XIAMEN UNIVERSITY MALAYSIA

Course Code	:	AIT103
Course Name	:	Introduction to Intelligence application
Lecturer	:	Dr. Shaidah binti Jusoh
Academic Session	:	2021/04
Assessment Title	:	Final project
Submission Due Date	:	07/26/2021

Prepared by :

Student ID	Student Name
AIT2009357	Bi Xiaoyang
AIT2009376	Wang Qipeng
AIT2009364	Hu Yaqi
AIT2009380	Zhang Yichi

Date Received : 04/06/2021

Feedback from Lecturer:

Mark:

Med-assistant: An attempt on designing a Seq2seq chatbot coping with COVID-19

Abstract

Chatbots are intelligent agents with which users can engage in conversations using natural language, usually via text or voice. After decades of evolution, highly developed chatbots like Siri, Cortana, and Alexa have left people surprised with their intelligence and capabilities. However, these super-smart chatbots require significant experience and expertise, that is to say, their perfection relies on the long-term collaboration and refinement of a large number of professionals, hard to be replicated. Therefore, in this project, Instead of focusing on refining the daily chat function, our chatbot was developed to focus on answering questions related to precautions and vaccine knowledge of COVID-19. We named it Med-assistant.

Med-assistant is a program written in python and based on the application of artificial intelligence knowledge such as Natural Language Processing and Neural Network. Compared to the earliest chatbots such as ELIZA, our Med-assistant uses a seq2seq based system instead of rule-based ones. Improved on the restricted conversation content and the inability in generating meaningful natural language conversations, it can generate richer, meaningful, and special conversational responses. Moreover, it solves the problem that the more complex the chatbot gets, the more hand-written codes there will be, which enables our vision on putting Med-assistant into application of updating information of COVID-19, as it is changing all the time.

Keywords: Chatbot, Neural Network, Python, Natural Language Processing (NLP), Sequence to sequence model

Med-assistant

Table of Contents

1. Introduction

Sequence to sequence model is a variant of Recurrent Neural Network that consists of Encoder and Decoder components. It is an important model in natural language processing and can be used in machine language translation, text summarization, and chatbot. In our project, we will apply it to building a chatbot.

We took this project as a great opportunity to understand the principle of **Seq2seq model** as well as to realize its application in a chatbot. Other than that, it also gave us a deeper understanding of Chatbot and when we really put our hands on designing one, we had obvious improvement in programming ability, logic, and designing skills. As chatbot being the most promising field in AI, this will be the first step we take in our academic career on the research of chatbot.

The key capabilities that Med-assistant has implemented is giving the best matched responses according to its training, this can basically manage the conversation of daily topics. In addition, we have equipped the chatbot with the ability to update the latest COVID-19 knowledge, so that it can respond to the epidemic-concerned questions and play its role in the struggle with the spreading disease.

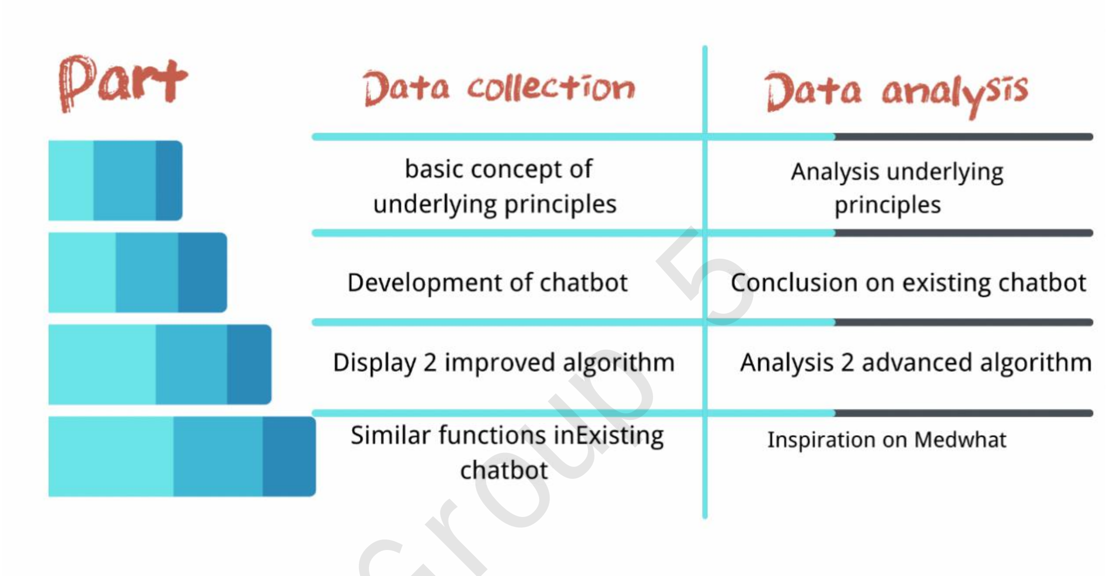
Being under the Covid-19 situation is a tough time, for countries where life is not yet back to normal, this is a good opportunity to let people know that the vaccine plays a crucial role in reducing the transmission rate of COVID-19 worldwide. Based on our research, we want this **Seq2Seq Chatbot** can be used on a large scale in countries where there is a lack of awareness of the COVID-19.

2. Literature Survey

2.1 Data collection

Before we started the design of our own chatbot, we set out to read the existing literature as a way to gain a clearer and deeper understanding of the field. We conducted a brainstorm and decided to start with the understanding of basic concepts and industry or laboratory leading chatbots, and we kept making adjustments to the direction of our collection as we were reading the papers.

Our data collection eventually presented recursively as the four parts of the diagram. The basic concepts contains Recurrent Neural Network and Sequence to Sequence model, they are both learnt at class, but reading papers illustrating them in different ways could provide a refreshing view for our project. For articles concerning the latest development, we got to known the improved algorithms using in Seq2Seq chatbot. We then moved on to learn more about the more advanced algorithms in use: LSTM and Attention Mechanism. And finally, we found an existing chatbot called Med-What, which is focusing on giving medical assistance, it shows the feasibility of our project.



2.2 Data Analysis

There’s a certain reconstruction on all the data so that we may drive the analysis by our logic. At first, we put together all he information about the development of Seq2seq chatbot in both laboratory research and industrial development, after effort in figuring out how these existing chatbots work, we drew a conclusion on their advantages and disadvantages of them. Then, we had a heated discussion on all the principles being applied in our chatbot. We focused on qualitative analysis: **What are them? Why did we choose them? How are we applying them to our own project?** We will refine to more specific questions and give answers in this part, it will lead us to a dialectical understanding of these principles. At last, we will define the scope of our chatbot, and we will analyze the chatbot Med-what, which actually gave us inspiration on our chatbot Med-assistant. The analysis to Med-what reflects on the feasibility on our study, it will also clarify the situations in which our chatbot shows its best ability.

2.2.1 Recurrent Neural Network (RNN)

According to our knowledge, RNN is a powerful and robust type of neural network and is one of the most promising algorithms in use because it is the only one with an internal memory. Because of their internal memory, RNN can remember important things about the input they received, which allows them to be very precise in predicting what's coming next.

