VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Yi Zhou

Robot Application In The Medical Area

Pepper Robot Health Assistant

Technology and Communication
2019

FOREWARD

I would like to express my appreciation to everyone who has helped me during my university life.

Thanks to my supervisor Dr. Yang Liu, I can now access more knowledge which goes beyond books and theory. With his help, I had a chance to learn about the robot and Artificial Intelligence. Under his guidance, I learned more about experiments and practice. He provided the chance to go aboard and see the different word.

And also, I would like to show my gratitude to all the staff of VAMK, thanks for your guidance and teaching.

Finally, thanks to my family and friends.

Yi Zhou

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
2+2 Double Degree Program of Information Technology

ABSTRACT

Author	Yi Zhou
Title	Robot Application In The Medical Area
Year	2019
Language	English
Pages	73
Name of Supervisor	Yang Liu

With the development of Big Data and Artificial Intelligence, the robot is currently deployed in many areas. This thesis aims to develop a health assistant program running in Pepper robot. After the training process, the robot can answer some basic health questions.

This thesis is mainly divided into five parts: Technology background, Data collection, Data training, Response, and Result. Technology background indicates relative technical specification such as a brief introduction to Machine Learning and Chatterbot library. Data collection mainly introduce how to collect data and re-organize data. The training process is presented in the Data training part, all the data mentioned above is trained by the Chatterbot library training method and additional algorithms. The response process, which is one of the most important parts of this thesis, indicates how to get an answer based on the question the user asks. Result part shows a demo video and improvement point of this thesis.

Before deploying this project, the data should be prepared. The medical materials and conversation will be fetched from the health website. And then, all the data will be listed systematically and making them fitted for training. After that, the robot will be trained by the prepared data and algorithm. An intelligent robot will be presented when finishing the training. Through the “ALAudioRecord” module, the Pepper robot records the questions the user asks and then store them in the local space. Google Speech API will upload the voice file. Google Cloud analyzes the voice file and converts it into a text message. The algorithm splits messages and mark them. Finally, the Pepper robot gets a suitable answer based on the question.

Python is the programming language for this project, the version is Python 2.7 for Linux. Because of the limited version of the Naoqi system, all the code run under Ubuntu 16.

Keywords Pepper, Chatterbot, NLP

CONTENTS

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
POS	Part of speech
AI	Artificial Intelligence
ML	Machine Learning
NLTK	Natural Language Toolkit
API	Application Programming Interface
SSL	Secure Sockets Layer
TTS	Text To Speech
HTML	Hypertext Markup Language
DOM	Document Object Model
URL	Uniform Resource Locator

LIST OF FIGURES AND TABLES

TABLE

1 INTRODUCTION

1.1 Background

With the development of software and hardware, AI becomes more popular when comparing earlier, thanks to the Big Data research area and better algorithms, the programs can do better than humans in some specific areas, such as AlphaGo. Besides in the software area, there is also a large improvement in the hardware area, especially in the robotics area. By combining those two parts, more intelligent robots are presented in public, which means that robot become smarter and everyone has a chance to train their own robot. In the medical area, there is always a lack of human resources, especially in some poor areas. Human resources are expensive and limited. To solve this, some hospitals have already replaced humans by a robot in some jobs for example in the reception.

1.2 Purpose

This thesis provides a way to decrease doctors' workload and make everyone have a chance to access a better medical condition. Through Machine Learning technology and Pepper robot platform, robots can learn about some disease conditions and treatment methods. After that, patients can get some medical advice by asking robots, which makes the robots function as health assistants.

1.3 Overview Structure

This thesis includes seven chapters. Chapter one aims to introduce some background and purpose. Chapter two presents the technology background like Machine Learning and NLP. Chapter three introduces important technology in detail, includes the Pepper robot platform, Chatterbot library and so on. This part is aimed to make an explanation of the relative technology. Chapter four introduces the whole structure of this project, includes a user case and a flowchart. Chapter five introduces the technical specification, how to train data, how to get robot response. Chapter six shows how to run in Pepper robot and the running result. The final chapter is the conclusion and future improvements.

2 TECHNOLOGY BACKGROUND

2.1 Introduction of Machine Learning

Machine Learning is one of the search areas of Artificial Intelligence (AI). The AI research area begins with “reasoning”, then “knowledge”, and “learning”. Machine Learning is one way in which AI can solve our daily problem.

The principle of Machine Learning is that everything should be quantitative. The data is the foundation and one of the most important points of Machine Learning. The Developer inputs data into a program and assigns the algorithm to train them, to teach them. After the training, the program will have basic knowledge about what you have taught, which can make a response based on your training.

The Machine Learning Algorithm is a developed version of the common algorithms. The programs will learn automatically by themselves based on an algorithm. You can collect some sample of what you want to teach, record all their features, input all of this data into the program to train, and finally, the machine will know what you have taught. Figure 1 shows the research area.

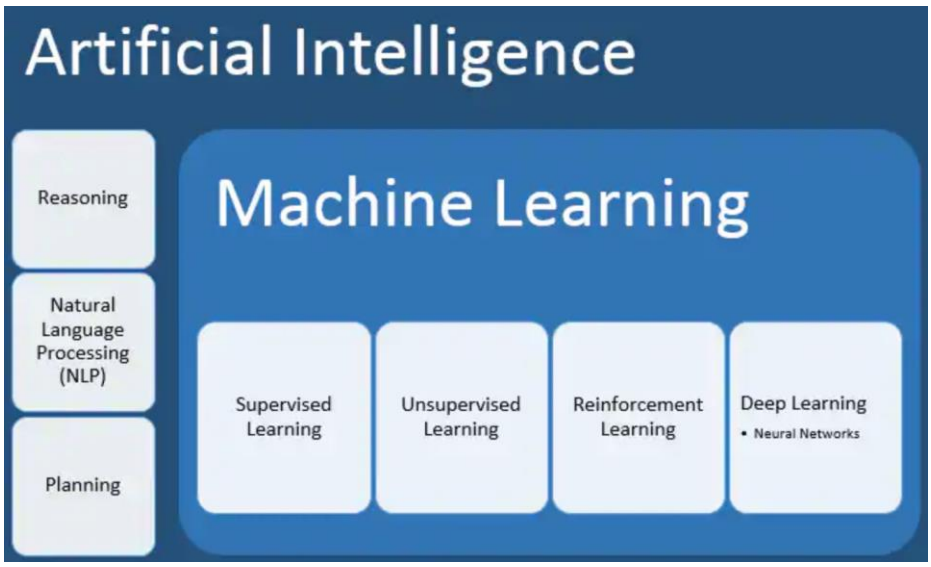

Figure 1. Machine Learning Research Area [1]

2.2 Introduction of NLP

Natural Language Processing can be understood as two parts, natural language and processing. Different from the computer language, natural language is a way of information exchange formed in the process of human development, including spoken and written language, which reflects human thinking. Now, all languages in the world belong to the natural languages, include Chinese, English. Then looking at "processing", the computer cannot process text like a human so it will require their own way to process human language. Therefore, natural language processing is that the user inputs in a natural language way, the computer will process the data based on the pre-defined algorithm in order to simulate the understanding process of a human, then return what the users want.

Now, NLP has been applied in many areas, such as translation, information extraction, question and answer systems, and conversation system. For this thesis, we focus on the conversation system and question and answering system.

"Question and answering system" means that when receiving the question user input in the natural language way, the answer is also returned by using the way of natural language, like search answer, extracting information from what you input, and conversation generation. In recent years, interactivity has also got increasing attention. A typical application is the intelligent customer service. Figure 2 presents a basic structure of QA systems.

"Conversation system" is similar to "Question and answering system". The difference is that "Question and answering system" aims to give a precise answer, the developer will not consider if the answer is suitable and like human spoken language. However, the "Conversation system" aims to use spoken natural language to solve users' problem. In this thesis, we mainly talk about Task-oriented Chabot.

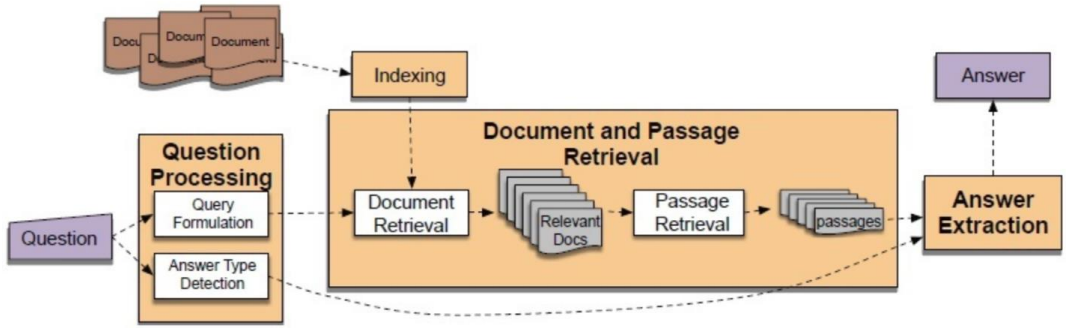

Figure 2. Question and Answer Process [2]

3 TOOLS AND TECHNOLOGIES

3.1 Platform

3.1.1 Introduction to pepper robot

The Pepper robot is a semi-humanoid robot designed by SoftBank Robotics which was introduced in a conference on 5 June 2014. Pepper is not a functional robot for a specific use. Instead, Pepper is a human-like robot to play with humans as Figure 3 shows. Besides it, The Pepper robot has the same core as the Nao robot, which is extendable and has some rich applications. Their designers hope that Pepper robot can do more by the effort of many independent developers.

Figure 3. Pepper Robot [3]

Overview:

Dimensions:

The Pepper robot does not have feet instead of using a wheel. At the front of Pepper, there is a tablet, which can interact with humans. For specific data, as Figure 4 and 5 show below.

Arm Down:

Height (mm)	Width (mm)	Depth (mm)
1210	480	425

Figure 4. Arm Down specification [4]

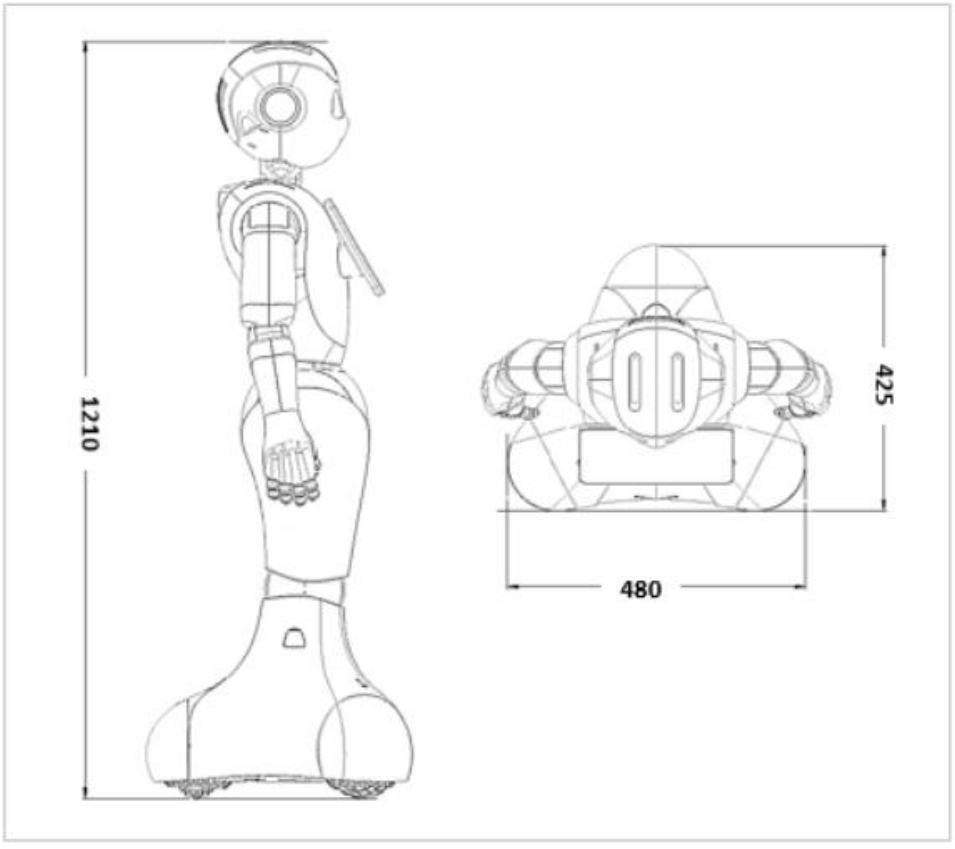

Figure 5. Pepper robot height and width [5]

Max with Arms Spread out:

Height (mm) - arm up	Width (mm) - arm on side	Depth (mm) - arm in front
1355	1196.9	648.2

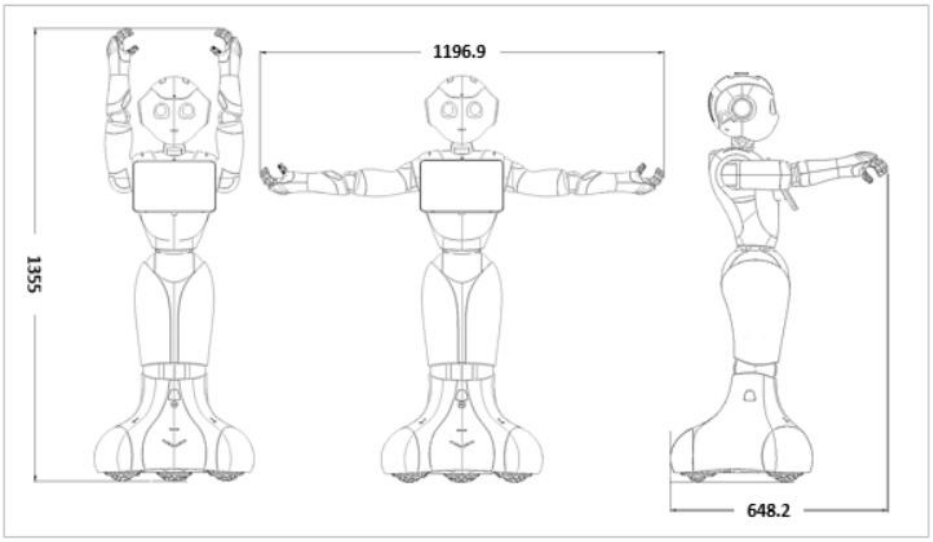

Figure 6. Max with arms spread out [6]

Motherboard:

Table 1. Motherboard information [7]

UNIT	VERSION
PROCESSOR	Atom E3845
CPU	Quad core
Clock speed	1.91 GHz
RAM	4 GB DDR3
FLASH MEMORY	8 GB eMMC

MICRO SDHC	16 GB
GPU	Intel HD graphics up to 792 MHz

Battery and Power:

The Pepper Robot has a 795 Wh battery. The minimum life cycle is about 300 ≥ 20.65Ah (70% of the capacity at 25°), expected 1000 ≥ 70% of the capacity at 25°C. Pepper can be used while it is charged by an adapter. But the battery will charge only if the Pepper robot in power off mode or plugged to its adapter.

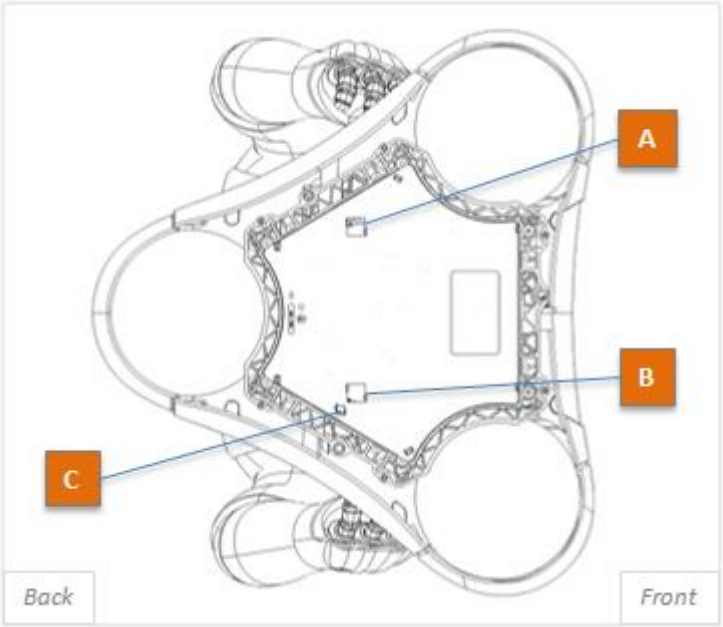

Figure 7. Robot base [8]

Table 2. Part information

Part	Connectors
A	Power +
B	Power -
C	Security

Tablet:

As I mentioned before, the Pepper Robot has a tablet on the chest part, which includes an “LG CNS Tablet” model and “Home”, “Back” buttons. The tablet provides more visible interaction between humans and the robot. Some important specifications are shown in Table 3.

