1. Design

1.1 Text normalisation and similarity calculation

All normalisation steps aim to reduce the vocabulary size without removing any critical content. The smaller the vocabulary is, the lower is the memory complexity, and the more robustly are the parameters for the words estimated. Text normalisation methods I have used includes:

- **Convert to lower case**: English words are case-sensitive, but the case of most words is the same meaning, so converting to lowercase can reduce the number of tokens.
- **Split into words:** I used the function called *word_tokenize( )* for splitting strings into tokens based on white space and punctuation.
- **Filter out punctuation:** Punctuation like commas and quotes are tokens that I am not interested in. The constant *string.punctuation* provided by Python lists the common punctuation. Therefore, the filter step can be done by iterating over all tokens and only keeping those tokens that are not in *string.punctuation.*
- **Filter out stop words:** Stop words do not contribute to the deeper meaning of phrase. As a result, it makes sense to remove them. I used the commonly agreed upon stop words for English language. Same as the last step, all tokens were iterated and those which are not in stop words list were kept.
- **Lemmatize words:** Lemmatization can reduce each word to its root. There are many Lemmatization methods. Here, I used *WordNetLemmatizer( )* function. As this function's use needs to know the part of speech of each word, I used *nltk.pos_tag( )* function to get the POS first before lemmatizing the word.

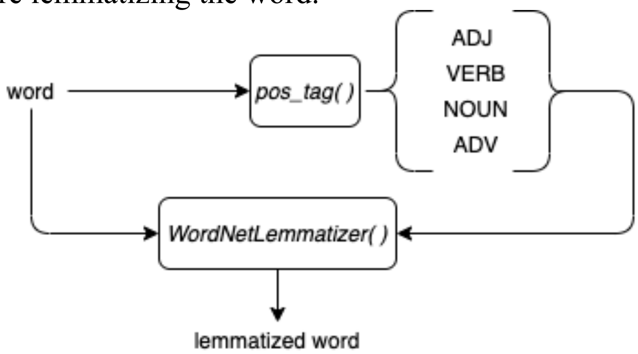

Figure 1. Illustration on how to lemmatize words

- **TF-IDF Approach**
  After the initial pre-processing phase, we need to transform the text into a meaningful vector of numbers. Here, I used the TF-IDF method in text transformation and got two real-valued vectors in vector space.
- **Cosine Similarity**
  Cosine similarity is a measure of similarity between two non-zero vectors. Using this formula, I can find out the similarity between any two documents. This is a crucial step for system to make an appropriate response.

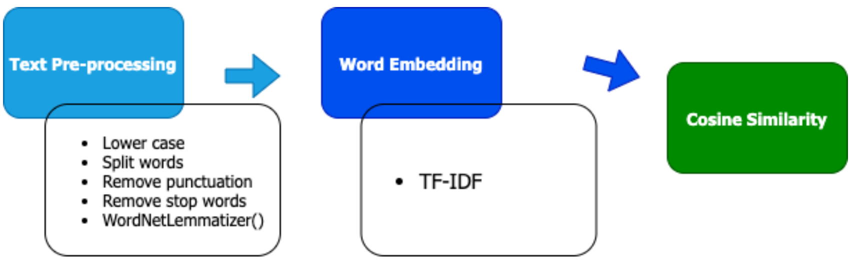

Figure 2. Progress of text normalisation and similarity calculation

1.2 Intent Matching

I wrote a function *SelectTask( )* to judge what the input wants. In this function, the input sentences are compared with a pre-stored file containing small talk and name management keywords, and the cosine similarities of different pairs are returned in the end. It is worth mentioning that the text normalisation methods I used for the Intent matching part do not include stop words filter. Give sentence 'How are you?' as an example; all the tokens 'how', 'are', 'you' are important when determining intent. The value of similarity will be inaccurate if these words are removed by stop words filter. Thereby, the stop words filter is omitted for the intent Matching. In terms of the different spelling, lemmatizer works to handle these variations by converting a word's related form to a common base form. In the end, this function will return a corresponding tag that stands for small talk intent or name intent if the max similarity value is greater than 0.85 (self-defined threshold). Otherwise, the chatbot will compare this input sentence with question answering intent where the new similarities value will be calculated.