Abonia Sojasingarayar made a detailed introduction on the working principle of RNN in *Seq2Seq AI Chatbot with Attention Mechanism.*

2.4 RECURRENT NEURAL NETWORKS

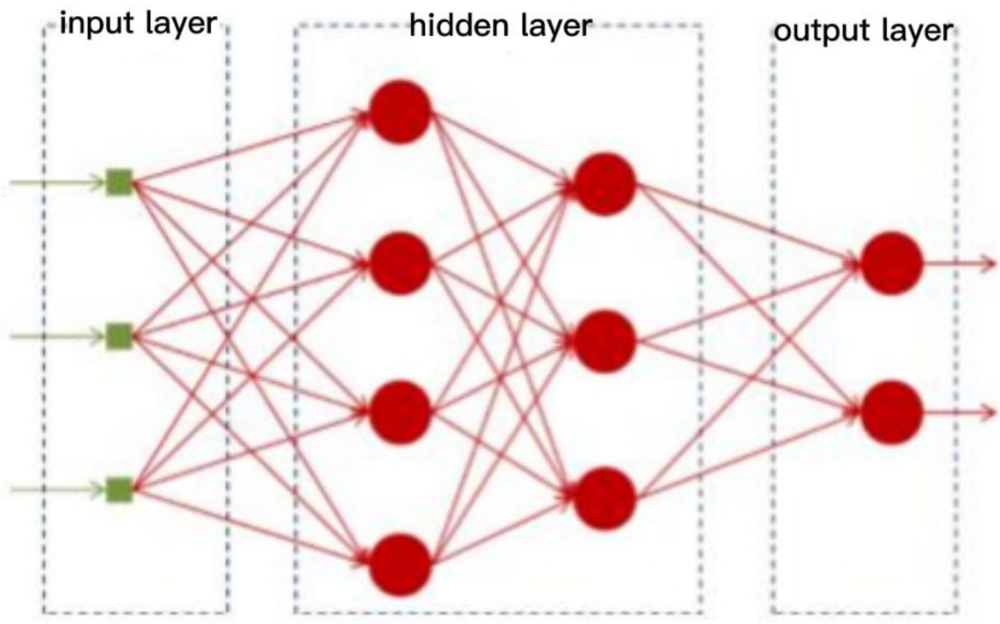
A recurrent neural network (RNN) [Rumelhart et al., 1988] is a neural network that can take as input a variable length sequence x = (x1,...,xn) and produce a sequence of hidden states h = (h1,...,hn), by using recurrence. This is also called the unrolling or unfolding of the network, visualized in Figure 1. At each step the network takes as input xi and hi−1 and generates a hidden state hi. At each step i, the hidden state hi is updated by

$$hi = f(Whi-1 + Uxi) \quad (1)$$

where W and U are matrices containing the weights (parameters) of the network. f is a nonlinear activation function which can be the hyperbolic tangent function for example. The vanilla implementation of an RNN is rarely used, because it suffers from the vanishing gradient problem which makes it very hard to train [Hochreiter, 1998]. Usually long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997]or gated recurrent units(GRU) [Choet al., 2014]are used for the activation function. LSTMs were developed to combat the problem of long-term dependencies that vanilla RNNs face. As the number of steps of the unrolling

As we know, Recurrent neural network is an important branch of neural networks, and our understanding of the importance of recurrent neural networks can start from our knowledge of ordinary neural networks.

A neural network can be treated as a black box capable of fitting any function, and given enough training data, given a specific x, the desired y can be obtained, and the structure diagram is as follows.



After training the neural network model, given an x in the input layer, it is able to get a specific y in the output layer after passing through the network.

RNNs were more specific, they were the first to be used in the field of natural language processing, for example, RNNs can be applied to model language models.

A language model is like: given the first part of a sentence, it predicts what the next word is most likely to be. Language model is of features of a language, and it has many uses. For example, in speech-to-text (STT) applications, the output of an acoustic model is often a number of possible candidates, and then a language model is needed to select the most likely one from these candidates. Of course, it can also be used in image-to-text recognition (OCR). These applications all are based on Recurrent Neural Network. RNN *can take as input a variable length sequence x = (x1,...,xn) and produce a sequence of hidden states h = (h1,...,hn), by using recurrence.*

2.2.2 Seq2Seq model

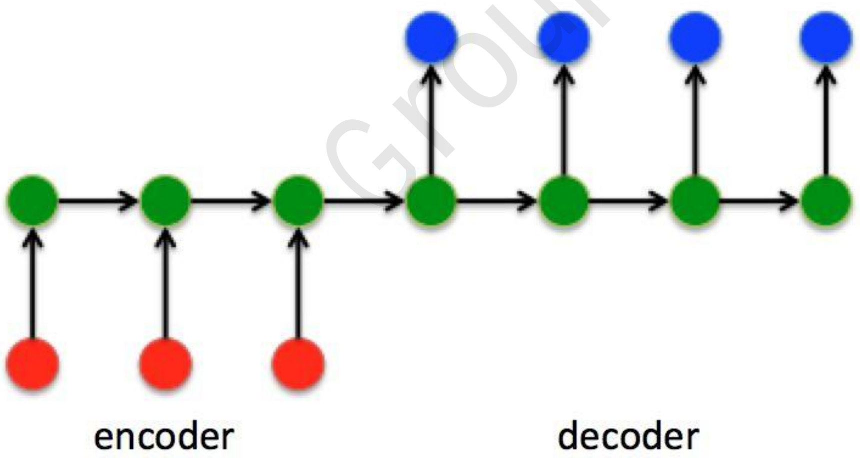
What is Seq2seq model? We read papers about Sequence to Sequence model's application in different fields such as Machine translation, they helped us understand how it works in different aspects.

We quoted from *Seq2Seq AI Chatbot with Attention Mechanism* to illustrate the principle of the model**.**

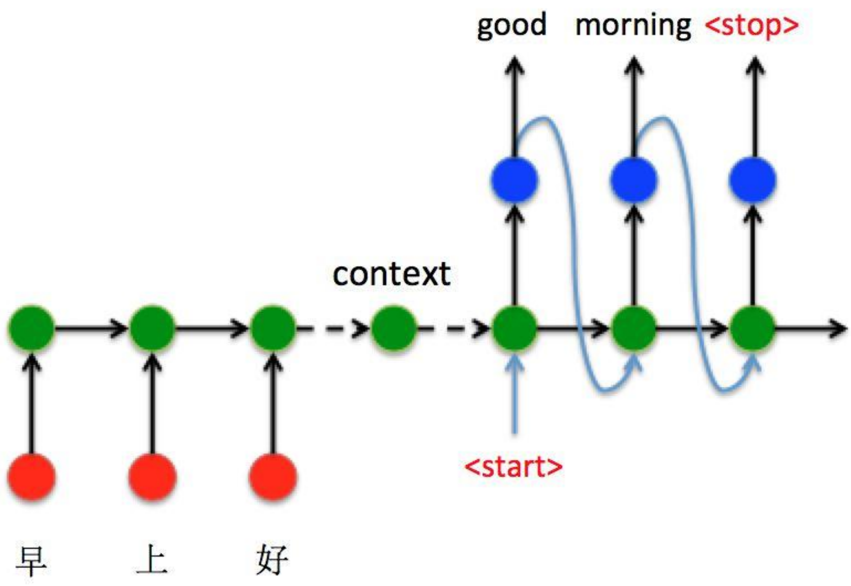
2.5 SEQ2SEQ MODEL

Sequence To Sequence model become the Go-To model for Dialogue Systems and Machine Translation. It consists of two RNNs (Recurrent Neural Network) , an Encoder and a Decoder. The encoder takes a sequence(sentence) as input and processes one symbol(word) at each time step. Its objective is to convert a sequence of symbols into a fixed size feature vector that encodes only the important information in the sequence while losing the unnecessary information. You can visualize data flow in the encoder along the time axis, as the flow of local information from one end of the sequence to another.
Each hidden state influences the next hidden state and the final hidden state can be seen as the summary of the sequence. This state is called the context or thought vector, as it represents the intention of the sequence. From the context, the decoder generates another sequence, one symbol (word) at a time. Here, at each time step, the decoder is influenced by the context and the previously generated symbols.

Compared to normal RNN, Seq2Seq model has a structure no longer requiring the input and output sequences to have the same length. This is actually a great progress, for conversation in natural language hardly have sentences in a same length. The improved structure diagram is as follows.