Table 3. Tablet information [9]

CPU	TCC8925 1GHz ARM CORTEX-A5 CPU + 677MHz(worst) 1080p Full HD VPU + Mali400 3D/2D GPU	
RAM	1 GB	
Flash Memory	eMMC/4GB	
Media	File Format	DAT, MPG, TS, TP, TRP, 3GP, AVI, WMV, ASF, MP4, MKV, DAT,
	Video Codec	WMV 9/8/7, MPEG1, DivX, XviD, H.264
	Audio Codec	MPEG1 Layer 1/2/3, WMA, OGG Vorbis, FLAC, PCM
HTML5 video	File Format	MP4

3.1.2 Introduction to Naoqi

Naoqi is the core program of the Pepper robot, users can control the Pepper robot by it. The Naoqi structure is a robot programming framework developed by Aldebaran Company, which allows making communications between each module such as the motion module and the speech module.

Naoqi is a cross-platform framework, it can run in Windows, Linux and MacOS environment, a developer can code in the Python language. When programs developed by using C++ language, in order to run in NAOqi OS environment, it requires the cross-compiler tool CMake generates the code that can run on NAOqi OS. Besides it, a real-time application, whether running on a single device or distributed across multiple processes or modules of multiple robots, is called in the same way with the correct IP address and port number (usually 9559). The method of calling the API file remotely and calling it locally is exactly the same.

Naoqi is also called “broker”. When a user starts up the robot, the robot will load an “autoload.ini” file automatically, and this is a setting file which defines the libraries that already loaded. Each library includes one or many modules, these modules will publish their methods through the broker. At the meantime, the broker provides service of search. Therefore, the modules under the Broker or other modules in the same LAN can call the method of issuing the module of the broker. Figure 8 shows the process.

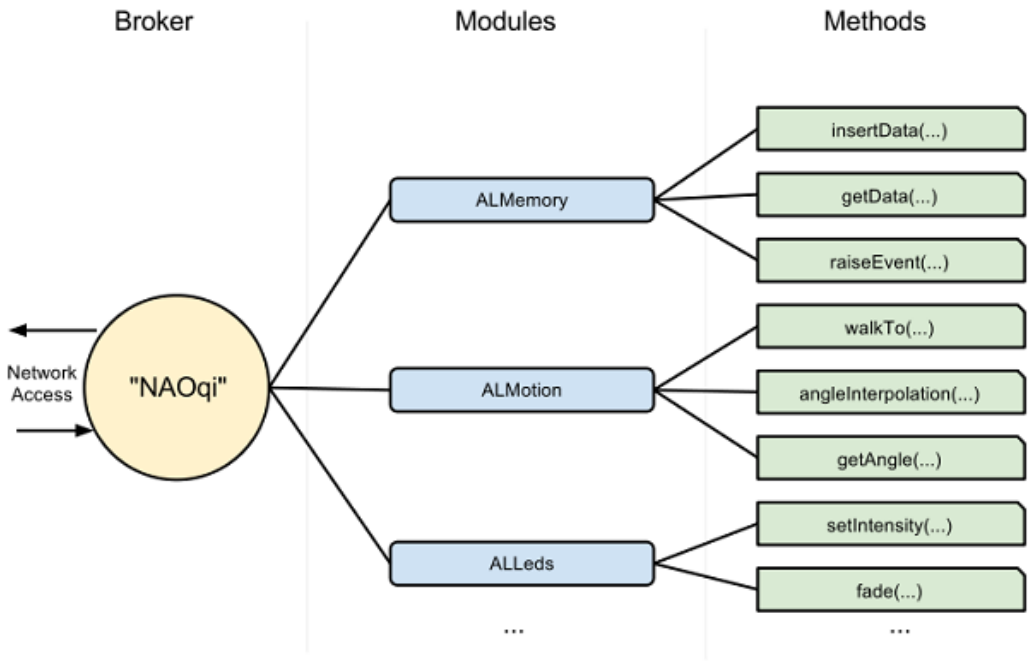

Figure 8. Naoqi process [10]

3.1.3 Python

Python is a high-level interpretive, compulsive, interactive, and object-oriented scripting language.

Python is easy to read, it uses English keywords and some punctuation marks more often than other languages. Python has a more distinctive grammatical structure than other languages. [11]

Nowadays, with the development AI, thanks to many libraries and framework, Python became more and more popular in research and commercial area. In this project, because Naoqi only supports Python 2.7, so the Python version is 2.7.

3.1.4 Web crawler

Web crawler also called “web spider” or “web robot”, which is the script that can exact some information from the website under some rules.

The web crawler is an automatic website information exaction program, it downloads website for search engine, which is the most important part of a search engine. Web crawler begins to get a website URL, after that, during the process of website fetching, it has to get new URLs continually until meeting the end requirements. But for focused web crawler, the process will be more complex. According to the filter rules the developer has pre-defined, the focused web crawler keep all the useful URLs, and store them into a waiting line. Then, it will choose a website for the next fetching based on the search strategy. Comparing with Common web crawler, Focused web crawler only needs to fetch some relative web pages.

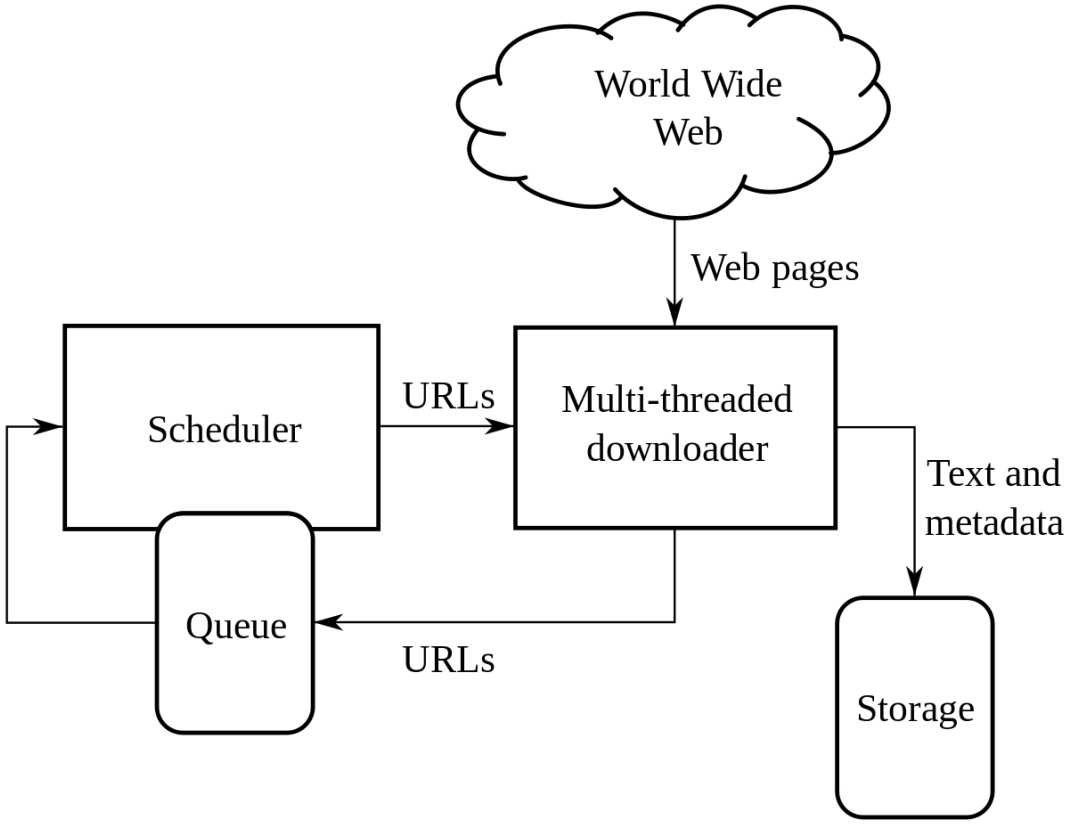

Figure 9. Web Crawler process structure [12]

3.1.5 BeautifulSoup

BeautifulSoup is a library of Python, which provides some simple, python-like functions to deal with search content and modify the web element tree. It is a tool case, which fetches data for the user by analyzing some web documents. BeautifulSoup will convert a document into a Unicode format automatically.

3.1.6 Tensorflow

Tensorflow is an open-source and free software library for dataflow. It is an outstanding math library which is implemented in machine learning application like the neural network. It is used in the research and production area at Google.

History:

In 2011, Google Brain built DistBelief as a proprietary machine learning system based on deep learning neural networks, which was applied in Alphabet Company in both research and commercial applications. Tensorflow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. Tensorflow can run on multiple CPUs and GPUs. It is also available on Linux macOS and Windows.

Advantage:

Model is flexible and easy to learn. For beginners, it has high-level API such as Keras and Slim which abstract a lot of pieces used in designing machine learning algorithms. Besides, as one of the most popular machine learning libraries in the world, it has many applications and APIs, which can be easily extended.

Brief Introduction

Tensor:

What is tensor? In the mathematics area, a tensor is an N-dimensional vector, which means that an N-dimensional dataset can be represented by a tensor. See Figure 10 below.

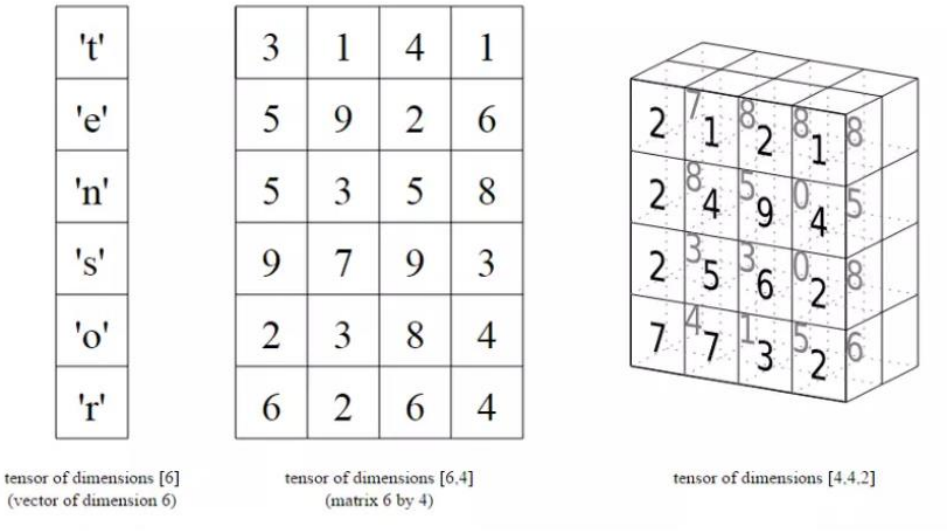

Figure 10. Tensor [13]

With the increment of dimensions, data will be more complex, for instance, one 2×2 tensor can be a 2×2 matrix, 2 × 2×2 cab be two 2 ×2 matrix, but if we increase the dimension, it will be hard to understand and present.

Flow:

What is Flow? Flow is a calculation chart or a simple chart, which cannot be a circle, in this chart, each node represented a calculation operation. After each operation, it will produce a new tensor. Like the on next page show.

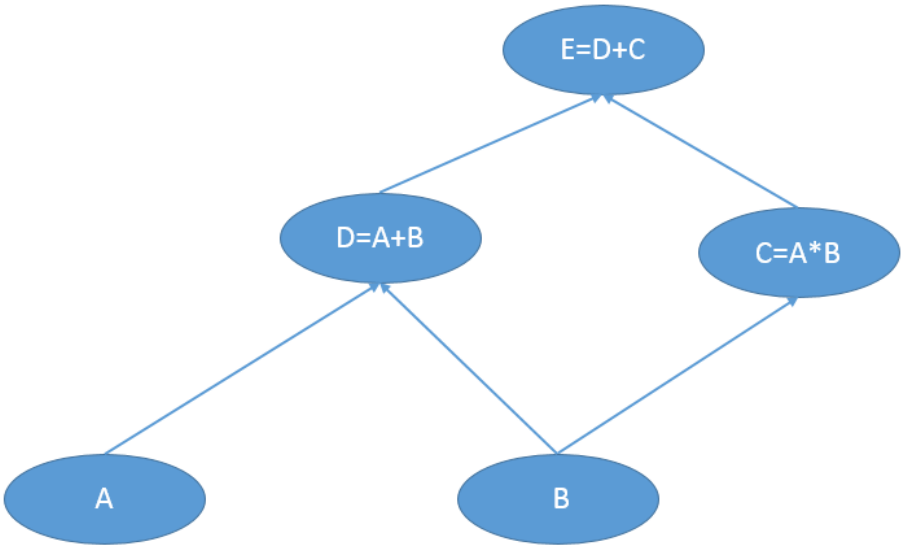

Figure 11. Flow process

Figure 11 shows the progress of function E= (A+B) + (A*B). The lowest node is always a tensor, which makes sure that all the operations cannot start at the beginning of this chart. For each operation, it will receive a tensor and produce a new tensor. In the meantime, tensor only works as the child node, they should always serve as an input.

Besides the character mentioned above, one of the most important characters is that at the same node level, each operation is independent, such as calculating “D” and “C” at the same time.

According to this point, Tensorflow allows users to run programs parallelly, all nodes or operations will be called automatically and computed parallelly.

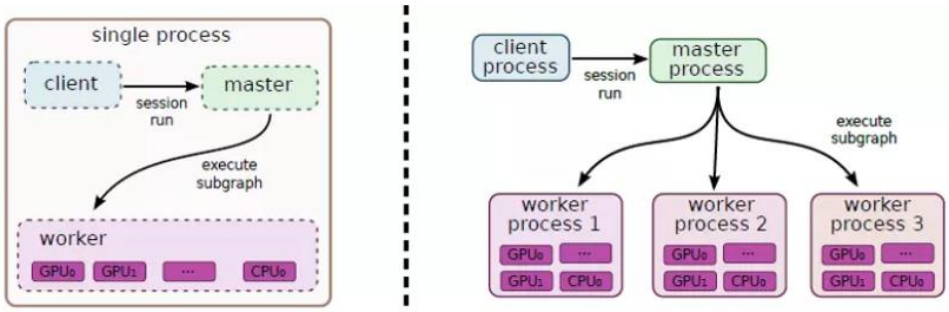

Figure 12. Tensorflow Process [13]

3.1.7 Natural Language Toolkit

Natural Language Toolkit also called NLTK, which is a leading platform to deal with human language. It provides some simple interfaces, according to these interfaces, programs can access more than 50 corpora and the word resources like WordNet. Besides, it also includes a database for classification, tokenization, stemming, parsing, and semantic reasoning.

Based on the manual programming guide on statistical linguistic topics and comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users.

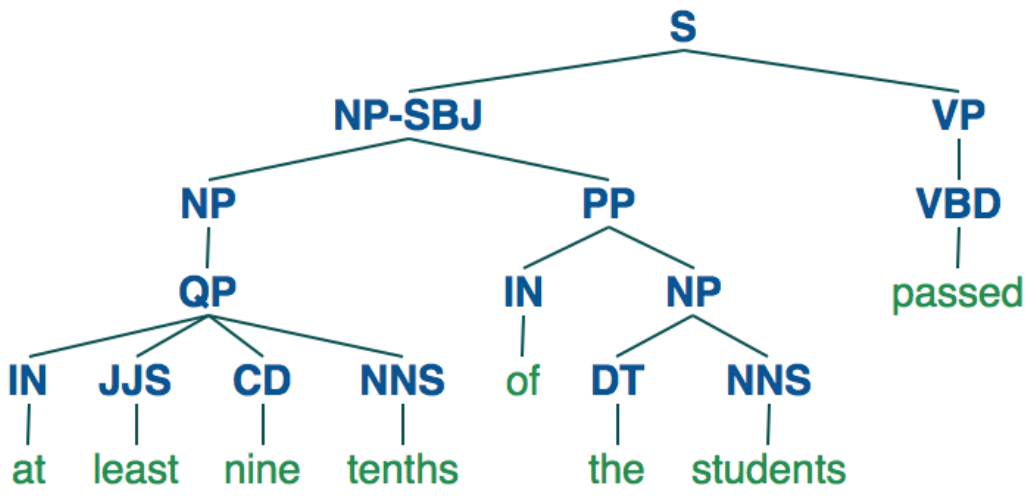

Figure 13. NLTK [14]

3.1.8 WordNet Interface

WordNet is one of the NLTK corpora readers. It is a dictionary that contains the relationship between one word and another word. WordNet has its own different structure; the basic unit of structure is “Synset”. Users are able to get a suitable word from the word sets to express a known concept. If we call WordNet as a database, “Synset” is more like a primary key. Each piece of data represents a meaning. WordNet not only lists concepts by means of synonym collection but also associates these synonym sets with certain relationship types, such as synonym, antisense relationship, upper and lower relationship, whole and partial relationship and inheritance relationship. Figure 14 shows the concept hierarchy.

