and Regression. To simplify the definition, Classification is simply to categorize every type of data, while Regression means to find a way to describe the data.

Figure 9: Classification and Regression result

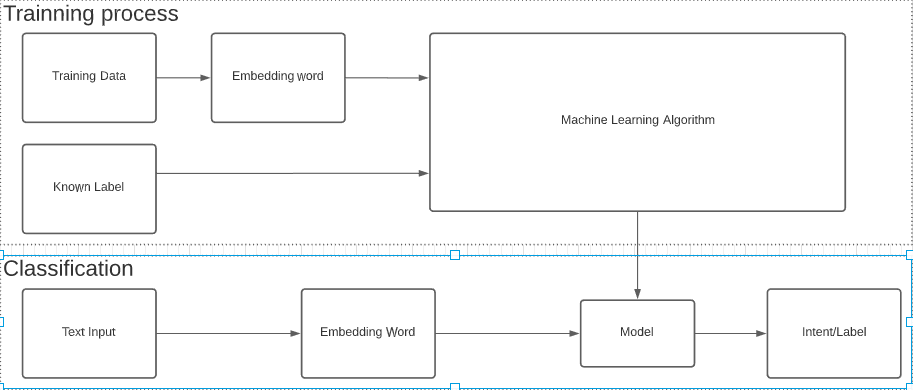


Figure 10: How supervised learning machine algorithm works

In the ML program, there are also 2 stages: fitting and predicting stage.

Fitting stage:

The program will receive many data to “adjust” its parameter through some statistical model to “fit” the input data best.

Figure 11: 3 different stages of ditting stage

Under-fitting refers machine learning model is too simple, less feature and regularized too much informed which will not be a suitable model and will have a poor performance on the training data.

Over-fitting refers to a model that models the training data too well or over-complicated that cannot fit additional data or predict observations reliably. This is by far the most common problem in applying machine learning.

To reach a good fit/robust which is very difficult to do we must look at performing a training data over time to either filled or held back additional data to the training model.

**Predicting stage:**

The program will give out a prediction for a new input based on the parameters it just calculated out. Classification and Regression algorithm is the best example of using prediction technique.

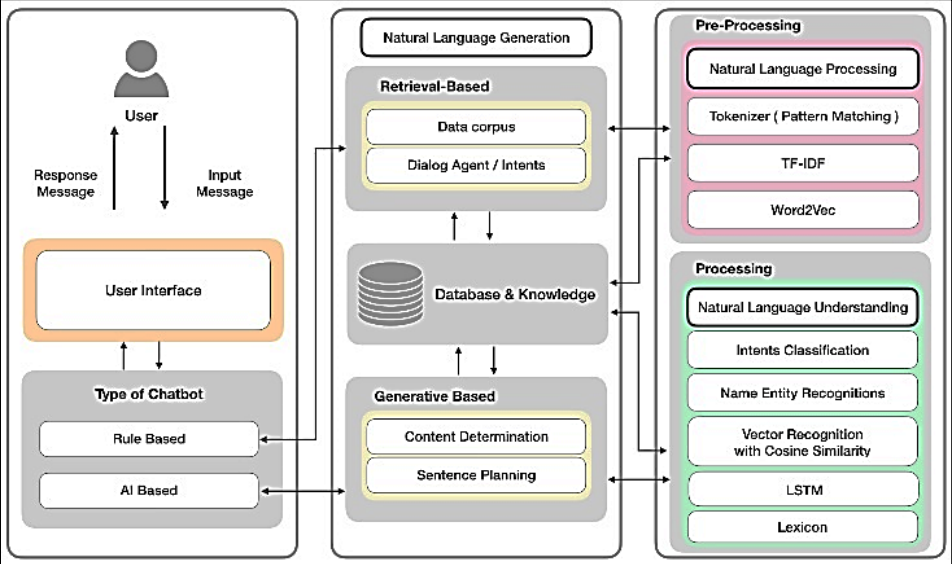


Figure 12: Operating comparison between rule-based (FSM) and AI-based (Deep learning) chatbot

# III. Analysis and design

## **1. Analysis**

Libraries & Data

All of the necessary components to run this project required on Python. Here’s a quick breakdown of the components:

### **1.1. NLTK & NLP**

NLTK is a platform used for build programs for text analysis for English written in the Python programming language.