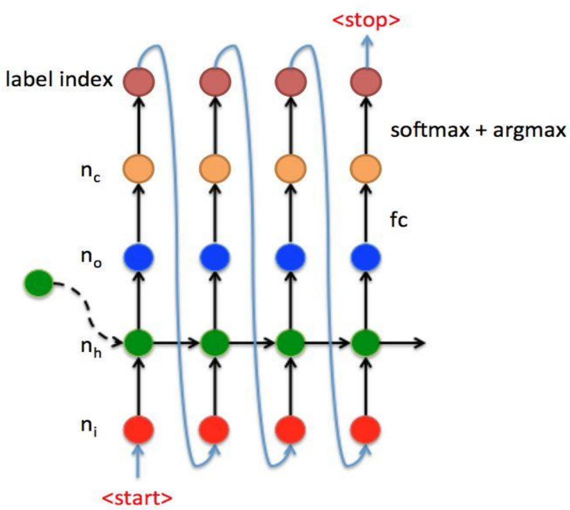


We'd like to take machine translation applying Seq2Seq model as an example.



We can see how to translate the Chinese "Good morning" into English by using the Seq2Seq structure above. Encode "Good morning" by Encoder, and use the hidden layer state t=3 at the last h_3 time as the semantic vector. Decoding is started by using the semantic vector as the h_0 state of Decoder and entering the <start> special identifier at the t=1 moment. After that, the output of the previous moment is continuously decoded as the input of the next moment, and the decoding ends with the direct output of the <stop> special identifier.

The above process is just a classical implementation of the Seq2Seq structure, let's have a look at the t time data stream at the Decoder side of the machine translation example above, as in Figure.

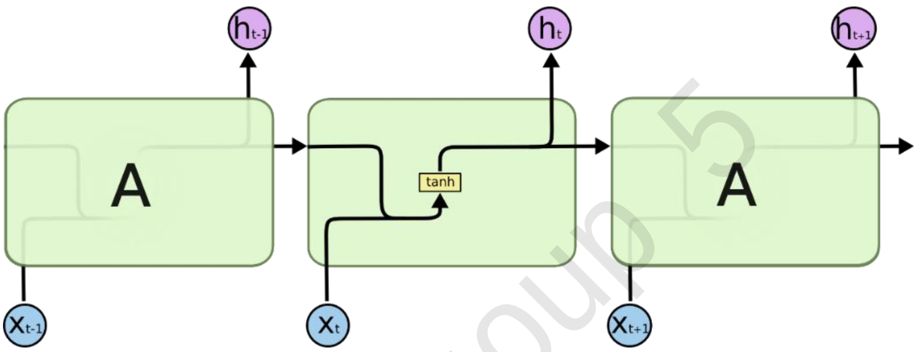


2.2.3 LSTM

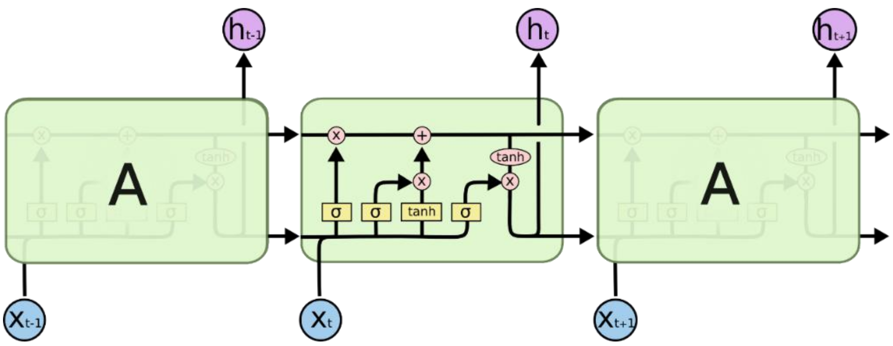
LSTM (long short-term memory) was first invented to solve the problem of vanishing-gradient. It is now frequently used to represent intelligence in language processing. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviors, not something they struggle to learn.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

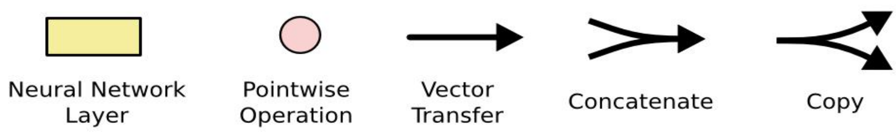
The principle of LSTM is well illustrated in the following.



LSTM has this chain like structure, but the repeating module has a different structure from RNN. Instead of having a single neural network layer, there are four, interacting in a very special way.



The repeating module in an LSTM contains four interacting layers.



In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

2.2.4 Attention Mechanism

Attention Mechanism is a method come up in order to prevent information from being lost when the length of sequence is too long. We understood how it works according to *Seq2Seq AI Chatbot with Attention Mechanism.*

> One of the limitations of seq2seq framework is that the entire information in the input sentence should be encoded into a fixed length vector, **context**. As the length of the sequence gets larger, we start losing considerable amount of information. This is why the basic seq2seq model doesn't work well in decoding large sequences. The attention mechanism, introduced in this paper, Neural Machine Translation by Jointly Learning to Align and Translate [2], allows the decoder to selectively look at the input sequence while decoding. This takes the pressure off the encoder to encode every useful information from the input.

> During each time step in the decoder, instead of using a fixed context (last hidden state of encoder), a distinct context vector ci is used for generating word yi. This context vector ci is basically the weighted sum of hidden states of the encoder.

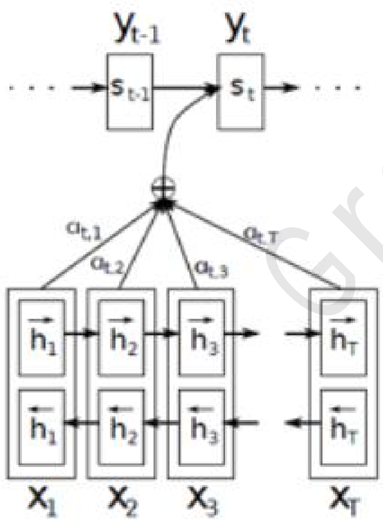
Attention Mechanism is another method come up in order to prevent information from being lost when the length of sequence is too long. It is arguably one of the most powerful concepts in the deep learning field nowadays. It is based on a common-sensical intuition that we "attend to" a certain part when processing a large amount of information.

Before Bahdanau et al proposed the first Attention model in 2015, Seq2Seq chatbot was based on encoder-decoder RNNs/LSTMs. Both encoder and decoder are stacks of LSTM/RNN units. However, RNNs cannot remember longer sentences and sequences due to the vanishing/exploding gradient problem. It can remember the parts which it has just seen. The performance of the encoder-decoder network degrades rapidly as the length of the input sentence increases.

Although an LSTM is supposed to capture the long-range dependency better than the RNN, it tends to become forgetful in specific cases. Another problem is that there is no way to give more importance to some of the input words compared to others while translating the sentence.

Now, let's say, we want to predict the next word in a sentence, and its context is located a few words back. So ***is there any way we can keep all the relevant information in the input sentences intact while creating the context vector?*** Bahdanau et al (2015) came up with simple but elegant idea where they suggested that not only can all the input words be taken into account in the context vector, but relative importance should also be given to each one of them.

So, whenever the proposed model generates a sentence, it searches for a set of positions in the encoder hidden states where the most relevant information is available. This idea is called 'Attention'.



This is the diagram of the Attention model shown in Bahdanau's paper. The Bidirectional LSTM used here generates a sequence of annotations (h_1, h_2,….., h_{Tx}) for each input sentence. All the vectors h_1,h_2.., etc., used in their work are basically the concatenation of forward and backward hidden states in the encoder.

$$h_j = \left[\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top\right]^\top$$

To put it in simple terms, all the vectors $h_1,h_2,h_3….,\ h_{Tx}$ are representations of Tx number of words in the input sentence. In the simple encoder and decoder model, only the last state of the encoder LSTM was used (h_{Tx} in this case) as the context vector.