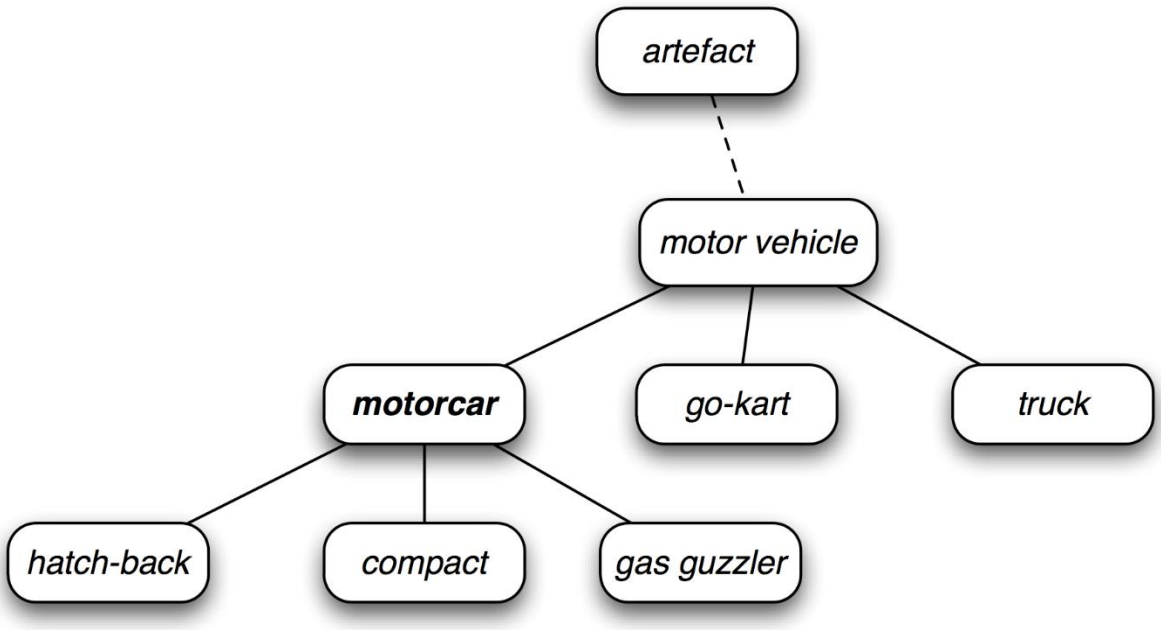

Figure 14. Wordnet concept hierarchy [15]

3.1.9 ChatterBot

Overview:

ChatterBot is a Python library, which can easily generate a response based on users' input. ChatterBot combines a variety of machine learning algorithms to make a different type of response. Developers can make a chatbot easily.

Language support：

The language structure of ChatterBot is independent. ChatterBot allows itself to be trained to speak any language. Besides it, the implementation of the machine-learning structure allows improving robots' own possible response as the conversation between a human and a robot by using an agent instance.

Principle:

In the beginning, an untrained instance of ChatterBot has no knowledge of how to communicate with users. Once users input some sequences or samples, the library model will save their input and responses. When more and more sequence users input, the response accuracy of this instance will increase. Based on the input of users, the program will select the best matching result to response. [16]

Training:

As an easier Python library, ChatterBot includes a simple tool to help developers train a chatbot instance. Chatterbot begins to load a sample dialog into the program database. In the meantime, Chatterbot will create some graph data structure that shows the sets of known statements and response. After providing the samples' data set, ChatterBot will build some entries in the knowledge structure in order to make a correct input and response.

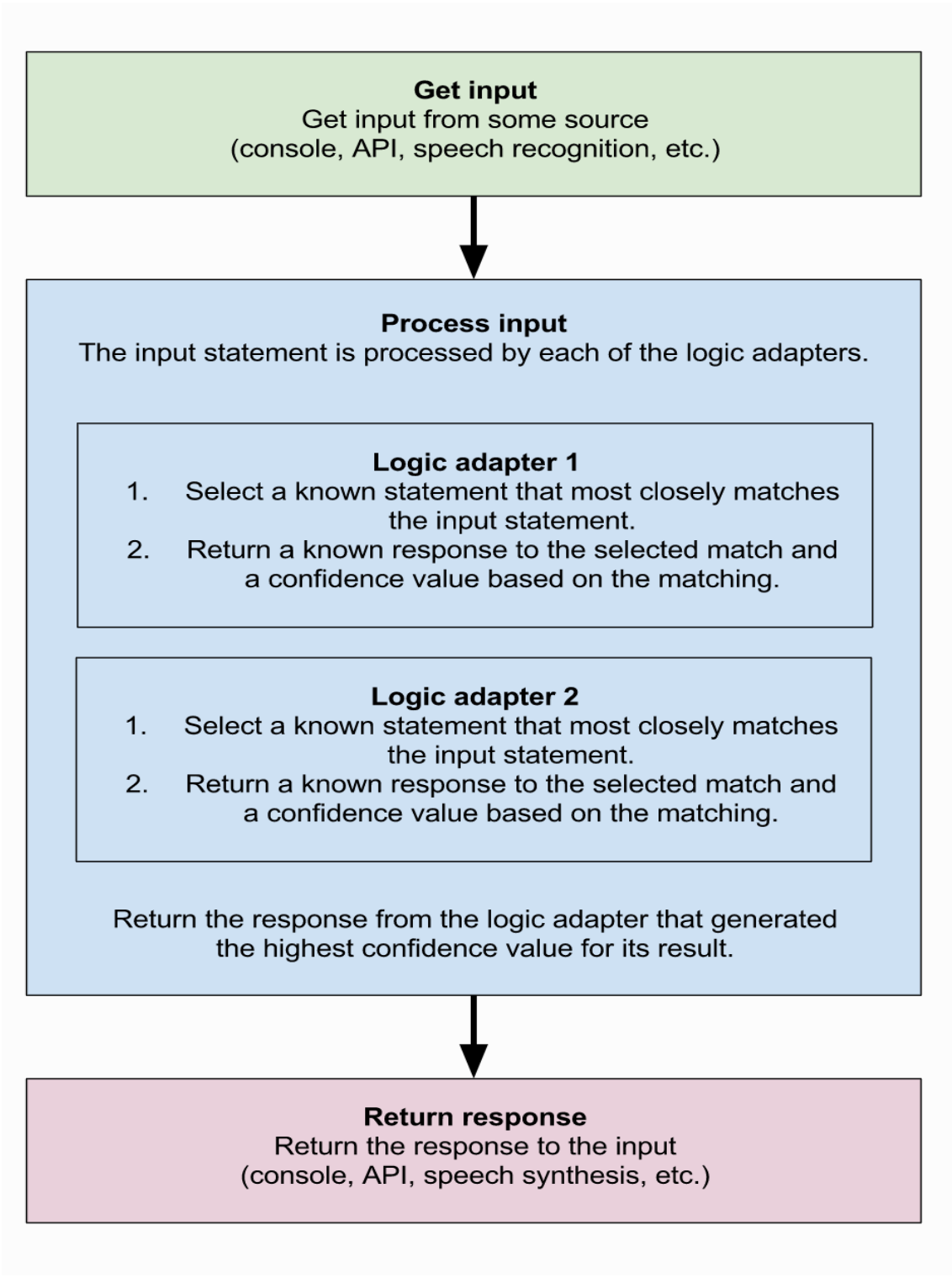

Figure 15. Chatterbot process flowchart [17]

Training preparation:

Training corpus is one of the most important factors of ChatterBot, all the corpus will be saved on a local machine.

Training processing:

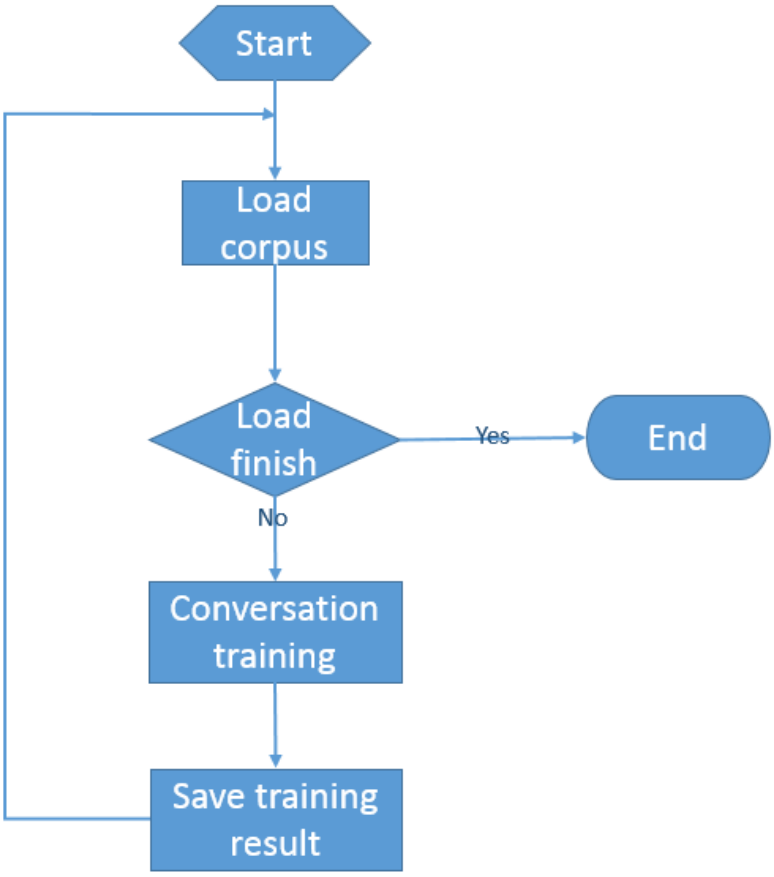

Figure 16. Chatterbot training process

- Loading corpora: before training, all the corpus should be stored in the local machine or loading into memory.
- Conversation training: according to the training corpus, the machine tries to learn it and understand it.
- Save training result: record all the training result.

Comparisons:

ChatterBot uses statement objects to hold information. An important part of how a chatbot selects a response is based on its ability to compare two. There are a number of ways to do this, and ChatterBot comes with a handful of methods built in for use to use. [18]

Advantage and Disadvantage

Advantage:

- The training corpus can be saved on many types of media
- The training result can be saved on many types of media
- The algorithm of response matching support different algorithm, such as Similarity matching, mathematical estimation algorithm, etc.
- A robot can be trained in a different language

Disadvantage:

- Large corpus needs to much time to train
- The response of large corpus will delay

3.1.10 Speech Recognition

Speech Recognition also called Automatic Speech Recognition, (ASR), aims to convert the human language into the input language that computers can understand. At present, most speech recognition technologies are based on statistical models. From the perspective of the speech generation mechanism, speech recognition can be divided into two parts: the speech layer and the language layer. The mainstream algorithms of current speech recognition technology mainly include dynamic time warping (DTW) algorithm, vector quantization (VQ) method based on the nonparametric model, hidden Markov model based on parameter model (HMM), artificial neural network based on artificial neural network. Speech recognition methods such as (ANN) and support vector machines.

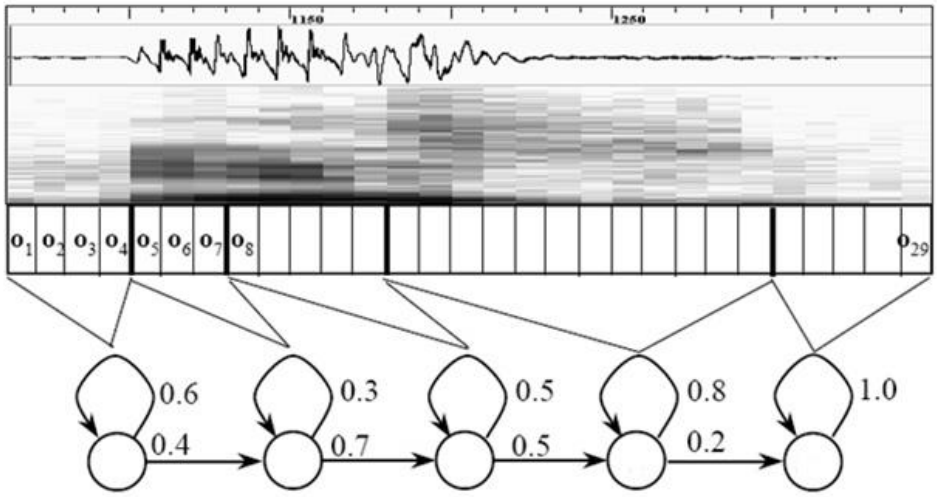

Figure 17. Speech Recognition [19]

Now, many companies pay more attention to Speech Recognition. Depending on different languages, many companies offer their APIs, such as Google Could Speech-to-Text, Bing Speech-to-Text, and iFlytek.

This thesis will use Google Cloud Speech-to-Text, because of higher recognition rate and higher speed in the English language area.

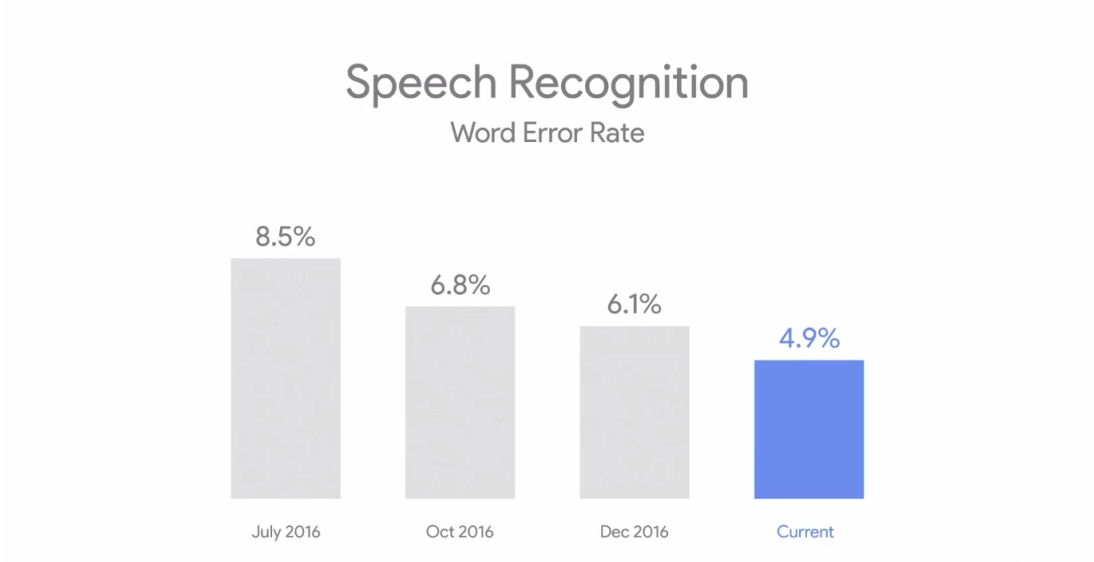

Figure 18. Speech Recognition word error rate [20]

3.1.11 Google Colab

Colab is a Python notebook tool released by Google, which supports TensorFlow, BigQuery, and Google Drive. Colab includes TensorFlow, Matplotlib, Numpy, Pandas and the other basic library of Deep Learning. We can also add the external library by ourself.

Colab can make the program run in a different environment, such as Python 2 and Python 3, which is user-friendly. This platform provides K80 GPU, this GPU was released in 2014, Kepler structure. Beginners can train their own data by using this free GPU.

4 OVERALL DESIGN

When fetching information such as disease symptom, treatment method, and other information by using a web crawler, all gained information is disorder and mess which is not readable for the computer. So we need to list and re-organize them for the computers. Secondly, the program stores data in different data training files. Based on different classification of disease, the program stores fetched data in different files in the way of conversation. Then data will be prepared and ready to be trained.

Training data is based on the Chatterbot library "train" method. In order to achieve a simple conversation, the model is also trained with corpus data.

When it starts, users ask for help based on their health conditions, the sentence will be recorded as a ".wav" file. After that, the Pepper robot will upload those files into Google Cloud and call "voice recognition" module to convert speech to text. Based on the trained result, Pepper will get a response and reply to users. Figure 19 shows the training process flowchart, Figure 20 shows the response process flowchart.