We highly recommended it for beginners to want to build linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning.

I will briefly explain about NLP:

Natural language processing is the ability of a computer program to understand human language as it is spoken.

NLP will play an important role for building chatbots. Without NLP, chatbots will cannot differentiate between certain phrases like “Hi” and “Hello”.

NLTK is essential for building a chatbot and there are suitable alternatives: spaCY, TextBlob, etc.

### **1.2. NumPy**

NumPy is a library for Python programming language, is designed for numerical computing which Python is not originally designed for. It supports for large, multi-dimensional arrays and matrices, along with an extensive collection of high-level mathematical functions to operate on these arrays. It is the best alternative to Python’s syntax for using less store data while maintaining high speed and performance.

### **1.3 Keras**

Keras is an open-source neural network library written in Python and supports multiple back-end neural network computation engines. For this, Keras is the deep learning framework we’ll be using.

### **1.4 Data files**

There are two data files we’re implying are JSON, a syntax for storing and exchanging data and pickle similar to JSON file but for pickle file.

## **2. Design**

Sequence diagram

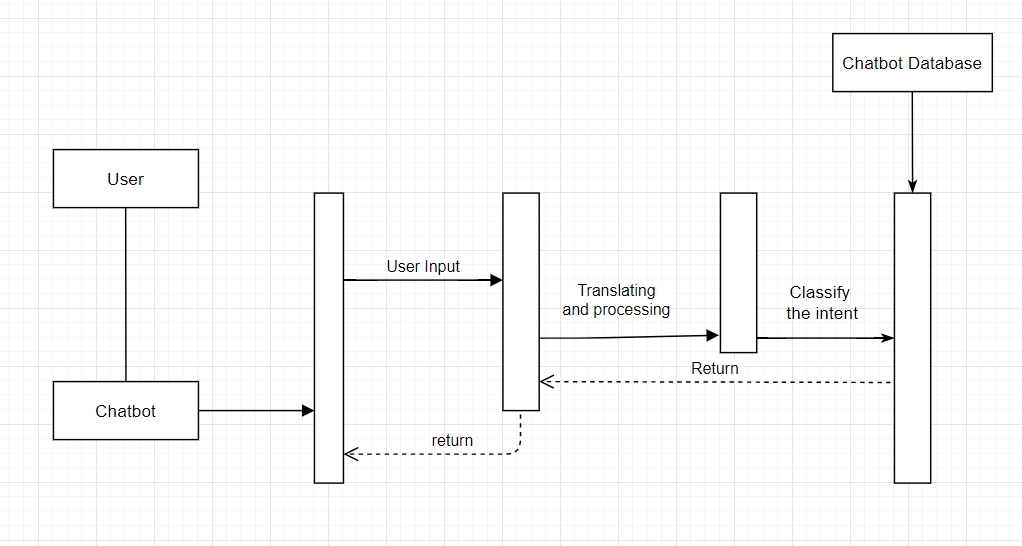


Figure 13 Sequence diagram

The user accesses to the chatbot, inputs a string of characters, and the system analyzes the input and processing. The system classifies and calculates it with intents to match the message. After that, it gives a response and translate it to the user.