The improvement by applying are shown in the experiment dataset in LSTM and Simple RNN Comparison In The Problem Of Sequence To Sequence On Conversation Data Using Bahasa Indonesia. And we get it straight. We found that the possibility of getting expected answers became higher after applying LSTM.

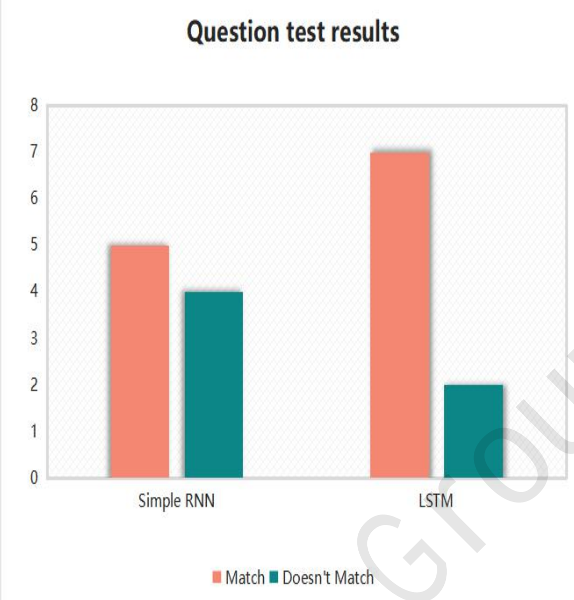


TABLE IV. ONE-PHRASE QUESTIONS TEST

Question	Expected Answer	Simple RNN (Match/ No)	LSTM (Match/No)
Hi	Hi	Match	Match
Good morning	Good morning	Match	Match
Halo	Halo	Match	Match
I am Denny	Hi Denny	Doesn't Match	Doesn't Match
Good morning. I'd like to make a reservation	Good morning	Doesn't Match	Match
I want to booking	What's your name	Doesn't Match	Match
Do you need an address	Need full address	Match	Match
Thank you for your help	Thank you	Match	Match
Are there any seats left	We'll check	Doesn't Match	Doesn't Match

2.2.5 Advantages and disadvantages

The most advanced chatbots are too hard for us to replicate or even understand, effort was made to analyze the advantages and disadvantages as a way to judge the seq2seq chatbot's performance.

We can actually get a glimpse of the reasons for choosing seq2seq chatbot as our project from the advantages of it:

(1) **Capture long-distance dependencies.** Seq2Seq chatbot is more capable of linking up memories than traditional chatbots. The low-level CNN captures the dependencies between words that are closer together, and the RNN captures the

dependencies between words that are farther away. Through the hsierarchical structure, a function similar to high-level CNN to capture the dependencies of sequences with a length of more than 20 words could be realized.

(2) **Various functions based on different models.** For different tasks, adjustments are usually made to the RNN model structure, which is divided into three more common structures depending on the number of inputs and outputs: N vs N, 1 vs N, and N vs 1. For example, N vs N model could be applied in lexical annotation, training language models, using previous words to predict the next word, etc. These variations enables seq2seq model being used flexibly in different functions.

Advantages as there are, Seq2seq Chatbot still has unsolved problems in its development.

Low efficiency. Assuming that a sequence has a length of n, modeling it using RNN (LSTM) requires n operations, and the time complexity is O(n). In contrast, the use of stacked CNN, another kind of Neural network, only requires n/k operations, and the time complexity is O(n/k), where k is the convolution window size.

3 LIMITATIONS OF CHATBOTS

The blunt truth to know is, chatbots are machines, not people. Although you can attempt to equip them with a casual tone, chatbots will never truly sound human.
Chatbots are great at providing facts and data. But they can never truly create an emotional bond with customers. And building an emotional bond can be a make-or-break factor in today's competitive environment.
They often fail in long conversations and have reduced relevancy in dialogue generation. Most of this chatbots are developed for the restricted domain. The majority of them are using simple rule-based techniques. They perform well in question answering sessions and in very structured conversational modes. But, fail to emulate real human conversation and lacks flexibility in functioning.

Failure in long conversation. They often fail in long conversations and have reduced relevancy in dialogue generation. This can be shown well in the following figure.

2.2.6 Existing application

During our data collection, we surprisedly found an existing chatbot with similar properties called Med-what. Med-what is a chatbot actually inspired us to apply our project to share about preventive measures against COVID-19 and about vaccination. Med-what answers medical and health questions for consumers and doctors instantly. It shows the feasibility of our chatbot.

From its official website, we found information we need and summarize it as:

1. its common state of the NLP models today are based on **attention mechanisms**, specifically multi-layer self attention mechanisms. This is the same technique that we will be using and that's what we learnt from it.

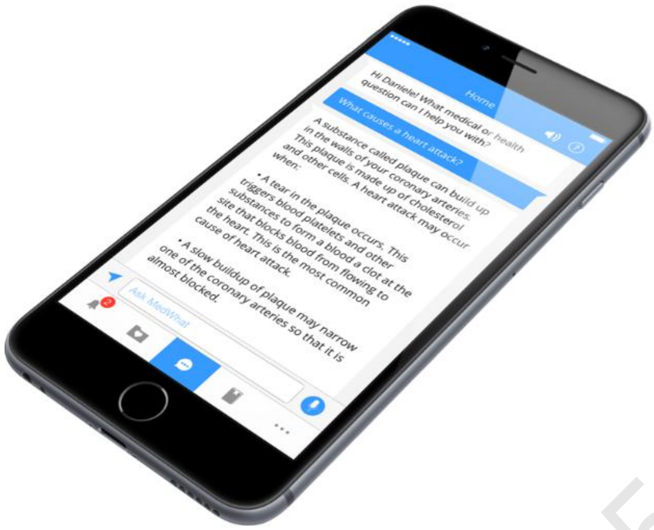
2. MedWhat applies data science techniques to healthcare data stored in 2D medical images, 3D medical images, electronic health records, and wearable devices. And this is a data-processing so we can find and combine our Corpus. We will exclude the Useless data and reorganize all the Corpus we found. Also, we sort and separate them into two types,the first type is daily conversation and second type is about COVID-19.

3. It also has some limitations:

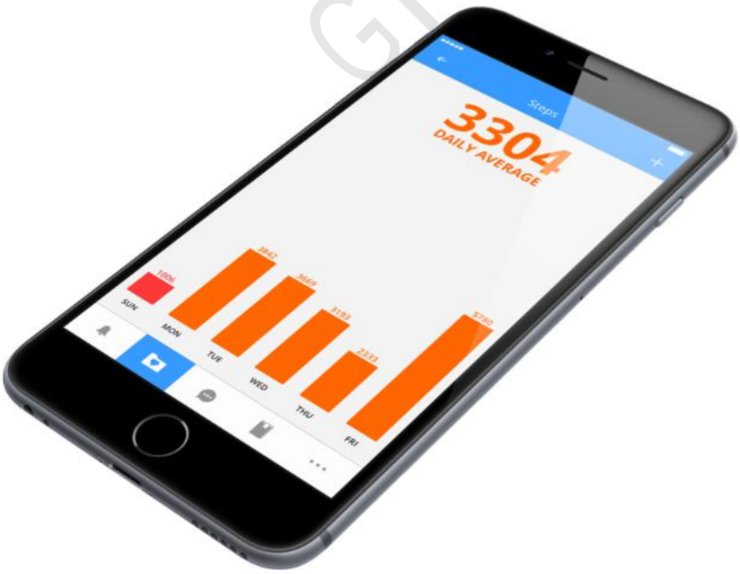
- Large context and summarization are difficult
- Unidirectional training creates limitations
- Models have the biases from the corpora it was trained on

- Large and costly inference

Next, we will have a look the user interface it show us, as in Figure.



We can see the figure, when we ask "What causes a heart attack?", it can response A long list of answers.