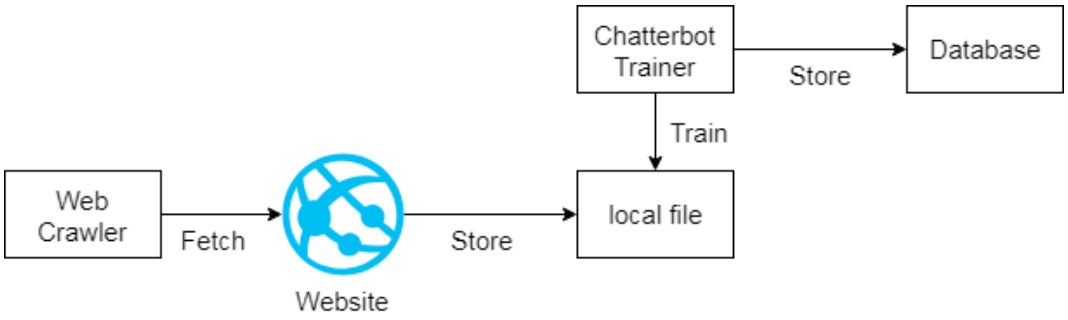

Figure 19. Training flowchart

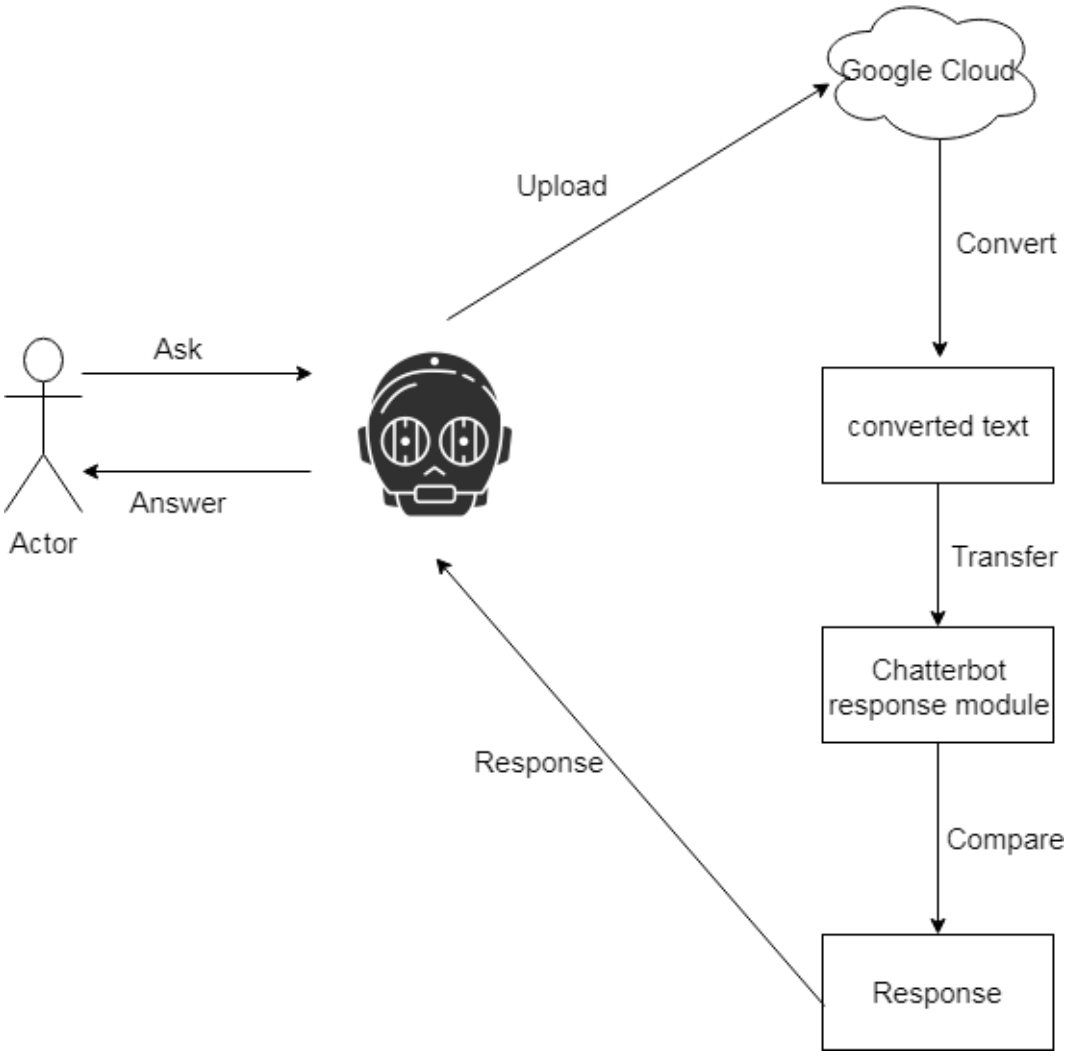

Figure 20. Response flowchart

Figure 21 shows the whole structure of this project.

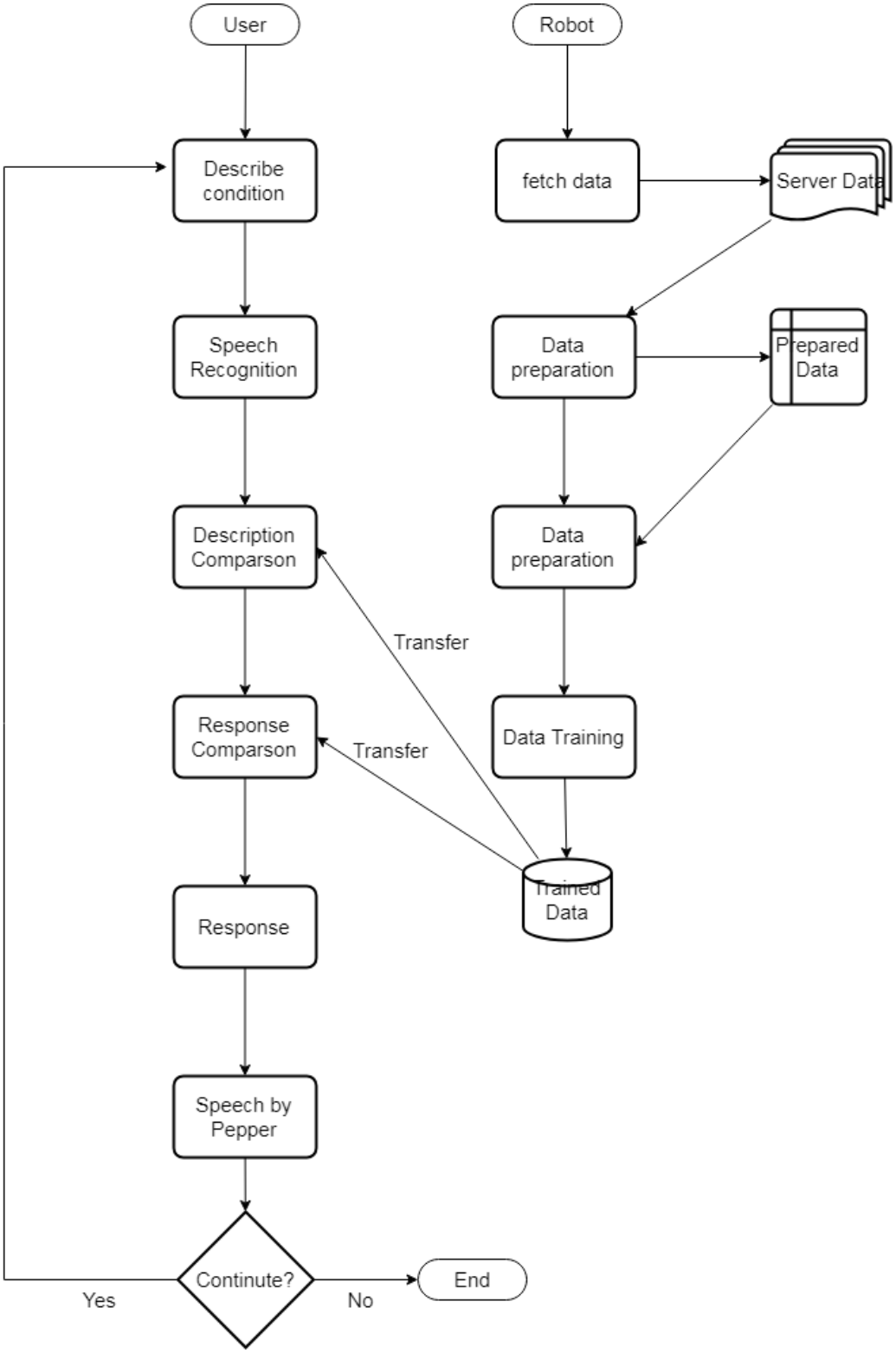

Figure 21. Project whole structure

5 TECHNICAL SPECIFICATION

5.1 Data Preparation

5.1.1 Information Exaction

Web Crawler

Before users deploy this application, the Pepper Robot needs to be trained by data, which can be collected by using a web crawler. The web crawler is also called Network Spider or robot. Now, the Internet includes so large data, and all the data is stored in the server, the user can look through the web content by a hyperlink. The web crawler aims to download server data by using URL. However, different websites have different rules and different tags. In order to get the correct information, the web crawler needs to filter the information firstly.

Website

In this thesis, we need very large data about disease symptoms and treatment methods. So, we focus on a health website.

- https://kidshealth.org/

Fetch tool

- Urllib
  Urllib is a component of Python which is used to get the URL. It provides a "urlopen" function as a simple interface. This method can create a class file object that represents the remote URL and then manipulates the class file object like a local file to get the remote data.
- BeautifulSoup
  As mentioned before, this library is a tool to analyze websites and get the information. At first, defining rules and creating a BeautifulSoup object to exact web information.

- SSL
  The SSL module is used to wrap socket objects using secure sockets layer (SSL), enabling data encryption and terminal authentication. Python applies OpenSSL to this module.

5.1.2 Web Crawler design

This Web Crawler is designed to fetch training data from a health website. Before fetching, it creates an SSL certificate to get the authorization. Then we can build the "BeautifulSoup" object. After that, we create a filter to get the correct information. Finally, we store data into different files. Figure 22 shows the process chart.

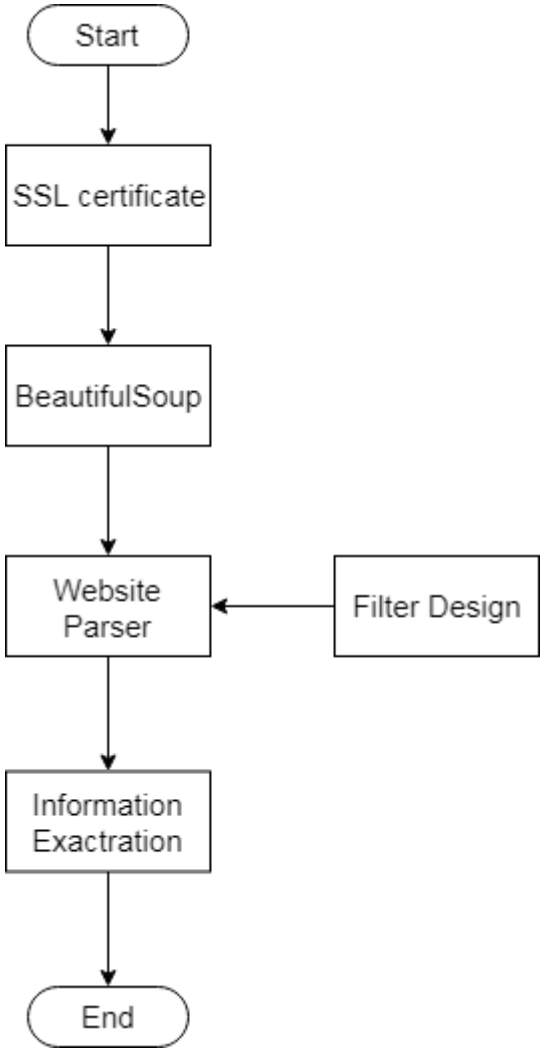

Figure 22. Web Crawler flowchart

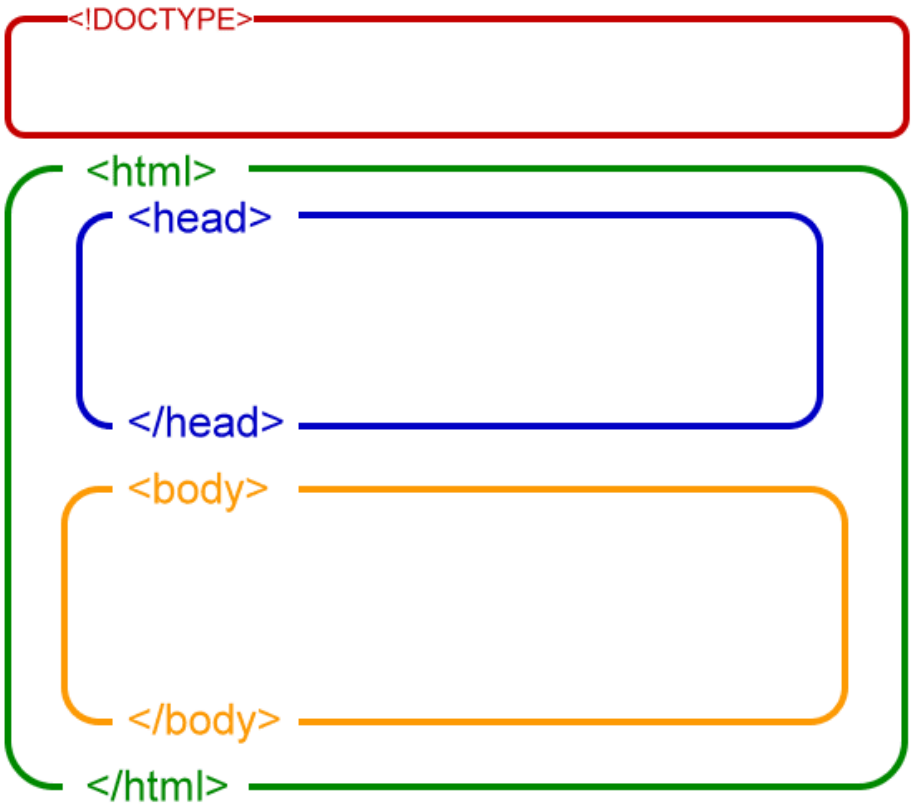

Figure 23. HTML page structure [21]

As Figure 23 shows above, an html page mainly includes DOCTYPE, html content. HTML content contains <head> and <body> tags. Almost all the content is defined in the <html> and <body> tag. Figure 24 show a page under browser develop mode.

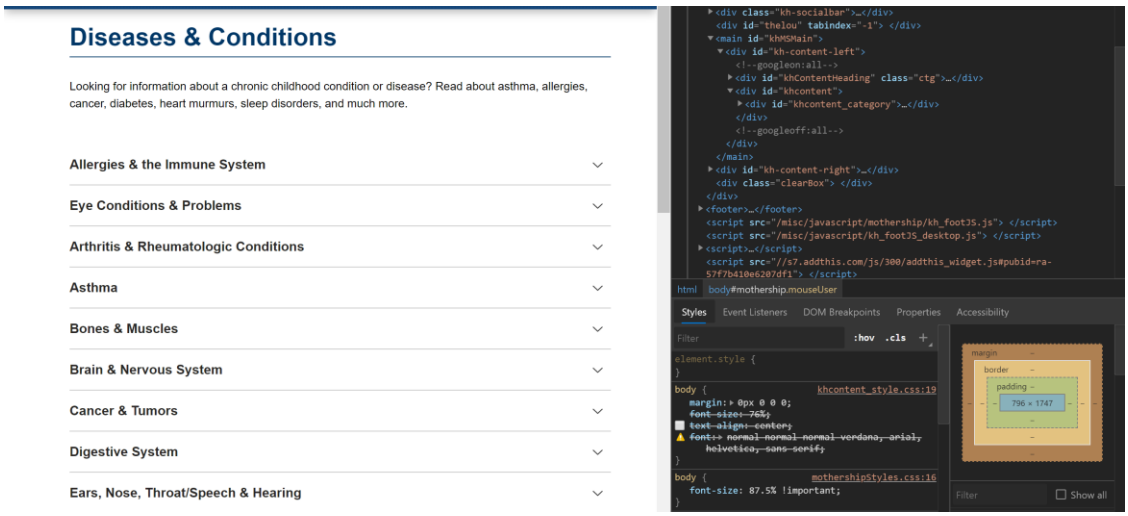

Figure 24. HTML page detail

Finding out the meaning of tags, Table 2 show some basic information of HTML tags

Table 4. HTML tags content

Tag	Function
<!DOCTYPE html>	The definition of this document is HTML5
<html>	Root element of an HTML page
<head>	Contain meta information about a document
<title>	The title for this document
<body>	Contains the visible page content
<h1>	The definition of a large heading
<p>	Define a paragraph
	Define a division or a section
<h2>	Second class HTML heading
	Define an unordered list
	Define ordered list
<a>	Define a hyperlink, link one page to another
<href>	The “href” attribute specifies the URL of the page the link goes to.

Before fetching data, the Web Crawler should know which part of the content we want. So, at first, we design a content filter based on tags. Depending on Beautiful-Soup, we need to find the seed URL in order to set a goal, then the Web Crawler will retrieve the page content and find all the tags we already defined through HTML DOM tree. Figure 25 below shows an HTML DOM tree structure.