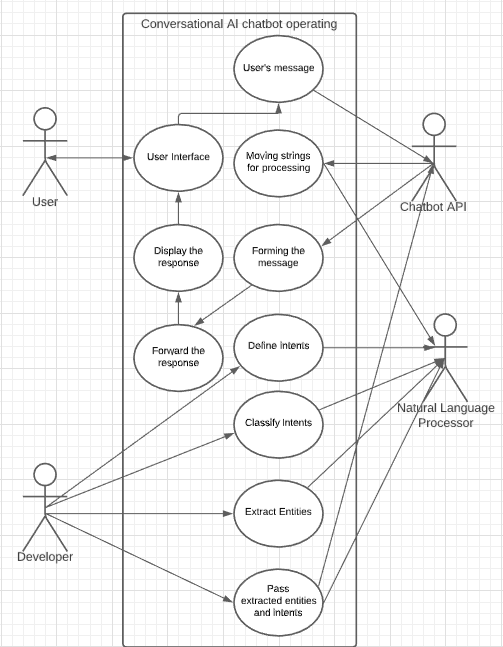


Figure 14: Use case diagram

User: can only access user interface to send the message and see the response.

Chatbot API: receive translating the user’s message to NLP for processing and forming the response to the user.

NLP: define, classify intents, extract entities and pass to API to translate.

Developer: Add/Update/Remove Intents and Entities

## **3. Implementation**

Here’s a quick breakdown of the components before implementing it:

train.py — the code for reading in the natural language data into a training set and using a Keras sequential neural network to create a model

chat.py — the code for cleaning up the responses based on the predictions from the model and creating a graphical interface for interacting with the chatbot

classes.pkl — a list of different types of classes of responses

words.pkl — a list of different words that could be used for pattern recognition

intents.json — a bunch of JavaScript objects that list different tags that correspond to different word patterns

chatbot\_model.h5 — the actual model created by train\_chatbot.py and used by chat.py

### **3.1. Initializing Chatbot Training**

Figure 15: Initializing training

Initialize all the lists where we’ll store our natural language data. We have our JSON file I mentioned earlier, which contains the “intents”. The JSON file looks like :