This figure is about the change in the number of cases in a country within a week, so maybe we can do like this to make the cases of COVID-19 visualized.

We can see It also have notifications for user. So we also can make notification for user like “What is the nearest hospital to this location?”, “No spicy food or alcohol for a fortnight after vaccination.”.

3. Methodology of the project

During the Data collection and analysis phase, we learn about the existing application. In this part, we will introduce the methodology of project.

Research topic: Our research topic is about Seq2Seq chatbot and how to put our idea into it.

Approach: We mostly use qualitative data, using secondary data collected by others

Research purpose: Our purpose is to build a chatbot as a Q&A system that can generate responses to not only daily dialogues, but also your questions about preventive measures against COVID-19 and about vaccination.

Tools and techniques: We choose Python as our programming language and use the SequenceToSequence (Seq2Seq), is one of our basic methods to realize our goal. It consists of two RNNs (Recurrent Neural Network) , an Encoder and a Decoder. The encoder takes a sequence(sentence) as input and processes one symbol(word) at each time step. In addition, the RNN, LSTM, Attention Mechanism will also be applied in our project.

Running configuration: For IDE, we choose Pycharm 2021.1 to edit it. And we also download some packages such as nltk, numpy, TensorFlow, and some modules from TensorFlow and python standard library.

Dataset: Considering the quantity and quality of corpus, we choose two corpus.The first one is a corpus consists of Movie lines from 50 European and American films used in the daily conversation.The second corpus is 200 long article about the knowledge and precautions of COVID-19 on Twitter.

Strategies： We follow the steps we need to do in instructions. First we choose one topic we interest, then start the data collection and analysis. After preparing the working in existing application, we continue the project in researching our own application.

4. Intelligent system design

4.1 Description of the system

After two weeks ‘s research for the existing Seq2seq Chatbot, we finally sum up our thoughts on how we make our Seq2seq Chatbot into applications and find the direction of what will we research in the next step.And also we have given him the name after a group discussion, named “**Med-Assistant”**.

Med-Assistant will use a technique called deep learning and its model based on a recurrent neural network (RNN) that reads a word from an input sentence one by one and then predicts the output word, which is concatenated to be a sentence. An RNN has a problem of vanishing-gradient so that our designed **Seq2Seq** models use an advanced RNN called long short-term memory (LSTM).

For the domain of our designed **Seq2Seq chatbot**, we think it will be a Q&A system about daily conversation and the knowledge of preventive measures against Covid-19.It means that not only it can chat with you when you are boring as the normal **Seq2Seq Chatbot**, they also can make a respond about the areas of expertise.

The application of **Med-Assistant** can be applied mainly in many developing country where is the population there is not aware of the Covid-19. What's more, it can also been used in the cross-platform centralized messaging. This idea comes from the WHO's official account on WhatsApp.

4.2 Conceptual design of the system/application

4.2.1 Review and Introduction

Through the collection and analysis of existing data and information and looked at a number of papers published in academic journals on the subject and investigated some of the chatbot applications that have been recognised by professionals in the last two years, we have gained new insights into **Seq2Seq Chatbot** . We have found that :

1) Seq2Seq chatbot not only can be used in Q&A system

2) The chatbot has a very promising future under COVID-19 situation because of the widespread use in the Internet .

3) The problem for template-based chatbot is it is time consuming and takes a lot of effort to write the rules manually so making a **Seq2Seq model** can reduce the cost of producing time.

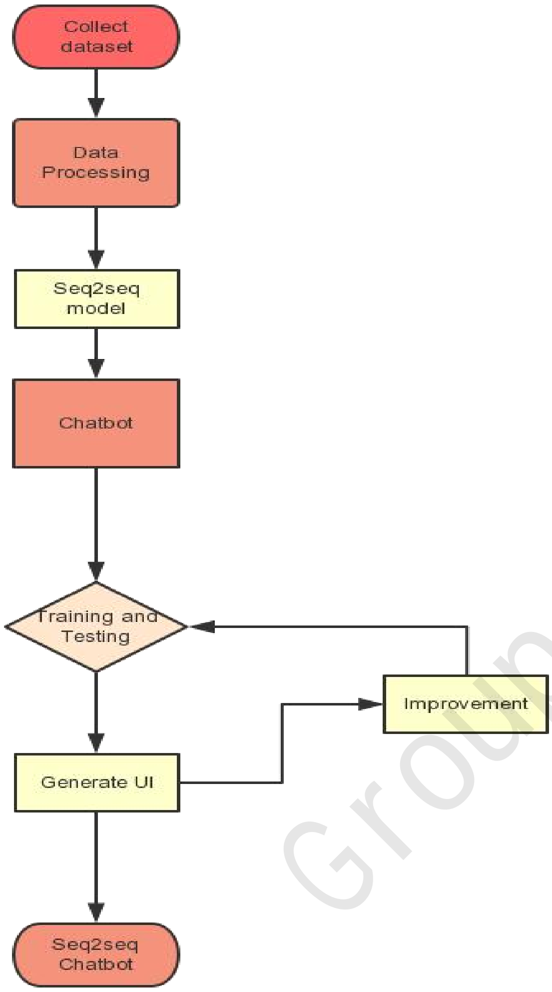
4.2.2 Production programme

(1) Overall production programme

First we found 20 blogs in the Internet about the COVID-19 as our dataset. Then we used **tensorflow** and **RNN** to generate a trainable train.py, then we used a series of training to generate a Seq2Seq model, then we used **natural language processing** and **tensorflow** to create a chatbot, and finally we used **HTML** to create a user interface. In the making process, we need to test it and measure its accuracy and loss. We set when the accuracy is greater than 0.9 then it can be seen as testing successfully.

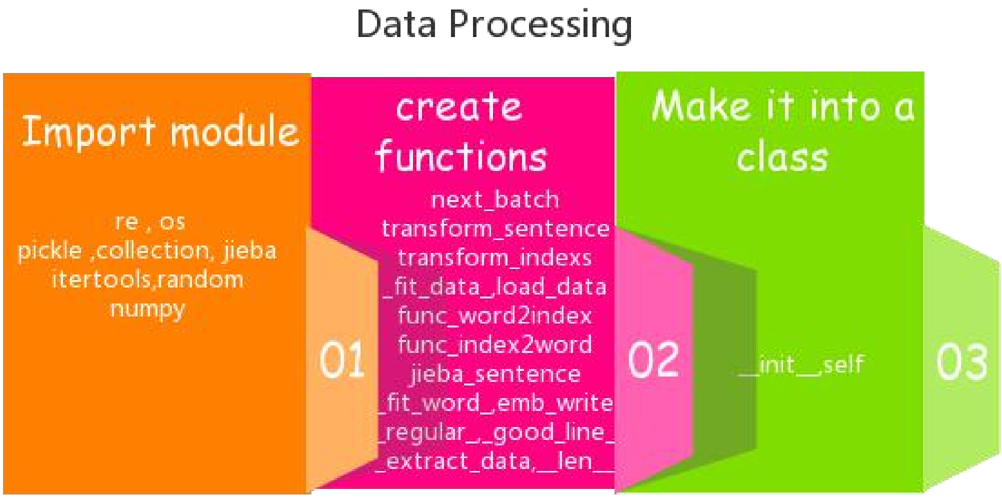
(2) Flow chart of specific production steps

[1] Generally



In the beginning of doing this project, We work in data collection and analysis. After we processing the data, we start to do Seq2Seq Chatbot. First, we build a Seq2Seq model and based on it, we make a chatbot. Then we train and test it, make improvements based on the errors. Finally, we generate an UI to show the functions of chatbot for user.