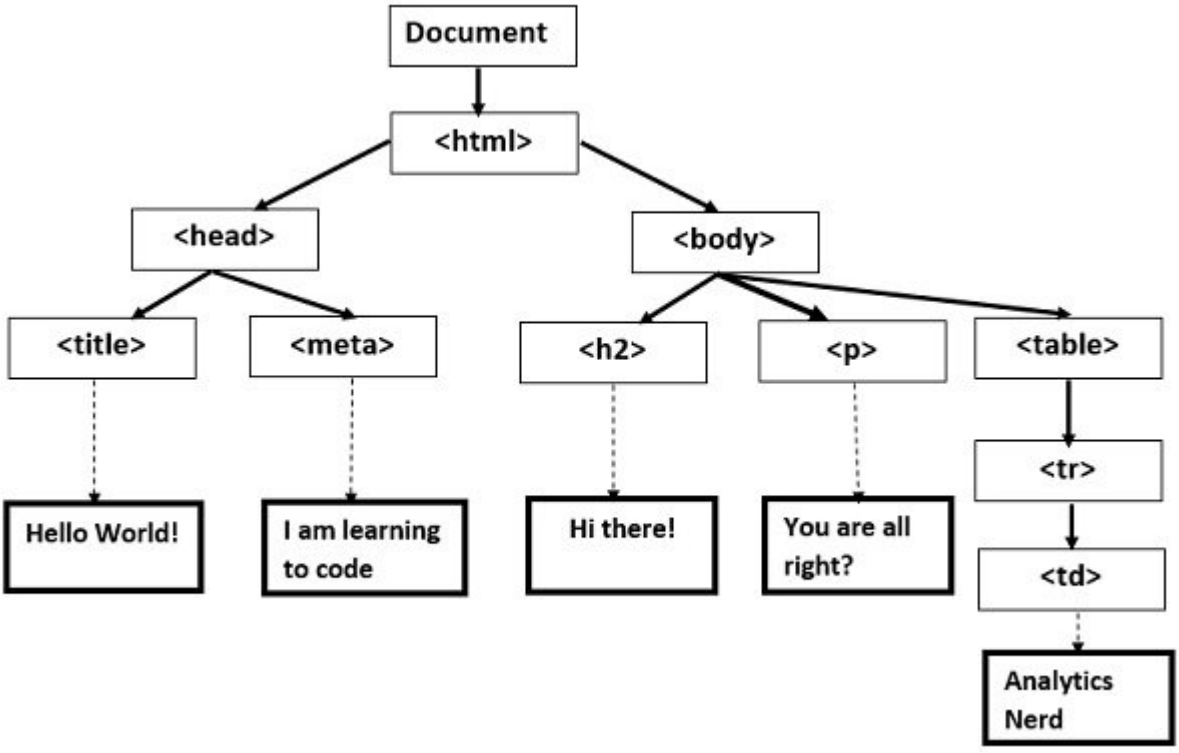

Figure 25. HTML page content structure [22]

Through analyzing websites, we can find the required page content. For this thesis, a web crawler will find basic medical information such as disease name and disease type from the first level link, as Figure 26 shows in the red mark. And then, after exacting the basic information, the web crawler will get the second level link in order to get more details, like the green mark in Figure 26.

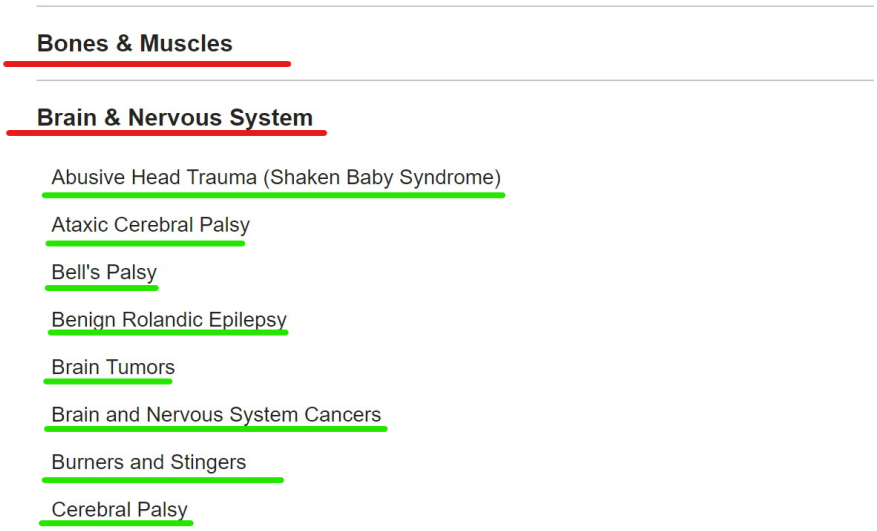

Figure 26. Disease name

By clicking the subtitle, the page will be redirected and we can find all the details of this disease, such as what causes it and how to treat it. Figure 27, 28, 29 show a part of disease information.

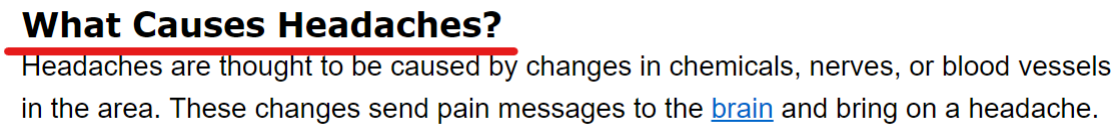

Figure 27. Disease treatment

Who Gets Headaches?

Headaches are common in kids and teens. Headaches (especially migraines) often run in families. So if a parent, grandparent, or other family member gets them, there's a chance that a child may get them too. Some kids are more sensitive to headache triggers than other kids.

How Are Headaches Diagnosed?

Your doctor will do an exam and get your child's medical history to help see what might be causing the headaches. The doctor will ask about:

- how severe the headaches are and how often they happen
- when the headaches first started
- what the headaches feel like, and where they hurt
- whether the headaches have a pattern or change over time
- any other symptoms
- any recent injuries
- anything that triggers the headaches
- your child's diet, habits, sleeping patterns, and what seems to help the headaches or make them worse
- any stress your child has
- any past medical problems
- any medicines your child takes
- any allergies
- any family history of headaches

Figure 28. Exact Data_1

How Are Headaches Treated?

Treatment for headaches depends on what the doctor thinks is the likely cause. But you can care for most everyday headaches at home.

To help ease the pain, have your child:

- Lie down in a cool, dark, quiet room.
- Put a cool, moist cloth across the forehead or eyes.
- Relax.
- Breathe easily and deeply.

Make sure your child has had something to eat and drink. Kids with migraines often just want to sleep and may feel better when they wake up. A big part of treating migraines is avoiding the triggers that can cause them. That's where a headache diary can be helpful.

Figure 29. Exact Data_2

5.1.3 Exacting basic data

All the “heading data” stored in tag. BeautifulSoup library provides “find_all” method to find specific content, as shown in Figure 30 below, all the disease classes are stored in <h2> tag, and the attribute is “class = ‘kh-category-name’”. So, we define the first level of filters to get all the disease categories names. After that, the web crawler should be able to access the disease lists by using the filler condition “<ul class=’kh-category-list’>”, all the disease links are listed in the list. After accessing all the items, the web crawler will get the related links and redirect to the disease detail. Figure 30 shows the list tag and Figure 31 indicates the heading tag content.

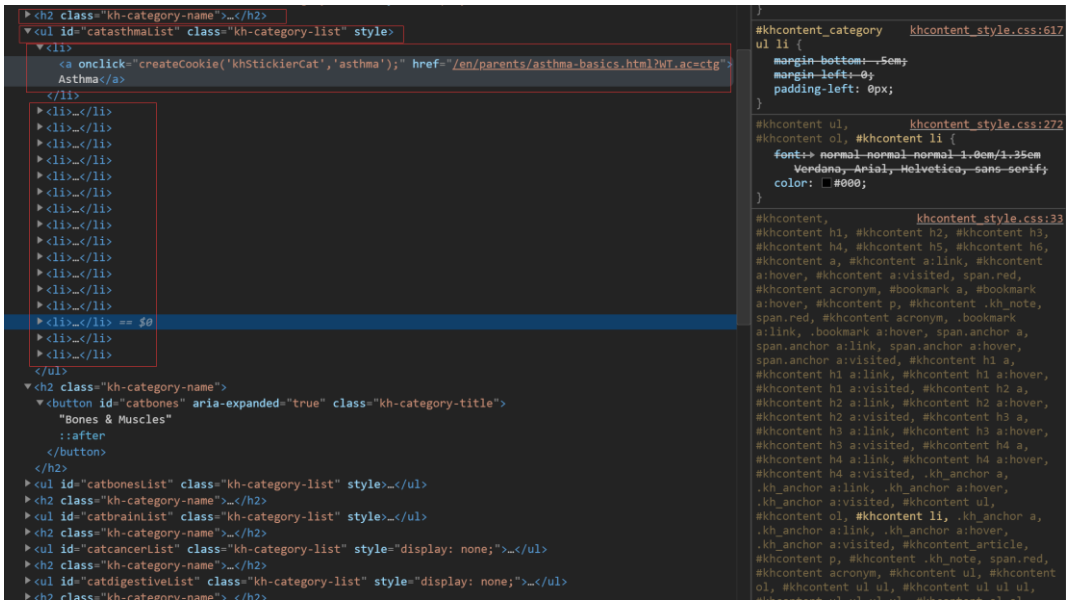

Figure 30. List tag content

5.1.4 Exacting detail

After getting all links of detailed information, the web crawler will add a link into the waiting line. All detail information is stored in “ ”. Furthermore, all the information includes “what is it?” and “How to treat it?” is stored in <p> tag, <h> type tag, and <li> tag as the Figure 31 shown below. So web crawler exacts information based in these filter conditions. Actually, all the tags belong to a root tag content, which makes sure that all the data is what we really need.

By using BeautifulSoup, web crawler is able to filter content from first level page to another page. Figure 32 shows some filter designs.

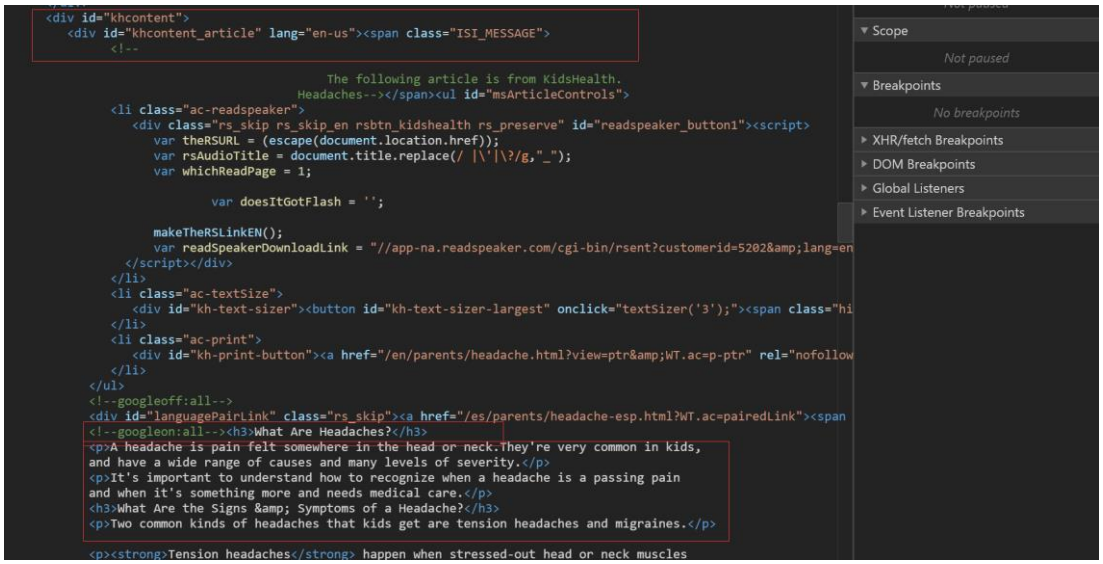

Figure 31. Tag content

```
bsObj=BeautifulSoup(html, "html.parser")
ulContents=bsObj.find_all("ul", attrs={"class":"kh-category-list"})
with open ("./data.txt", "w", encoding="utf-8") as f:
        for bulContent in ulContents:
                bulContents=bulContent.find_all("li")
                for liContent in bulContents:
                        disease=liContent.find("a").text
                        #print (disease)
                        f.write(disease+'.'+'\n')
                        link=liContent.find("a").get('href')
                        newHtml=urlopen("https://kidshealth.org"+link, context=context)
                        newbsObj=BeautifulSoup(newHtml, "html.parser")

                        khContent=newbsObj.find('div', attrs={'id':'khcontent'})
                        if(khContent!=None):
                                pContents=khContent.find_all('p')
                                if(pContents!=None):
                                        for pContent in pContents:
                                                #print (pContent.text)
                                                f.write(pContent.text+'.'+'\n')
                                subliContents=khContent.find_all('li')
                                if(subliContents!=None):
                                        for subliContent in subliContents:
                                                if("var" not in subliContent.text):
                                                        #print (subliContent.text)
                                                        f.write(subliContent.text+'.'+'\n')
```

Figure 32. Web Crawler design

5.1.5 Information Arrangement

The original information fetched from websites is excursive. However, the Chatterbot accepts ordered sentences in conversations, so all the original sentences should be arranged once again and stored in files. Figure 33 indicate soma example of original data.

```
sneezing.
itchy nose and/or throat.
stuffy nose.
coughing.
wheezing.
trouble breathing.
coughing.
hoarseness.
throat tightness.
stomachache.
vomiting.
diarrhea.
itchy, watery, or swollen eyes.
hives.
swelling.
a drop in blood pressure, causing lightheadedness or loss of consciousness.
A drop of a purified liquid form of the allergen is dropped onto the skin and
                the area is scratched with a small pricking device..
A small amount of allergen is injected just under the skin. This test stings a
                little but isn't painful..
Keep family pets out of your child's bedroom..
Remove carpets or rugs from your child's room (hard floors don't collect dust
                as much as carpets do)..
Don't hang heavy drapes and get rid of other items that allow dust to build up..
Clean when your child is not in the room..
Use special covers to seal pillows and mattresses if your child is allergic to
                dust mites..
If your child has a pollen allergy, keep the windows closed when pollen season
                is at its peak, have your child take a bath or shower and change clothes after being
                outdoors, and don't let him or her mow the lawn..
Keep kids who are allergic to mold away from damp areas, such as some
                basements, and keep bathrooms and other mold-prone areas clean and dry..
Allergies.
Allergy Shots.
Many kids have allergies — in fact,
                they're the most common cause of chronic nasal congestion in children..
Allergen immunotherapy (allergy shots) can be an effective treatment
                for certain allergies. Here are the basics on allergy shots and how to help a child
                deal with them..
```

Figure 33. Original data

Based on the disease category, we store each type of disease in different files, which is easy to add and search. All the data is stored in the ".yml" file format based on the area as Figure 34 shows below. All the related disease condition and treatment method should be shown in the conversation type. Figure 35 presents a sample of conversation.

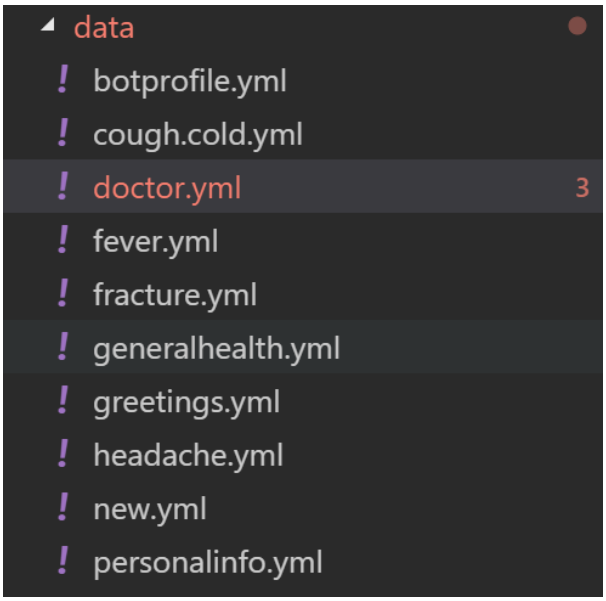

Figure 34. Re-organized file

```
- - My nose is choked.
  - Okay, this is a symptom of cold, do you have cold?
- - yes i have cold.
  - You should take a sudafed tablet after your meal.
- - Ok, i will take the sudafed tablet.
  - let me know if you are feeling better after having sudafed tablet.
- - I took the sudafed tablet.
  - Good, now take some rest and then let me know if you are feeling better after having sudafed tablet.
- - No, i am still not felling better after having sudafed tablet.
  - Ok, its not just a normal cold, you should consult a doctor.
- - Can u book an appointment for me.
```

Figure 35. Re-organized conversation

5.2 Data Training

After preparing the data, all the prepared data should be trained by the Chatterbot training method.