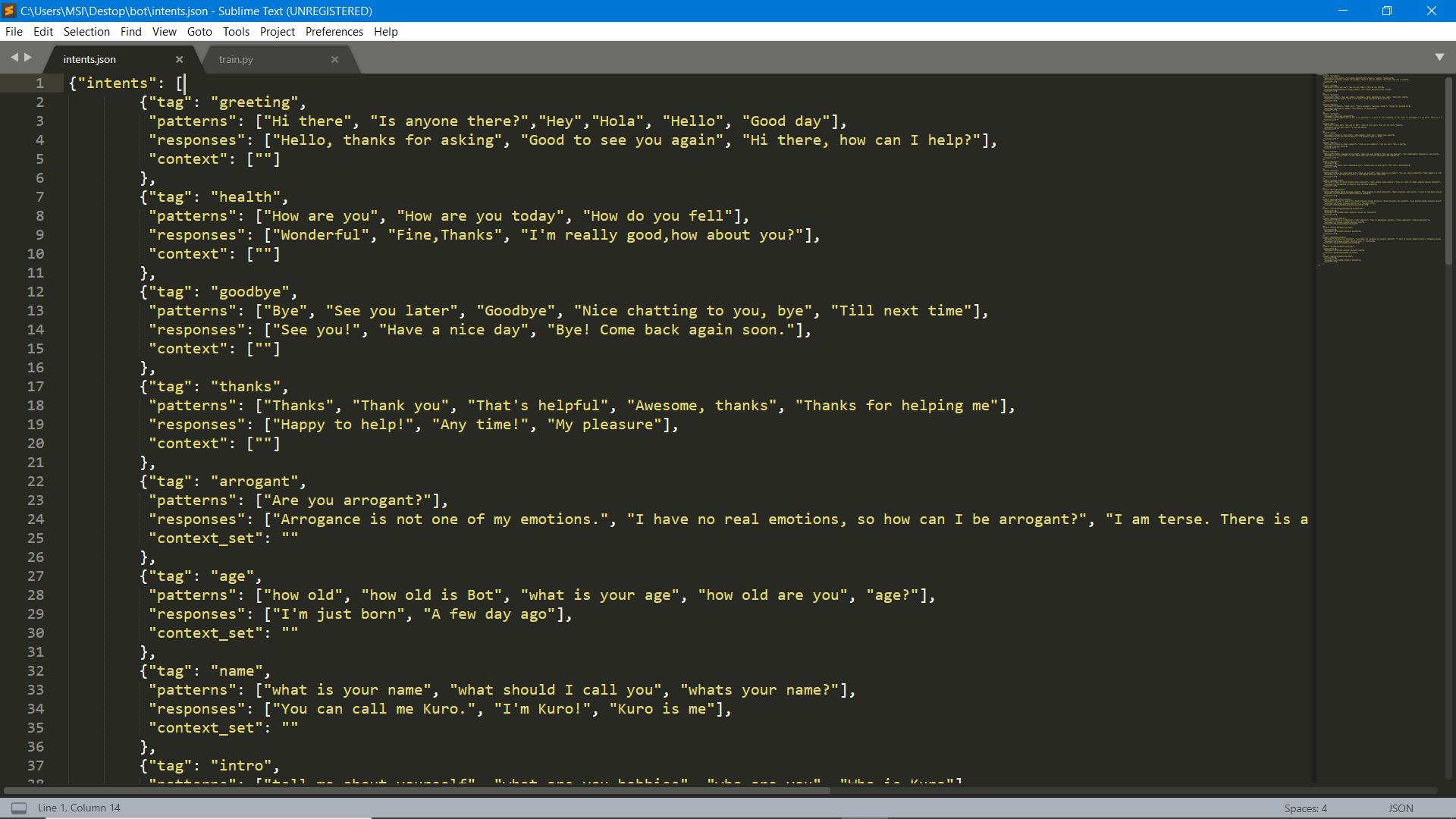


Figure 16: JSON file format

We use the JSON module to load in the file and save it as the variable intents.

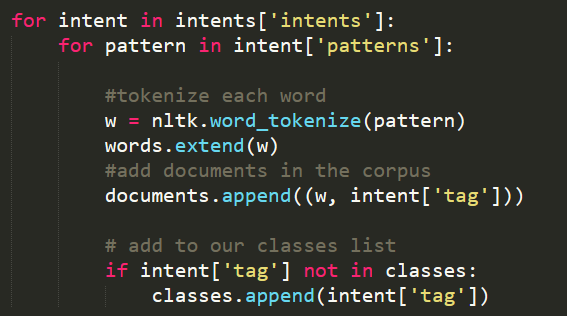


Figure 17: Tokenize the intent

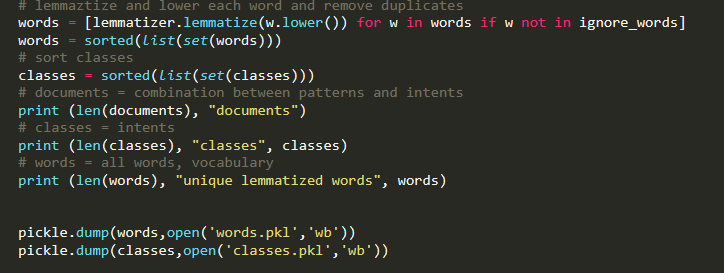


Figure 18: Tokenize the intent

I will take the words list and lemmatize and lowercase all the words inside. In case you don’t already know, lemmatize means to turn a word into its base meaning, or its lemma.

For example, the words “walking”, “walked”, “walks” all have the same lemma which is just “walk”.