[2]Data Processing

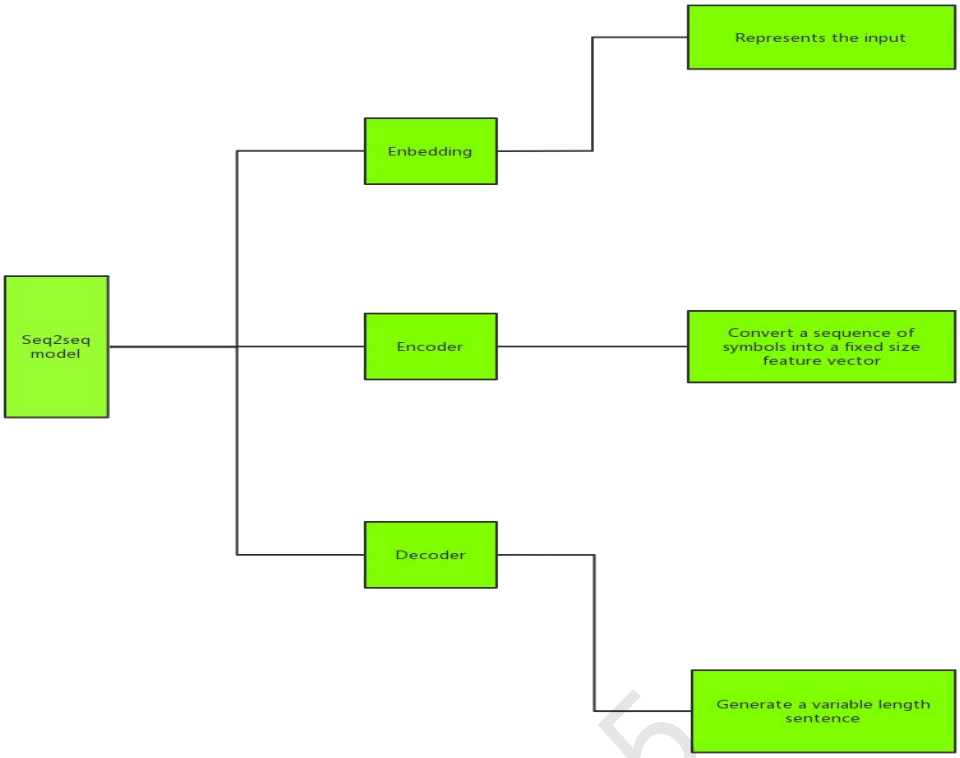


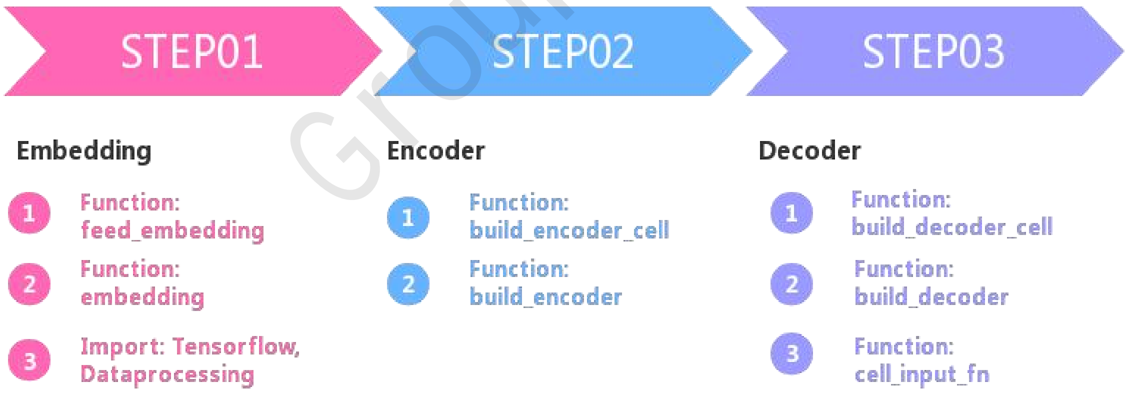
In the data processing we create a file call *DataProcessing.py*, its function is to :

1. Set English character range
2. Replace non-English characters and full-angle characters with blanks
3. Sentences processed with errors for filtering
4. Return the number of question-answer pairs in the processed corpus

And first step to make it is importing module, and then it will have create functions.Then finally, it will be made into a class because it will be used in the *main.py*.

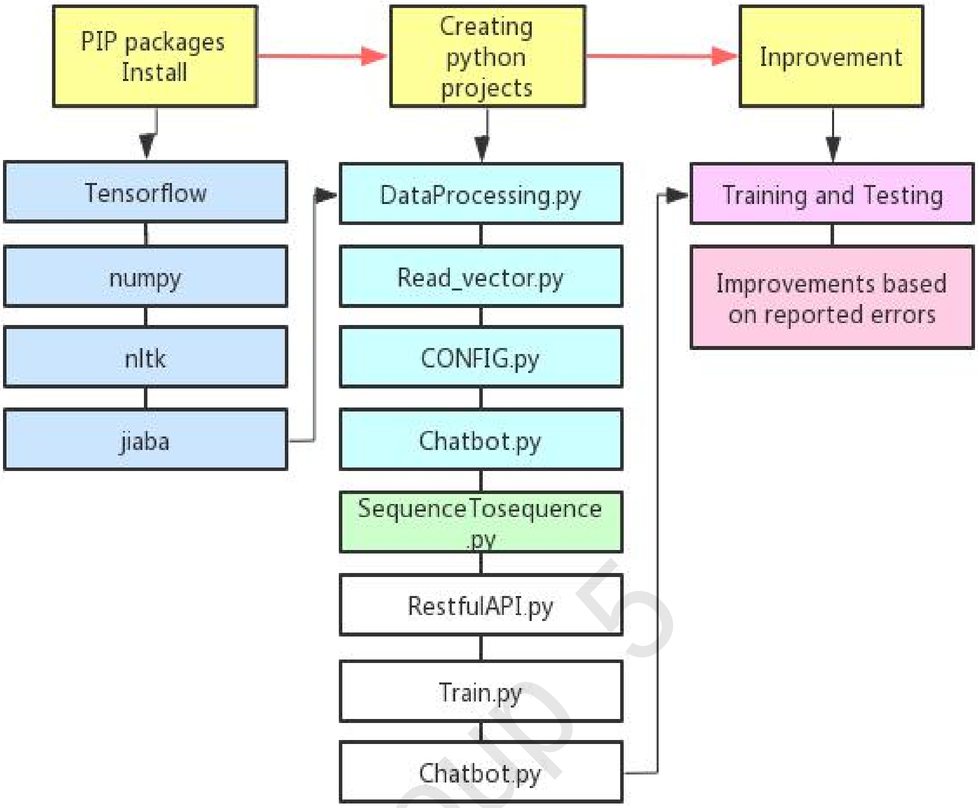
[3]Seq2seq model





We all know the Seq2Seq model have three parts: **Enbedding, Encoder, Decoder**. And what we need in the making Seq2Seq model? In making ***Enbedding***, we need two functions and some important modules such as ***Tensorflow, DataProcessing***. And then we start to build Encoder and Decoder, it is combined by several functions like ***Build_encoder_cell, Build_decoder_cell***.

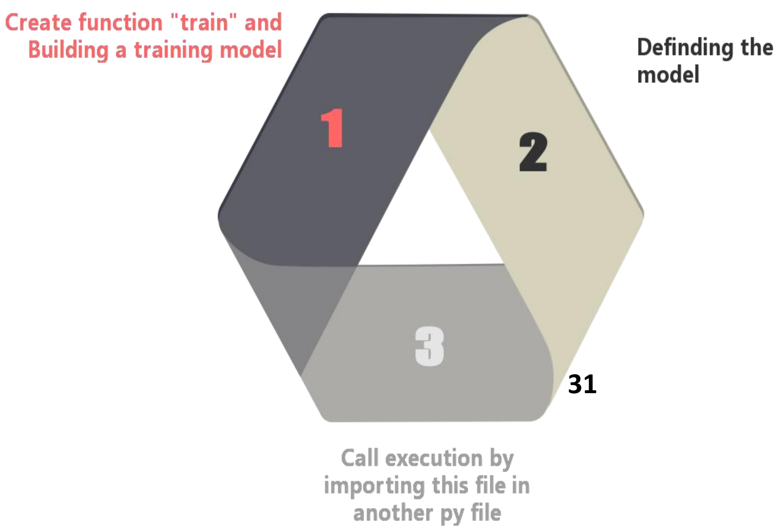
[4] Chatbot



After we construct the Seq2Seq model, we need a chatbot to implement the chatting function. This chatbot is no different from a normal chatbot, We use a number of packages such as ***Tensorflow, numpy and nltk***. And also we create a file to make it function, called *Chatbot.py*. After finishing it, we need to correct the errors it shows in python console and make improvements.