5.2.1 SpaCy

SpaCy is an open-source python library that is applied in advanced Natural Language Processing. Showing in mid-2014, it is known as "Industrial-Strength Natural Language Processing in Python" and is a Python NLP toolkit with industrial strength. CyCon is heavily used in spaCy to improve the performance of related modules. This is different from the more academic Python NLTK, so it has the practical value of the industry application.

5.2.2 Training

Chatterbot provides a simple way to train data, which also can be rewritten by users. Before the robot training, all the setting should be set already. Because the data we fetched is not suitable for the default training method, so self-designed class inherit from “chatterbot.trainers.Trainer” class is needed. We define our own method to train specific data. The defined trainer is inherited from the root trainer, which defined initialization, sentence analyses, and statement entries to make it readable for Chatterbot. Figure 36 shows the structure of the trainer.

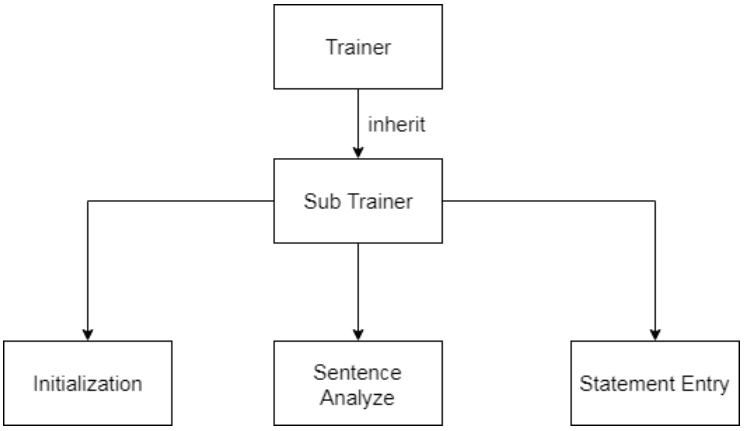

Figure 36. Trainer structure

At first, set the name of Chatterbot, and the language option, in this thesis, we set language as English and set the name “Pepper_ChatBot”. We also need to set the logic adapter. Figure 37 shows the initialization. A logic adapter is one of the most important parts of Chatterbot, Chatterbot provides three default logic adapters which are “Best Match Adapter”, “Time Logic Adapter”, “Mathematical Evaluation Adapter”, and also, you can define by yourself, Figure 37 shows the details of the logic adapter.

```
robot_ip="192.168.1.145"
robot_port=9559
english_bot = ChatBot('Pepper_chatbot',
            storage_adapter='chatterbot.storage.SQLStorageAdapter',
            logic_adapters=[
{
    "import_path": "chatterbot.logic.BestMatch",
    "statement_comparison_function": chatterbot.comparisons.SynsetDistance,
    "response_selection_method": chatterbot.response_selection.get_first_response
},

],
trainer='chatterbot.trainers.ListTrainer')
```

Figure 37. Initialization of a Chatbot

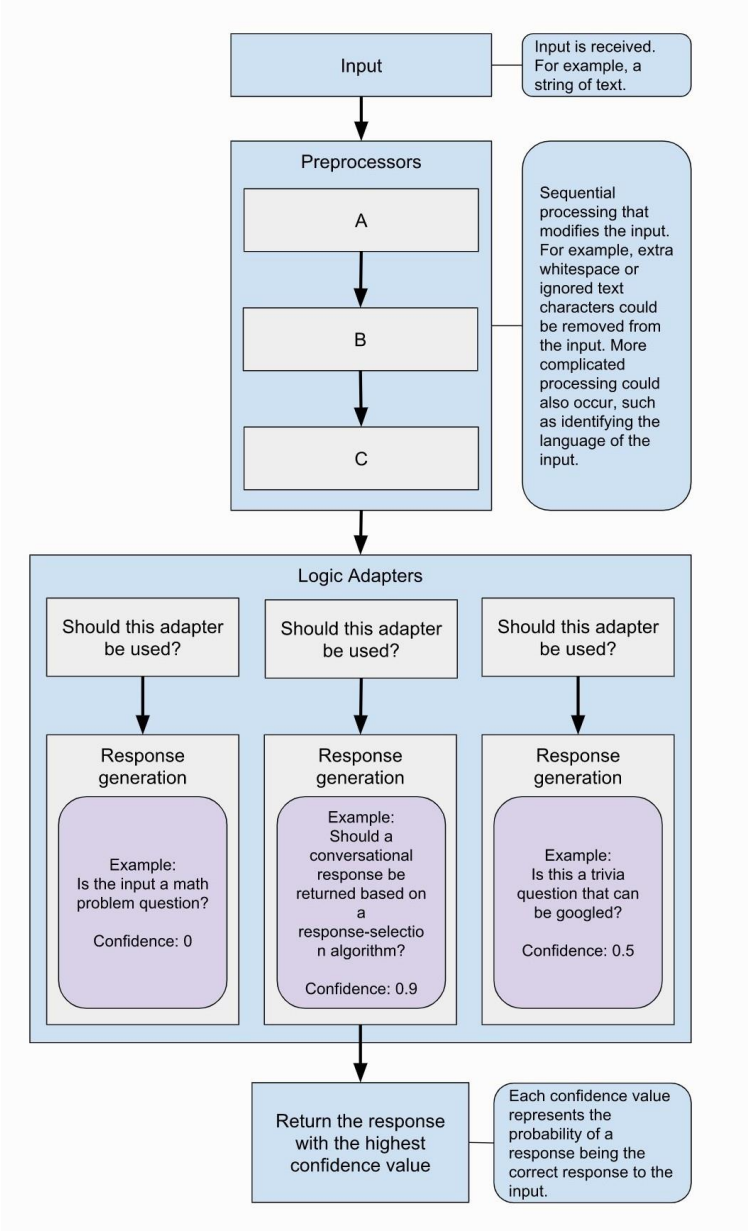

Figure 38. Chatterbot process [23]

All the input will be pre-processed by the "processor," and then, the input will get one of the logic adapters to analyze and process. After initializing a Chatterbot object, the program will transfer all the conversation into "training" method, and in the meantime, all conversation will be token based on spaCy library internal methods. Besides it, we use "pos_" properties to define the part of speech. Finally, we create statement entries and save them. Figure 38 shows the flowchart of this process.

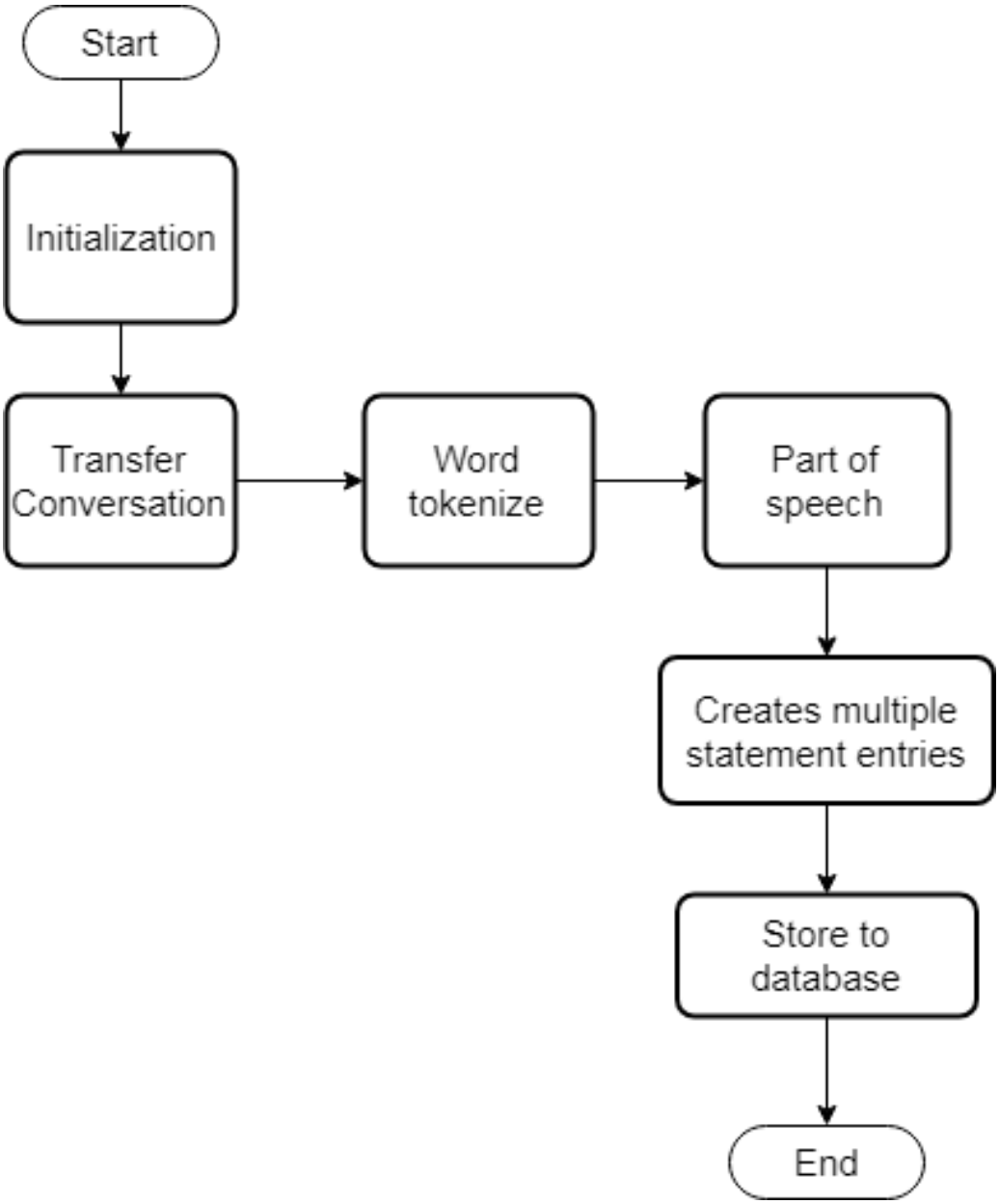

Figure 39. Training process flowchart

Once the message or conversation is input, at first, it will be pre-processed.

The sentence will get their pair string by using “get_bigram_pair_string” method. [21]. For example, the sentence “What a beautiful swamp” will become “DT: beautiful JJ: wetland”

At first, all punctuations will be removed based on the NLTK interface. Then each word of the sentence will be split one by one until they meet “English stop words”. Then all the separate words will be retrieved and tag. Tagging method is imported

from NLTK pos_tag package. Pos_tag package provides a tagger as a part-speech-tagger, which process a sequence of words and attach “part of speech” tag to each word. For example, a sentence “I have a cold”, “I” means pronoun and tag as “PRP”, “have” is a verb tag as “VBP”, and “cold” is a noun, which tag as “NN”.

```
>>> text = word_tokenize("And now for something completely different")
>>> nltk.pos_tag(text)
[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'),
('completely', 'RB'), ('different', 'JJ')]
```

Figure 40. Tagger example [22]

After that, the algorithm will get hypernyms by using the tags array we have got. NLTK wordnet provides an easy to get words’ hypernyms. In simpler terms, a hyponym is in a type-of relationship with its hypernym. For example, pigeon, crow, eagle and seagull are all hyponyms of bird (their hypernym); which, in turn, is a hyponym of animals. [23]

At first, all content stored into “pos_tags” array will be retrieved and converted into the basic unit “Synset”. By calling “synset.hypernyms()” method and we get the hypernyms. Figure 41 shows the code of this method.

```
    hypernyms_results = []

    for word, pos in pos_tags:
        try:
            synsets = wordnet.synsets(word, utils.treebank_to_wordnet(pos), lang=self.language.ISO_639)
        except WordNetError:
            synsets = None
        except LookupError:
            synsets = None

        if synsets:
            synset = synsets[0]
            hypernyms = synset.hypernyms()

            if hypernyms:
                hypernyms_results.append(hypernyms[0].name().split('.')[0])
            else:
                hypernyms_results.append(word)
        else:
            hypernyms_results.append(word)

    return hypernyms_results
```

Figure 41. The code of getting hypernyms

Then, we need to correspond “POS” property stored in “pos_tags” with hypernyms. As Figure 42 shown below.

```
for index in range(1, len(pos_tags)):
    word = pos_tags[index][WORD_INDEX].lower()
    previous_word_pos = pos_tags[index - 1][POS_INDEX]
    if word not in self.get_stopwords() and len(word) > 1:
        bigram = previous_word_pos + ':' + hypernyms[index].lower()
```

Figure 42. Combine POS with hypernyms

After pre-processing the sentence, we need to convert normal sentences into statement type.

As I mentioned above, the conversation is stored in “Statement” type, actually, the statement is Python tuple, there are several attributes in this statement, including “text”, “search_text”, “in_response_to”, “search_in_response_to”, and “conversation”. The conversation property is constant and defined by the user. “Text” store the original text of conversation, “search_text” represents the tagging sentence. “in_response_to” and “search_in_response_to” represent the previous text and previous tagging text separately. Figure 43 shows a normal structure of the statement.

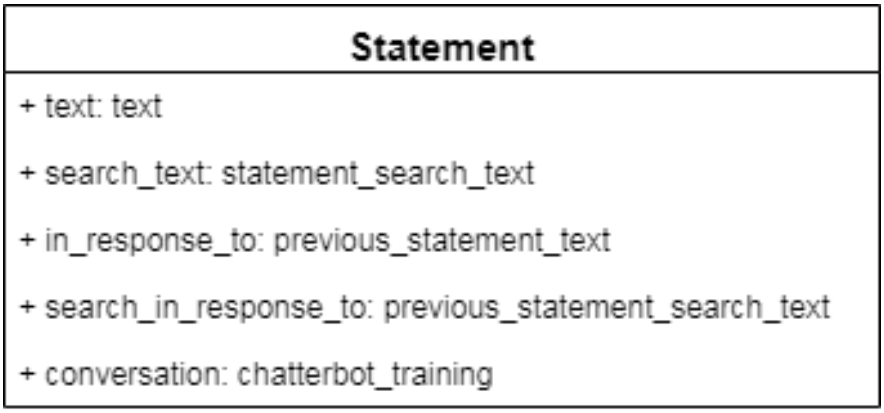

Figure 43. The basic structure of the statement

Figure 44 shows the training process on a Linux machine.

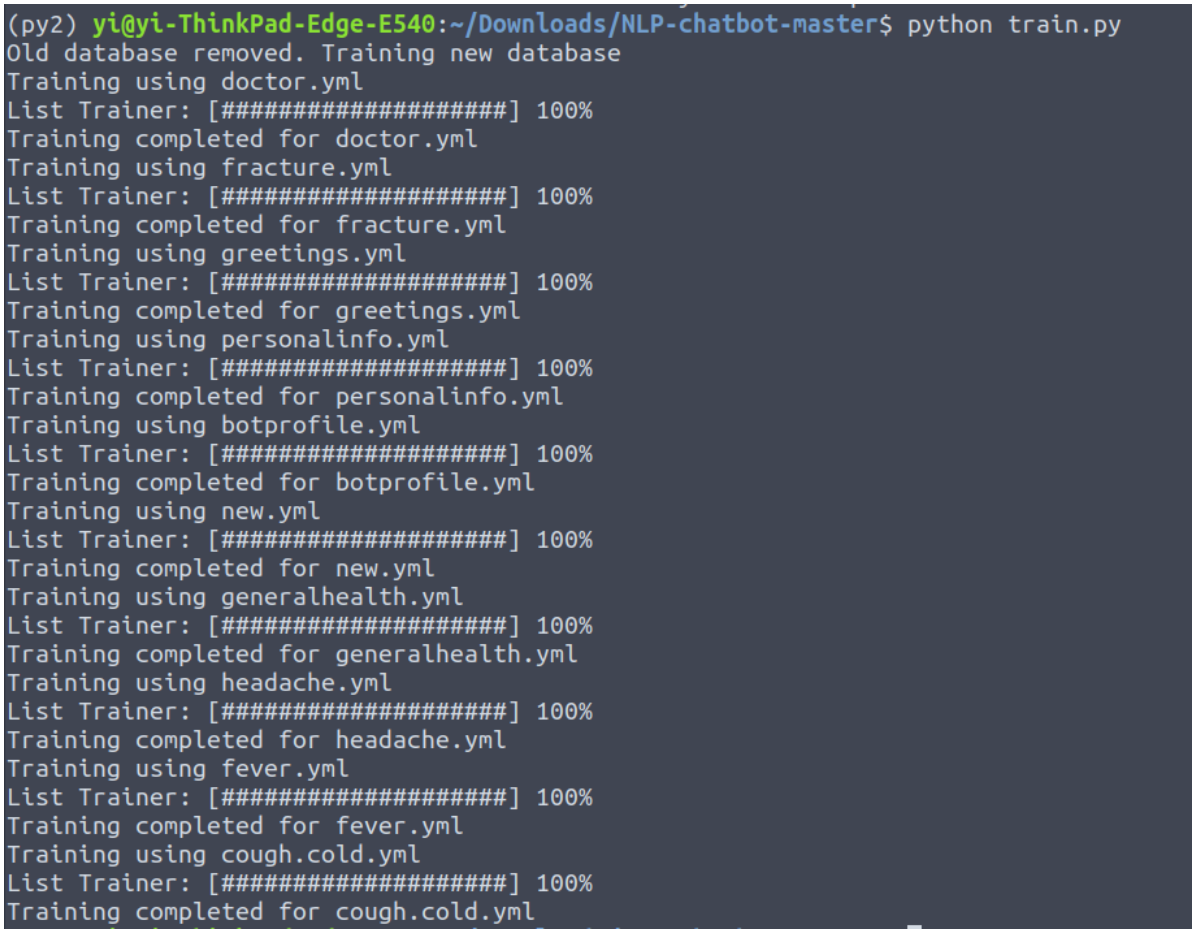

Figure 44. Training process

5.3 Training Result Storage

As defined in the definition phase, the storage adapter is “chatterbot.storage.SQL-StorageAdapter” which means that all the results will be stored in the Chatterbot SQL format file. “SQLStorageAdapter” supports Chatterbot insert data in any database supported by SQL Alchemy ORM. “SQL Alchemy ORM” is a powerful Python ORM tool kit, which provides full-function SQL and ORM operation. SQLAlchemy provides a standard interface that allows developers to create database-agnostic code to communicate with a wide variety of database engines. ORM is also

called Object Relational Mapper. Simply said, ORM is a technique for accomplishing the operation of a relational database through the syntax of instance objects.