### **3.2. Building the Deep Learning Model**

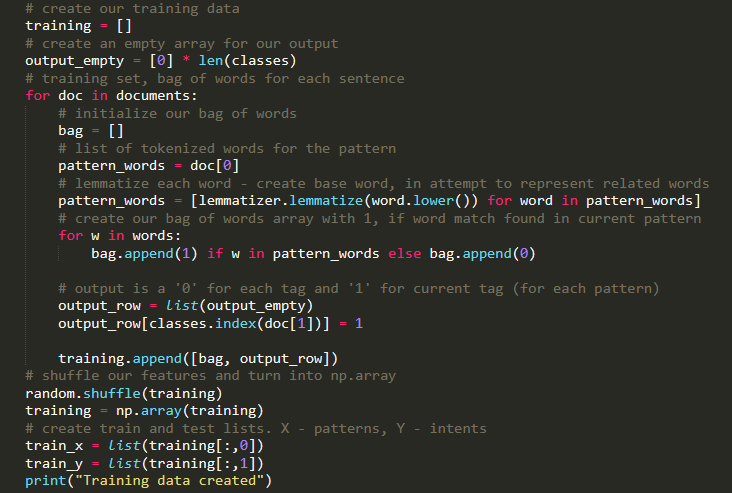


Figure 19: Building the Training model

Let’s initialize our training data with variable training. We’re creating a giant nested list which contains bags of words for each of our documents. We have a feature called output\_row which acts as a key for the list. We then shuffle our training set and do a train-test-split, with the patterns being the X variable and the intents being the Y variable.

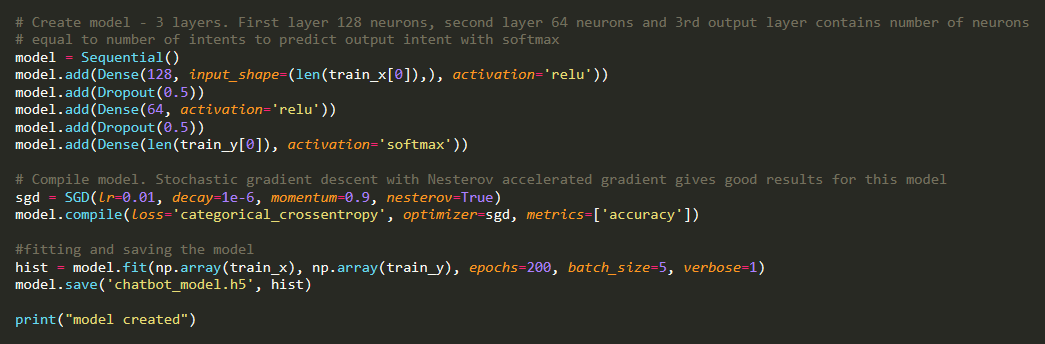


Figure 20: Initialize the Tranining

Now that we have our training and test data ready, we will now use a deep learning model from Keras called Sequential.

### **3.3. Build simple GUI chatbot**

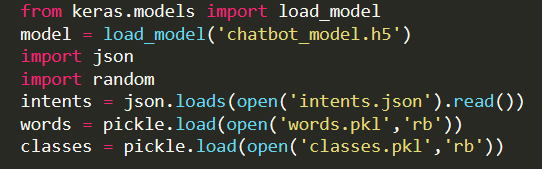


Figure 21: Import infomation

We import the information from our files.



Figure 22: Processing for running the GUI

Here are some functions that contain all the processes for running the GUI and encapsulates them into units. We have the clean\_up\_sentence() function which cleans up any sentences that are inputted. This function is used in the bow() function, which takes the sentences that are cleaned up and creates a bag of words that are used for predicting classes (which are based on the results we got from training our model earlier).

In our predict\_class() function, we use an error threshold of 0.25 to avoid too much over-fitting. This function will output a list of intents and the probabilities, their likelihood of matching the correct intent. The function getResponse() takes the list outputted and checks the JSON file and outputs the most response with the highest probability.

Finally, our chatbot\_response() takes in a message (which will be inputted through our chatbot GUI), predicts the class with our predict\_class() function, puts the output list into getResponse(), then outputs the response. What we get is the foundation of our chatbot. We can now tell the bot something, and it will then respond.



Figure 23: Build simple GUI chatbot

We can create our GUI with Tkinter, a Python library that allows us to create custom interfaces.

We create a function called send() which sets up the basic functionality of our chatbot. If the message we input into the chatbot is not an empty string, the bot will output a response based on our chatbot\_response() function.