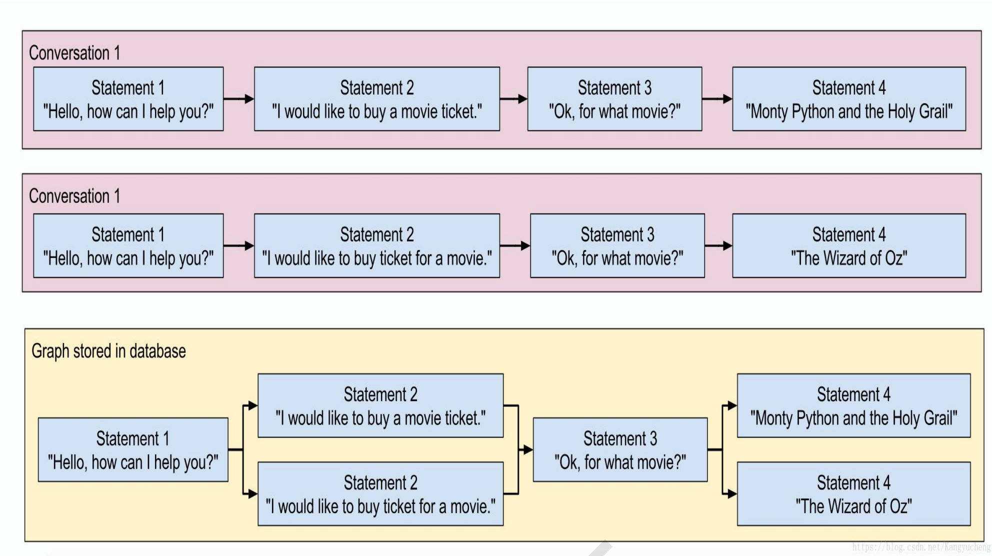
[5] Train

To train this Seq2Seq model, we need to build a file called *Train.py* and in this file, we define a training model that can train the chatbot. And also, it will be made into a class



because it is used in *Sequence2Sequence model.py* file.

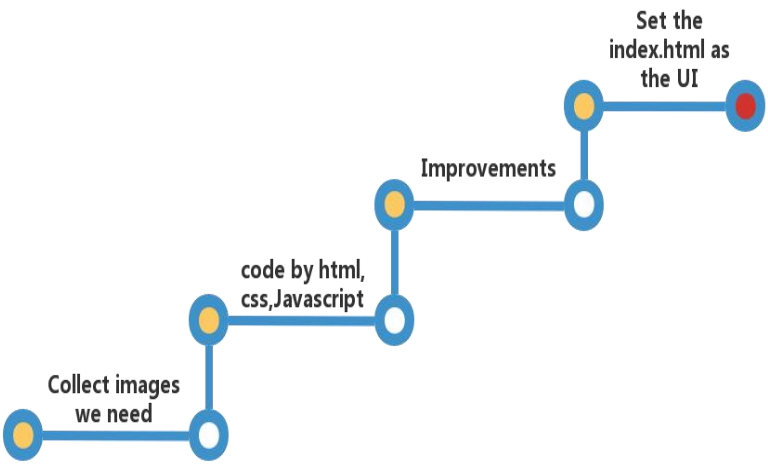
[6] Test



We

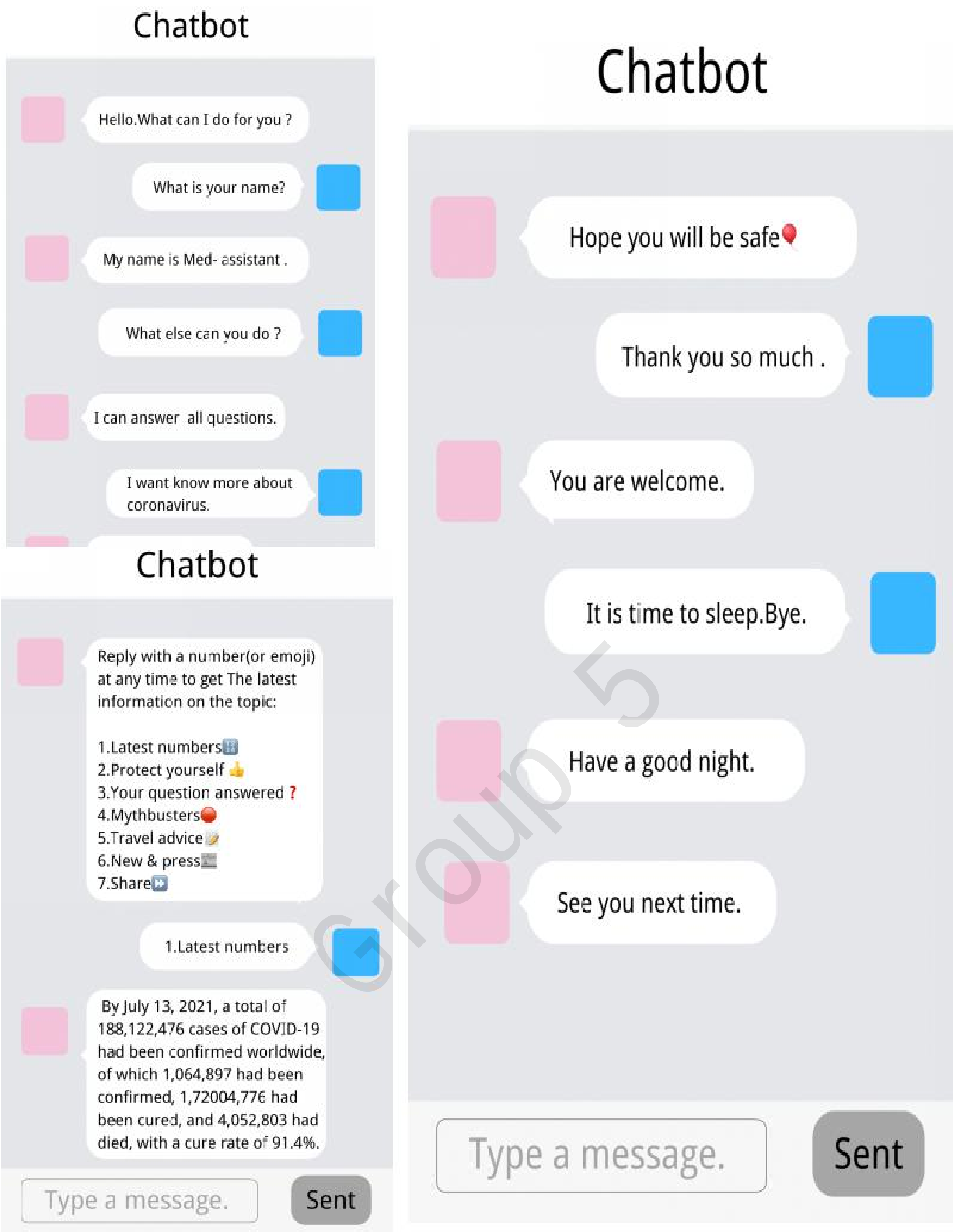
test it by a series of conversation.The first asking sentence is ***"Hello, how can I help you?"*** .The chatbot will answer two different sentence since it train the corpus and the data have been stored in database. Similarily, when asking ***"OK, for what movies?"***, it will randomly answer one of two different sentence stored in database.