- Table → Class
- Record → Object
- Filed → Attribute

ORM uses objects to encapsulate database operations, so you do not access the SQL language. Developers only use object-oriented programming, interacting directly with data objects, without worrying about the underlying database

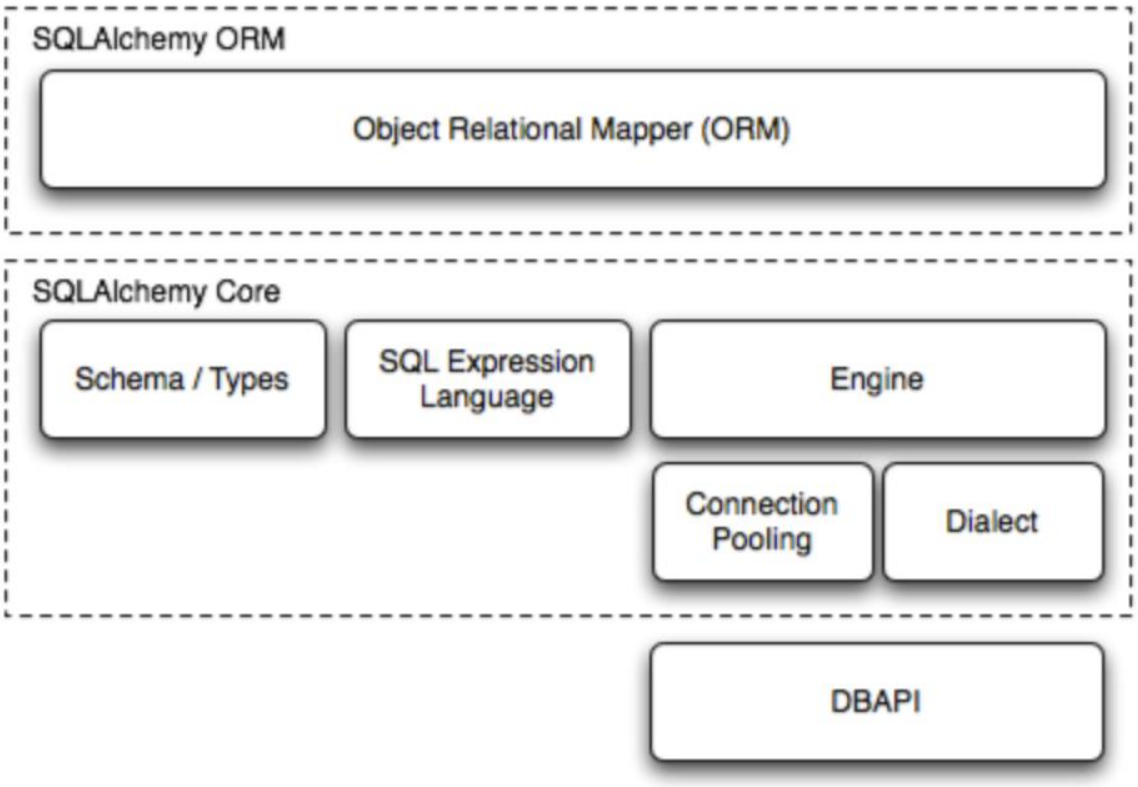

Figure 45. SQLAlchemy structure [24]

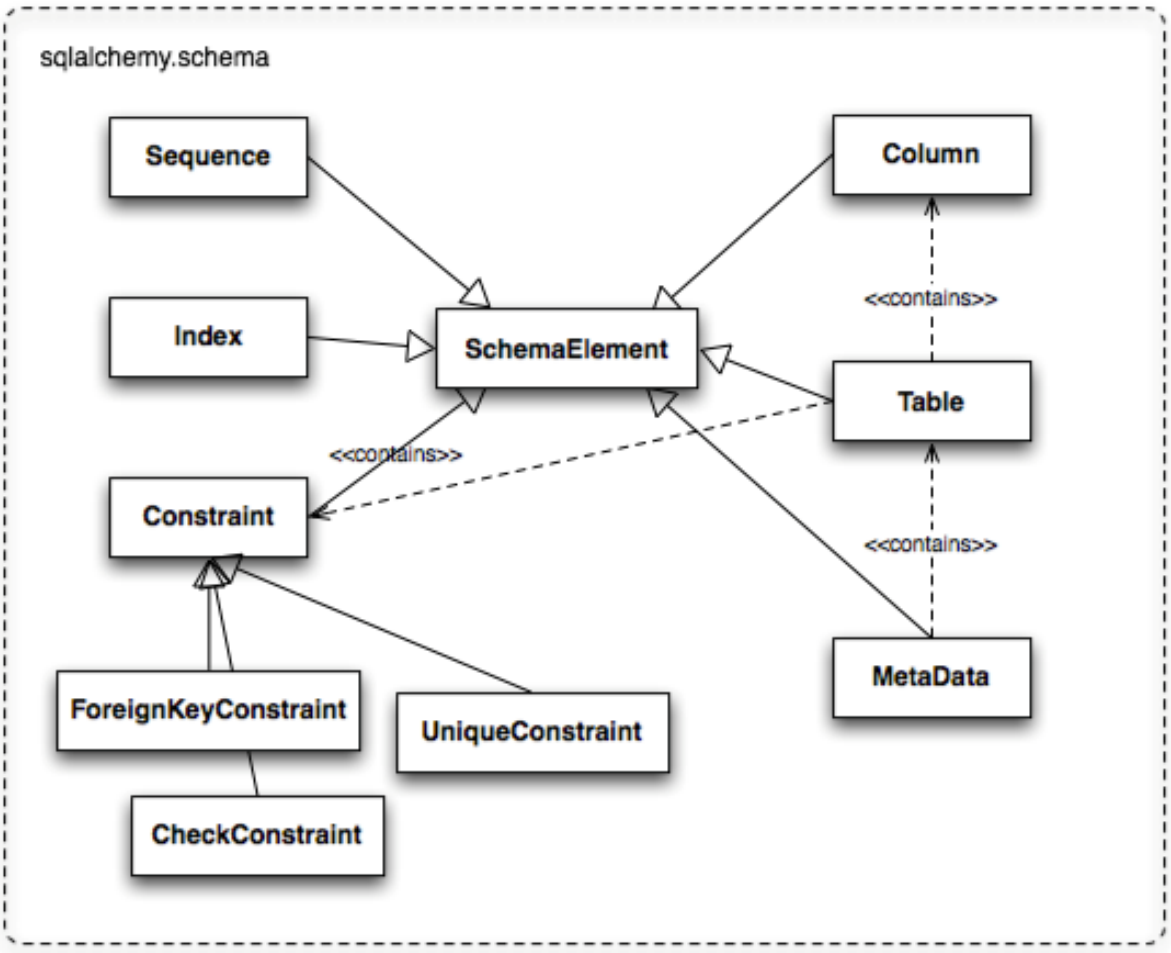

Figure 46. SQLalchemy schema [25]

To store all normal sentences as statements, at first, the Pepper robot will get all attributes of a “Statement” and “Tag”, which means that a container will be prepared. Figure 47 indicates a sample container.

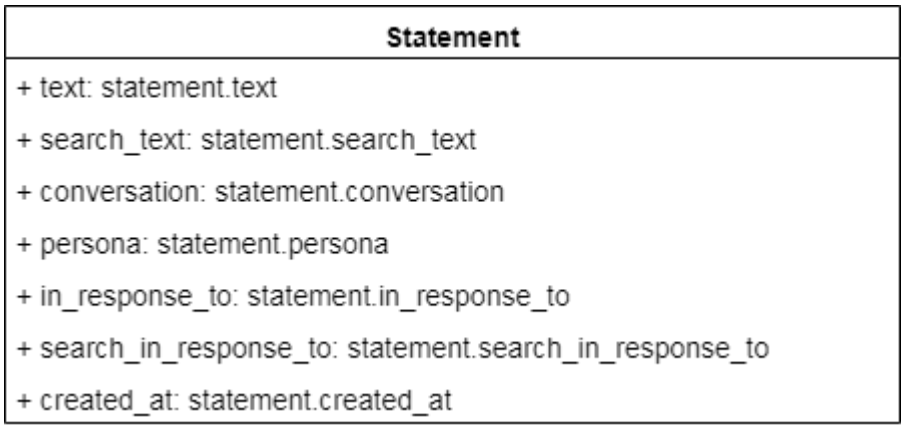

Figure 47. A sample container of stored statement

Now, we copy all properties from the input statement. From Figure 47, we can see that two important parts of a statement, “search_text” and “search_in_response_to”. We need to make sure that these two parts have been already tagged and token by using the same method with the training process.

Finally, all statements we input are prepared and ready to store into the database as Figure 48 shows.

id	text	created_at	occurrence	statement_text
1	- - is a doctor available ?	2019-04-12 12:16:31	1	- sorry, all doctors are occupied at the moment. W…
2	- sorry, all doctors are occupied at the moment. W…	2019-04-12 12:16:31	1	- - hi can i see a doctor ?
3	- - hi can i see a doctor ?	2019-04-12 12:16:31	1	- all doctors are attending patients. we shall revert …
4	- all doctors are attending patients. we shall revert …	2019-04-12 12:16:31	1	- - i want to talk to a doctor
5	- - i want to talk to a doctor	2019-04-12 12:16:32	1	- you shall be called once the doctor is available. D…
6	- you shall be called once the doctor is available. …	2019-04-12 12:16:32	1	- - can you book an apointment with a doctor?
7	- - can you book an apointment with a doctor?	2019-04-12 12:16:32	12	- yes please state time and doctor's name in the fol…
8	- yes please state time and doctor's name in the fo…	2019-04-12 12:16:32	1	- - Patient Name Date Time Doctor Hospital
9	- - Patient Name Date Time Doctor Hospital	2019-04-12 12:16:32	1	- appointment details saved. you shall be contacte…
10	- appointment details saved. you shall be contacte…	2019-04-12 12:16:32	2	- - thanks
11	- - thanks	2019-04-12 12:16:32	1	- happy to help!
12	- - my hand is aching.	2019-04-12 12:16:32	3	- okay, did you get hurt, or it is a suddern aching.
13	- okay, did you get hurt, or it is a suddern aching.	2019-04-12 12:16:32	1	- - my leg is aching.
14	- - my leg is aching.	2019-04-12 12:16:33	2	- okay, did you get hurt, or it is a suddern aching.
15	- okay, did you get hurt, or it is a suddern aching.	2019-04-12 12:16:33	1	- - my joint is aching.
16	- - my joint is aching.	2019-04-12 12:16:33	1	- okay, did you get hurt, or it is a suddern aching.
17	- okay, did you get hurt, or it is a suddern aching.	2019-04-12 12:16:33	2	- - yes i got hurt.
18	- - yes i got hurt.	2019-04-12 12:16:33	2	- ok, is there a swelling or a physical deformity or b…
19	- ok, is there a swelling or a physical deformity or b…	2019-04-12 12:16:33	1	- - it is a mild ache.
20	- - it is a mild ache.	2019-04-12 12:16:33	1	- ok, is there a swelling or a physical deformity or b…
21	- ok, is there a swelling or a physical deformity or b…	2019-04-12 12:16:33	1	- - yes there is a swelling
22	- - yes there is a swelling	2019-04-12 12:16:33	3	- okay, it is a common symptom of fracture, should …

Figure 48. Statement database

5.4 Response

"How to get a response?" is also one of the most important parts of this chatbot. Chatterbot provides a method called "get_response" to return a conversation sentence based on the user's input. After receiving an input statement, the processor will process the sentence as a statement and check if it has attributes and if it is an instance. After the pre-processing, all the statement include "input_statement" and "additional_response_selection_parameters" will be transferred into "generate_response" to generate a response.

At first, a robot will retrieve all the logic adapters we mentioned before, based on each logic adapter rules, the robot tries to process. Actually, the robot will not match the response, all responses belong to the statement, and all statements have their response. The robot only needs to match all possible input statements closely and query all statements from the database based on the designed filter rules. Then get

all possible response statement. Through comparing the result confidence value, get the best match result, as the flowchart Figure 49 shows.

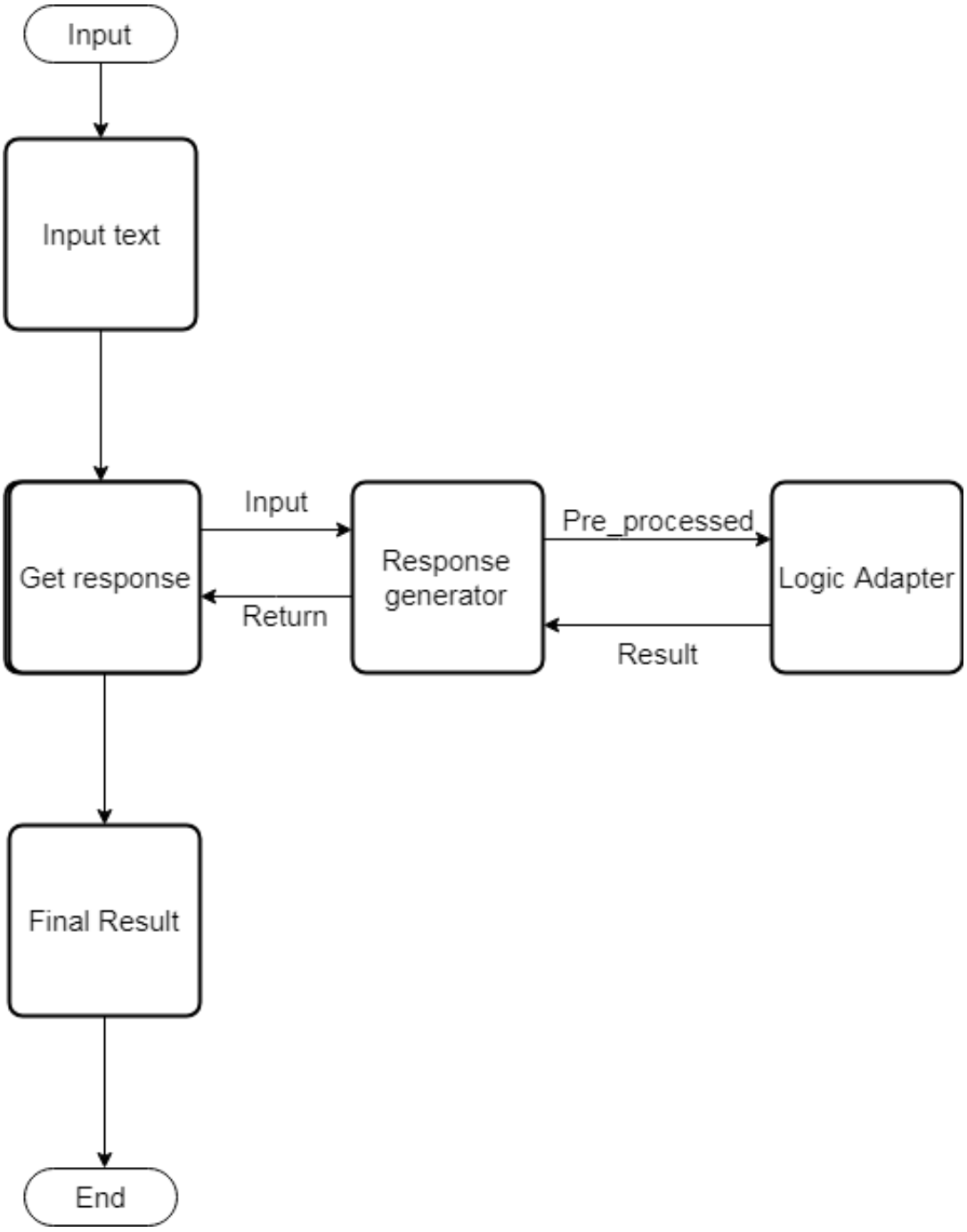

Figure 49. Response process flowchart

However, as I said before, all the sentences in Chatterbot are stored as statement type, so the robot needs to know how to compare statements and select the best response. The first step is searching the database for a known statement that matches or closely matches the input, once the robot selects a suitable response, it will select a known response to the selected match.