After this, we build our chat window, our scrollbar, our button for sending messages, and our text box to create our message. We place all the components on our screen with simple coordinates and heights.

**Running Chatbot**

Here’s what the finished product will look like.

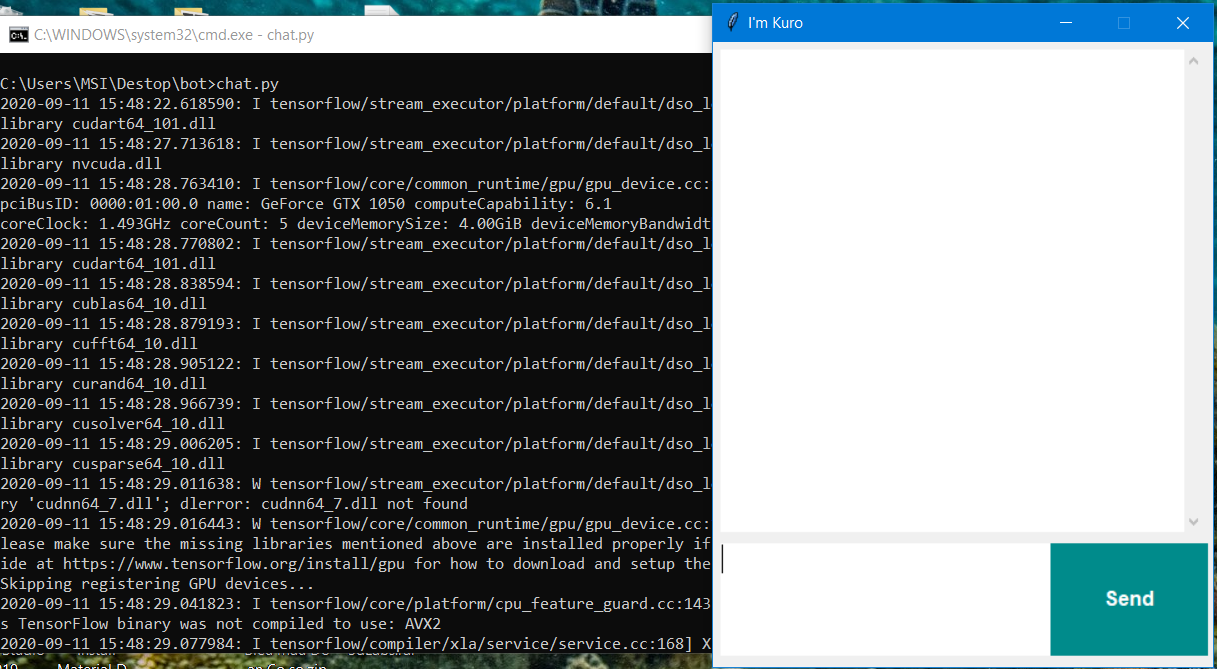


Figure 24: Simple interface

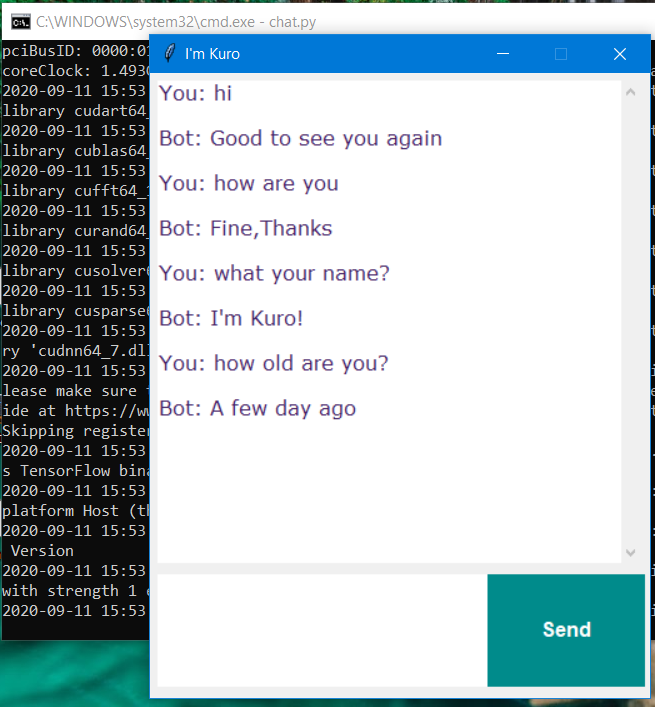


Figure 25: Simple conversation

# IV. Evaluation

There is no single perspective can deliver a universal framework for chatbot evaluation. We evaluate it through user experience’s perspective, which is the most popular evaluation method for chatbots, where users interact with the target chatbot and rate the experience from dissatisfaction to satisfaction.

Chatbots are essentially an interactive-typed software application, we will test our project through usability point of view, and focus on measures such as task completion, user satisfaction.

Human-machine interaction is the important feature of our project to evaluate as our primary aim. Through our test, although we can talk our chatbot successfully, we can confirm that our JSON file limited in terms of the variety of possible intents and responses. We can update JSON file to make it complex, but to create a chatbot can talk like a human requires far more complex to do it.

The process of training an ML model involves providing an ML algorithm with training data to learn from. The term *ML model* refers to the model artifact that created by the training process. Although, we found task completion no error thanks to the training model, but to retrain model will affect the user’s experience negatively.

# V. Conclusion

## **1. Overall Result**

In this project, our group knows many things about researching for developing a method for a chatbot. Through that, we will split into two parts to easier judge into both sides.

### **1.1. Achievements**

Judging by the overall results, we have made out a very first basic chatbot; it has featured some functions such as

  + Chatbot can say ‘Hi’ to the user.

  + Some question can be asked to the user.

  + User can find it convenient when interacting with it