[7] UI



To make a user interface, we found a template and we imitate it. First we found some images and then start to code by ***HTML, CSS, Javascript***. Also it still to make improvements when we repeatedly observed the UI presentation.

4.2.3 .Graphical format of expectation(UI)



4.2.4 Source code

Attention mechanism

```
self.attention_mechanism = BahdanauAttention(
    num_units=self.hidden_size,
    memory=encoder_outputs,
    memory_sequence_length=sequence_length
```

Train

```
def train(self, sess, encoder_inputs, encoder_inputs_length,
              decoder_inputs, decoder_inputs_length, keep_prob):
    input_feed = self.check_feeds(encoder_inputs, encoder_inputs_length,
                                      decoder_inputs,     decoder_inputs_length,
keep_prob,
                                      False)
    output_feed = [
        self.update, self.loss,
        self.current_learning_rate
    ]
    _, cost, lr = sess.run(output_feed, input_feed)
    return cost, lr
```


LSTM

```
encoder_outputs = tf.concat(
    (encoder_bw_outputs, encoder_fw_outputs), 2)
encoder_final_state = []
```

Saved model

```
for seed in seeds:
    print("Query >", seed)
    top_n = 3
    for i in range(top_n):
        sentence = inference(seed, top_n)
        print(" >", ' '.join(sentence))

tl.files.save_npz(model_.all_weights, name='model.npz')
```

main.py:

https://drive.google.com/file/d/1WaTYpQDMtMs4-MUjMTLVbt6MW3wKs9pf/view?usp=sharing

Dataprocessing.py:

https://drive.google.com/file/d/19XHcc4qXKQoncCTzbbLqXGLeg_5_qEmn/view?usp=sharing

Mainly processing the corpus, including corpus processing, encoding index, generation of the word vector file EMB of the corpus, etc.

RestfulAPI.py:

https://drive.google.com/file/d/1UcH4F9K3BuUigBOhWfwuXVD0AQBSBf4J/view?usp=sharing

Run this file, and then open index.html to start the man-machine conversation.

CONFIG.py:

https://drive.google.com/file/d/164DBqPxrvGVNZq5x--Sf8ZiedQ2o9qPg/view?usp=sharing

The configuration of model super parameters and related file path is mainly carried out.

Read_vector.py:

https://drive.google.com/file/d/1D9z70-ws17oykKBM8sSiuen2hbHIYlnm/view?usp=sharing

The original word vector is trained by the Wikipedia corpus word2vec. Now we need to modify the original word vector to some extent.

Sequence2Sequence.py:

https://drive.google.com/file/d/1rwrC7XTL4zmXL8w7N8sgAEf_pCObHwgl/view?usp=sharing

Defines model encoder, decoder, optimizer, training, prediction

Train.py:

https://drive.google.com/file/d/1YYsdxbURXZL6foQkazLYl5UcrSg-GBMy/view?usp=sharing

The operation simply needs to run this file.

5. Discussion

In this report, we show that a Seq2Seq chatbot which plays a role as a Q&A chatbot. Importantly, we come up with an idea that it be used on a large scale in countries where there is a lack of awareness of the COVID-19, suggesting it has more than one functionality can be used.

So here, we describe two functions it has:

1) First, no doubt about it can answer the daily questions as a normal chatbot. We want it can answer more than 70% of life's topics so that it can makes the people it is talking to feel as if they are talking to a real person.

2) The second function is that it can generate responses to your questions about preventive measures against COVID-19 and about vaccination. I think that's what makes our Seq2Seq chatbot different from other chatbots, it meets the needs of the current environment and there are not many chatbots on the market with this functionality.

However, it still some problems in training currently:

1) We have trained it two weeks but still can't start the test. we think this is one of the major regrets of our researches.

2) In source code, we think it can be more simpler. We always have a question : Whether the code is too redundant for others to check because in main.py it have nearly 500 lines of code and 7 files in this project.

We also encountered some challenges in this research:

1) At the beginning of the data analysis, our direction was wrong and not in line with

what our teacher asked and instructed us to do, which led us to break the original structure. We encountered many challenges in the reconstruction , such as which data we should choose as our focus, which part of our analysis should we focus on and which field of application we will analyse.

2) We also encountered some challenges when we were building the blueprints, as we hadn't finished building the user interface we couldn't provide the UI we had made but one of us thought we could put the UI on the drawing but then our challenge was what would our ideal UI look like? We thought long and hard about it and after comparing many templates we finally put its blueprint by some graphic techniques.

But the good news is that we managed to overcome all of these challenges and in the process of solving them we gained a different understanding of the seq2seq chatbot and research:

1) Whether a seq2seq chatbot can be successfully built depends on the construction of its seq2seq model.

2) The chatbot has a variety of functions. Different innovative ideas can create different chatbots.

3) Natural language processing is an important branch of artificial intelligence with a wealth of knowledge that deserves to be explored, and we are very interested in it.

6. Conclusion

Chatbot, one of the most promising areas of AI today, has been highly developed, but there are still many problems that are not solved well enough, for instance, the conversations generated by them are more likely to be imitation to the dialogues they read in the training dataset, instead of truly understanding the logic of natural language as human beings. This study, as a rather shallow attempt, clearly does not give a feasible direction for the remaining problems. At the same time, **Med-assistant** has even more limitations: since the underlying code is derived from a Sequence to Sequence robot, it cannot realize the function of contextualization in long conversation. But overall, we have made our effort to design our Med-assistant under the COVID-19 situation, which is the first step that we have taken in our academic career to researching chatbot. More importantly, its application of preventive and control measures and knowledge of vaccines for COVID-19 shows the humanism of our group members and the expectation of a safe and harmonious world in this special time.

7. References

1. Abonia Sojasingarayar, **Seq2Seq AI Chatbot with Attention Mechanism (2020)**

2. Jurgita Kapociute-Dzikiene, **A Domain-Specific Generative Chatbot Trained from Little Data (2020)**

3. Kulothunkan Palasundram, Nurfadhlina Mohd Sharef. Norul Amelina Nashruddin, Khairul Azhar Kasmiran, Azreen Azman, **Sequence to Sequence Model Performance for Education Chatbot (2019)**

4. Yulius Denny Prabowo, Harco Leslie Hendric Spits Warnarrs, Widodo Budiharto, Achmad Imam Kistijantoro, Yaya Heryadi, Lukas, **Lstm And Simple Rnn Comparison In The Problem Of Sequence To Sequence On Conversation Data Using Bahasa Indonesia (2018)**

5. Panitan Muangkammuen, Narong Intiruk, Kanda Runapongsa Saikaew, **Automatic Thai-FAQ Chatbot using RNN-LSTM (2021)**

6. Ankil Shah, Bhargav Jain, Bhavin Agrawal, Saurabh Jain, Simon Shim, **Problem Solving Chatbot for Data Structures (2018)**

7. Yogi Wisesa Chandraa , Suyanto Suyanto, **Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model (2019)**

8. Dan Shewan, **10 of the Most Innovative Chatbots on the web(2021)**

9. Dzmitry Bahdanau, Kyunghyun cho, Yoshua Bengio, **Neural Machine Translation by Jointly Learning to Align and Translate9[Submitted on 1 Sep 2014 (v1), last revised 19 May 2016 (this version, v7)]**

10. Kyunghyun cho, Bart van Merrienboer, Caglare Gulcehre, Dzmitry Bahdanau, Fethi bougares, Holger Schwenk, Yoshua Bengio,

Learning parase representations using RNN Encoder-Decoder for Statiscal Machine Translation[Submitted on 3 Jun 2014 (v1), last revised 3 Sep 2014 (this version, v3)]