In this thesis, we use the"Synset distance" method to compare two statements. "Synset" is a set of synonyms that share a common meaning. "Synset distance" aims to calculate the similarity of two statements. Based on 'wordnet_' and 'NLTK_' to determine the similarity of two statements.

At first, each statement will be token by using nltk.word_tokenize, all stop words will be removed. Then, getting the English language stop words in order to further processing. Figure 50 shows below.

```
tokenStatement=word_tokenize(statement.text.lower())
tokenOtherStatement=word_tokenize(other_statement.text.lower())
tokenStatement = set(tokenStatement) - stop_word_set
tokenOtherStatement = set(tokenOtherStatement) - stop_word_set
```

Figure 50. Word token process

To compare two statements, the max possible similarity is set as the number of words in the longer one. As Figure 51 shows.

```
max_possible_similarity = min(
    len(tokenStatement),
    len(tokenOtherStatement)
) / max(
    len(tokenStatement),
    len(tokenOtherStatement)
)
```

Figure 51. Max possible similarity

Next, getting the highest matching value for each possible combination of Synsets. By using the “path_similarity” method to get similarity between two synsets, the method will return a score denoting how similar two-word senses are [26].

Before comparing the path of similarity, we need to get all possible combination of word of two sentences or statements through Python “itertools”. All the possible combinations will be retrieved and get the similarity. Figure 52 shows this process.

```
for combination in itertools.product(*[tokenStatement, tokenOtherStatement]):

    synset_first = wordnet.synsets(combination[0])
    synset_second = wordnet.synsets(combination[1])

    if synset1 and synset2:


        for synset in itertools.product(*[synset_first, synset_second]):
            similarity = synset[0].path_similarity(synset[1])

            if similarity and (similarity > max_similarity):
                max_similarity = similarity

if max_possible_similarity == 0:
    return 0

return max_similarity / max_possible_similarity
```

Figure 52. Synset distance

After getting all possible similar statements, the robot will filter all possible responses. For normal conversations, we cannot always repeat one sentence, so before generating a response, a recent repeated response need to be discarded. The Chatterbot library provides an internal method called “get_recent_repeated_response” to gain.

6 IMPLEMENTATION ON PEPPER ROBOT

In this thesis, the Pepper robot is the platform that performs all the achievements. As a semi-humanoid robot, the Pepper robot can do better in interaction and conversation. Depending on Naoqi operation system, Pepper can run Python file in terminal windows.

6.1 Pepper Training

Because of the limitation of the Pepper robot, the training process needs a good environment, as mentioned before, all the training results will be pre-prepared in computer and transferred to Pepper robot internal storage. Figure 53 shows a prepared database.

13	- - my hand is aching.	128,2,125,113,1,46
14	- okay, did you get hurt, or it is a suddern aching.	128,2,125,113,1,46
15	- - my leg is aching.	128,2,125,113,1,46
16	- - my joint is aching.	128,2,125,113,1,46
17	- - yes i got hurt.	128,2,125,113,1,46
18	- ok, is there a swelling or a physical deformity or bleeding at the aching region.	128,2,125,113,1,46
19	- - it is a mild ache.	128,2,125,113,1,46
20	- - yes there is a swelling	128,2,125,113,1,46
21	- okay, it is a common symptom of fracture, should I book an an appointment for you.	128,2,125,113,1,46
22	- - yes there is a physical deformity.	128,2,125,113,1,46
23	- - yes it is bleeding.	128,2,125,113,1,46

Figure 53. Conversation Data table

6.2 Speech Recognition

The Pepper robot uses the internal method “ALAudioRecorder” and microphone to record voice as a “.wav” file. Here is the code:

As the code shows above, the robot will start record after print “start” flag, and the sleep time is 5. After recording, the “.wav” file is stored in “/home/nao/nao_medical_record/”. Figure 54 shows in detail.

```
tts=ALProxy("ALTextToSpeech", self.robot_ip, self.robot_port)
record=ALProxy("ALAudioRecorder", self.robot_ip, self.robot_port)
record.stopMicrophonesRecording()
tts.say("start record")
print("start record")
channels=[0,0,1,0]
record.startMicrophonesRecording(self.recordFilePath+'record.wav', "wav", 16000, channels)
time.sleep(5)
record.stopMicrophonesRecording()
print("record over")
```

Figure 54. Recorder module code

By using the “paramiko” library, Pepper will transfer the “.wav” file between computer and the Pepper robot as Figure 55 shows below.

```
robot_ip="192.168.1.145"
robot_port=9559
robot_username="nao"
robot_password="nimdA"
recordFilePath="/home/nao/nao_medical_record/"
saveFilePath="./data/"
def transferFile(self):
    try:
        t = paramiko.Transport(("192.168.1.145", 22))
        t.connect(username="nao", password="nimdA")
        sftp = paramiko.SFTPClient.from_transport(t)
        files=sftp.listdir("/home/nao/nao_medical_record/")
        for f in files:
            print ('')
            print ('##############################################')
            print ('Beginning to download file from %s %s' % ("192.168.1.145",datetime.datetime.now()))

            print ('Downloading file:', os.path.join("/home/nao/nao_medical_record/",f))

            sftp.get(os.path.join("/home/nao/nao_medical_record/",f),os.path.join("./data/",f))
            print ('Download file success %s'%datetime.datetime.now())
            print ('')
            print ('##############################################')
        t.close()
    except Exception as e:
        print (e)
```

Figure 55. Transmission

When getting the record file, the Pepper robot will upload files into Google Cloud by using the Google Speech API. Based on machine learning, deep neural network and Google research's algorithm, Google has greatly improved the accuracy. After importing google.cloud library, the program will open the file as audio. Before uploading the file, Google Cloud Speech API requires a config which defines encoding, sample rate, and language code. Then making a sync request, Google Cloud platform will process the file. Finally, after receiving a response, the library will

process the synchronization recognition response and get the final speech-to-text. Figure 56 shows the code of the process.

```
self.transferFile()
recognizer=sr.Recognizer()
try:
    with sr.WavFile("./data/record.wav") as source:
        audio=recognizer.record(source)
        recognitionText = recognizer.recognize_google(audio)
        return recognitionText
except Exception as e:
    print(e)
```

Figure 56. Voice recognition

After receiving the health condition of the human, the program will analyze and process the sentence by using the module and methods we talked about before. Then the robot will get the response for this input. Then "ALTextToSpeech" will be called to convert the text into the speech signal. The process will be repeated as Figure 57 shows.

```
speech=speechToText()
information = speech.voice_recongination()
print (information)

while information!="end":
    subInfor=information
    if information ==None:
        tts.say("Please say again")
        information=speech.voice_recongination()
    else:
        #
        response=str(english_bot.get_response(subInfor))
        tts.say(response)
        information=speech.voice_recongination()
```

Figure 57. Conversation code

7 RESULT IMPROVEMENT, AND CONCLUSION

7.1 Result

Finally, we got a Pepper robot health medical assistant. After training, the Pepper robot has learned about much medical knowledge. All conversation will be recorded as Figure 57 shows. Figure 58 shows a result capture when running. Users can follow the instruction to ask some basic healthcare question, for instance, “I have a cold”. According to the question user ask, Pepper robot will generate a response like “Does your temperature has raised?”, and then, you can continue to communicate with Pepper robot. A demo video will be shown after clicking the link below.

conversation_id	statement_id
16	155
16	96
17	156
17	33
17	157
17	102
17	158
17	96
17	159
17	132
17	160
17	61

Figure 58. Record data

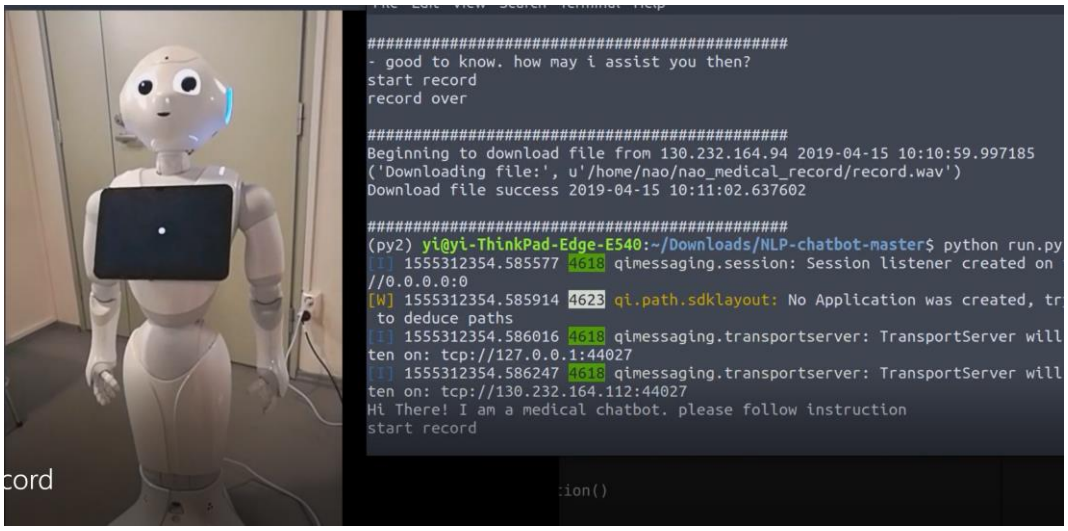

Figure 59. Result capture

Demo Video:

https://www.youtube.com/watch?v=viy2zdSaR-A

7.2 IMPROVEMENT

This project still has many improvement points. First, the data is not sufficient. For AI or Machine learning, the data is one of the most important parts. Now, we only fetch a health website to train, all the fetched data we cannot use directly, all the data should be trained. Although we fetch the whole website data, the data is unorganized and deficient. When Crawler fetches data from the website, but they cannot arrange it, all the data should be processed again in order to make it suitable to be trained. Second, the algorithm is not good enough. No matter training or response, a good algorithm is so important. With a better algorithm, programs can train data faster, and in the meantime, humans will get a better answer. Third, the hardware has limitations. Although we have much data, the limitation of the hardware increases the training time. This thesis uses Google Colab for training, although the performance is very good, the training time is still very long. Fourth, response result. Due to the internal response method and external response algorithm, this study can create a good response based on the question the user asks. However, for

some complex questions and sentences, users may get the wrong results or no response. Fifth, the speech recognition accuracy and speed. Google Cloud provides a good platform to convert voice to text, but because the limitation of the network or Google Speech Recognition engine, the upload speed, and conversion speed is not stable, which cause a delay in conversation.

7.3 CONCLUSION

Health issues have always been a concern for humanity. In some countries, medical conditions are ripe, medical technology is developed, and people can enjoy a better medical environment. But for some resource-poor areas, there are few opportunities for timely health advice and treatment options. But with the development of artificial intelligence, machines can solve some problems through learning. Many countries and regions have invested a lot of resources to develop artificial intelligence. People hope that this alternative way can reduce the burden of human work, or make the working side easier and more efficient.

Human resources are expensive and not always available. In this case, medical robots undoubtedly have a wide range of uses. By making good knowledge in advance and making a preliminary judgment on the patient's condition, the robot can give the patient timely medical advice. For the family, medical expenses will be reduced and the waiting time will be reduced.

This paper introduces the technical background and the process of development. Based on the Chatterbot platform, algorithms and Pepper robots, the robot can respond to problems by training the robot with data acquired by the web crawler. This article details how to design a web crawler to crawl data, how to process data, and how to get the right answer, and finally, how to run on the Pepper robot.

BIBLIOGRAPHY

[1] Shankar, "Web Personalisation with Drupal and Machine Learning," 3 November 2018. [Online]. Available: https://opensenselabs.com/blog/articles/machine-learning-drupal.

[2] D. J. &. J. H. Martin., "Question Answering," *Speech and Language Processing Chapter 23,* p. 1, 2018.

[3] S. Robotics, "Pepper - Developer Guide," SoftBank Robotics, [Online]. Available: http://doc.aldebaran.com/2-4/family/pepper_technical/index_pep.html.

[4] S. Robotics, "Arm Down Technical overview," [Online]. Available: http://doc.aldebaran.com/2-4/family/pepper_technical/index_pep.html.

[5] S. Robotics, "Dimensions," [Online]. Available: http://doc.aldebaran.com/2-4/family/pepper_technical/dimensions_pep.html.

[6] S. Robotics, "Dimensions_Max," [Online]. Available: http://doc.aldebaran.com/2-4/family/pepper_technical/dimensions_pep.html.

[7] S. Robotics, "Motherboard," [Online]. Available: http://doc.aldebaran.com/2-4/family/pepper_technical/motherboard_pep.html.

[8] S. Robotics, "Power entries," [Online]. Available: http://doc.aldebaran.com/2-4/family/pepper_technical/power_entry_pep.html.

[9] S. Robotics, "Tablet Information," [Online]. Available: http://doc.aldebaran.com/2-4/family/pepper_technical/tablet_pep.html.

[10] S. Robotics, "NAOqi Framework," SoftBank Robotics, [Online]. Available: http://doc.aldebaran.com/1-14/dev/naoqi/index.html.

[11] Python. [Online]. Available: https://www.python.org/.

[12] Wikipedia, "Web crawler," [Online]. Available: https://en.wikipedia.org/wiki/Web_crawler.

[13] N. P. HN, "Tensorflow," [Online]. Available: https://towardsdatascience.com/a-beginner-introduction-to-tensorflow-part-1-6d139e038278.

[14] N. Project, "NLTK 3.4.1 documentation," 17 April 2019. [Online]. Available: https://www.nltk.org/.

[15] E. K. a. E. L. Steven Bird, "Natural Language Processing with Python," [Online]. Available: https://www.nltk.org/book/ch02.html.

[16] ChatterBot, "ChatterBot Document," [Online]. Available: https://chatterbot.readthedocs.io/en/stable/.

[17] Chatterbot, "Process flow diagram," [Online]. Available: https://chatterbot.readthedocs.io/en/stable/.

[18] Chatterbot, "Comparisons," [Online]. Available: https://chatterbot.readthedocs.io/en/stable/comparisons.html.

[19] J. Zhang, "What is Speech Recognition," [Online]. Available: https://www.zhihu.com/question/20398418.

[20] E. PROTALINSKI, "Google’s speech recognition technology now has a 4.9% word error rate," 17 May 2017. [Online]. Available: https://venturebeat.com/2017/05/17/googles-speech-recognition-technology-now-has-a-4-9-word-error-rate/.

[21] Chatterbot, "tagging.py Github," [Online]. Available: https://github.com/gunthercox/ChatterBot/blob/3ecccеddd2a14eccaaeff12df7fa68513a464a00/chatterbot/tagging.py.

[22] NLTK, "NLTK--Using a Tagger," [Online]. Available: https://www.nltk.org/book/ch05.html.

[23] Wikipedia, "Hyponymy and hypernymy," [Online]. Available: https://en.wikipedia.org/wiki/Hyponymy_and_hypernymy.

[24] SQLAlchemy, "sqlalchemy structure," [Online]. Available: http://aosabook.org/images/sqlalchemy/.

[25] SQLAlchemy, "sqlalchemy schema," [Online].

[26] "WordNet Interface," [Online]. Available: http://www.nltk.org/howto/wordnet.html.

[27] W3School, "HTML Introduction," [Online]. Available: https://www.w3schools.com/html/html_intro.asp.

[28] W3School, "HTML DOM Tree," [Online]. Available: https://www.w3schools.com/html/html_intro.asp.

[29] Chatterbot, "Logic adapter," [Online]. Available: https://chatterbot.readthedocs.io/en/stable/logic/index.html.

[30] Chatterbot, "About ChatterBot," [Online]. Available: https://chatterbot.readthedocs.io/en/stable/.

[31] Chatterbot, "Chatterbor code," [Online]. Available: https://github.com/gunthercox/ChatterBot/tree/3eccceddd2a14eccaaeff12df7fa68513a464a00/chatterbot.