And judging in knowledge aspect, our group has improved the programming skills and techniques on Python, understanding more about developing the chatbot, understanding more about developing chatbot methods, developing group working skills, etc, …

### **1.2. Limitations**

The main limitation is the lack of training data. Because of some confidential and privacy reasons, we can only provide a limited amount of data whether or not is relevant. And because of lack of training data, it also limits the performance.

Because this is a basic product, so it will come to no surprise that there are barely many functions in it.

These are the issues we will improve in the next upcoming projects, and we will figure more function for our chatbot.

## **2. Future improvements**

In the next upcoming projects, we will continue to research more method and experimenting with an alternative neural network to find it whether or not it is easier.

Also, we will try to improve it by adding more advanced and professional features to our chatbot.

# VI. Reference:

1. What Do Your Customers Actually Think About Chatbots?

<https://www.userlike.com/en/blog/consumer-chatbot-perceptions>

2. Chatbot Report 2019: Global Trends and Analysis

<https://chatbotsmagazine.com/chatbot-report-2019-global-trends-and-analysis-a487afec05b>

3. Spiceworks Study Reveals 40 Percent of Large Businesses Will Implement Intelligent Assistants or Chatbots by 2019

<https://www.spiceworks.com/press/releases/spiceworks-study-reveals-40-percent-large-businesses-will-implement-intelligent-assistants-chatbots-2019/>

4. Mental health guide for college students

<https://collegestats.org/resources/mental-health-guide/>

5. Weizenbaum 1996, p.36

<https://dl.acm.org/doi/10.1145/365153.365168>

6. Chatbots: Are they really useful ?

<https://www.researchgate.net/publication/220046725_Chatbots_Are_they_Really_Useful>

7. Chatbots vs Ai: What’s the difference?

<https://www.netomi.com/chatbots-vs-ai-whats-the-difference>

8. A framework on chatbot evaluation

<https://isquared.wordpress.com/2017/01/24/a-framework-for-chatbot-evaluation/#:~:text=The%20user%20experience%20perspective%3A%20chatbots,task%20completion%2C%20user%20satisfaction%20etc.>