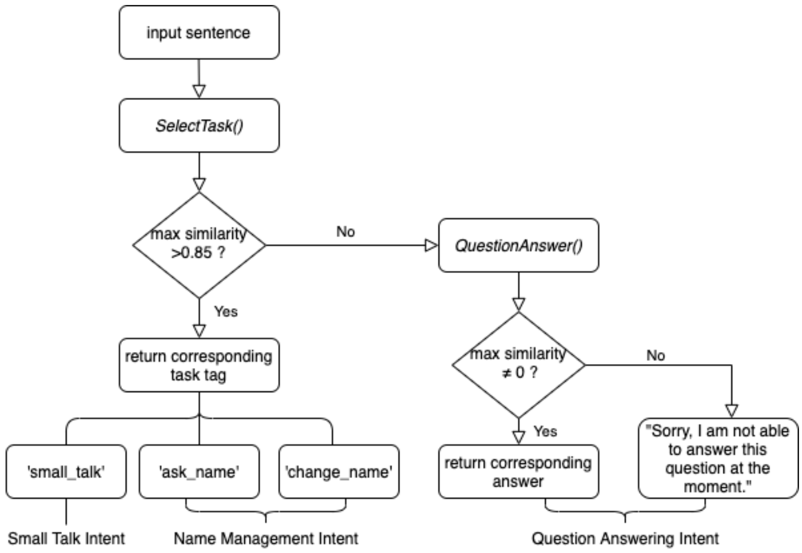

Figure 3. Flowchart of how this chatbot works

1.3 Name Management

This part consists of 2 tasks: store username and return username. After the program starts, the chatbot will first ask the user's name and initialize the global variable representing the username. By comparing the similarities, *SelectTask( )* can judge out whether the queries are related to name management (e.g., "Change my name to"," Do you know my name?"). If it is, the chatbot will then change the username or the answer the username, respectively, based on the tags he gets. For example, suppose the input is 'What is my name?', *SelectTask( )* function should find this is similar with the stored instance 'what is my name' which has tag 'ask_name'(base on max similarity method), and then return this tag. After receiving the tag, the chatbot will then output the stored username. If input "Change my name to X", a 'change_name' tag will get, and the reserved name will be changed based on user's input.

Table 1. Tags to call different name management functions

Sub-tasks	Corresponding Tag	Action
Ask name	ask_name	return the stored user name
Change name	change_name	change the stored user name

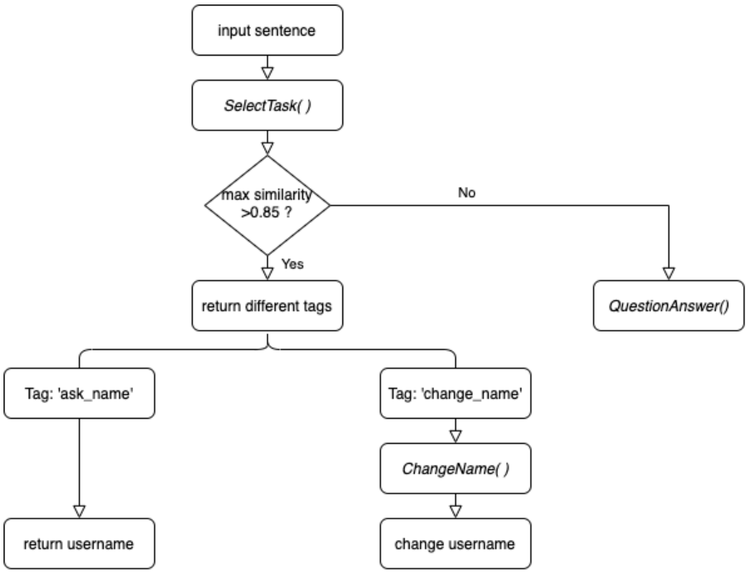

Figure 4. Flowchart of name intent progress

1.4 Small Talk

I divided this part into 3 subtasks: Daily greetings, Thankfulness expression, Time asking. Like name management part, I wrote a function *SmallTalk( )* that accepts the different tags from *SelectTask( )* process and performs corresponding tasks respectively. Suppose the input is 'Thank you!', after text normalisation method, this sentence will be processed into 'thank you' and then compare with a pre-defined text corpus to get text similarity. Finally, a sentence most similar to the input will be detected based on the largest similarity value and its corresponding tag will be returned. Finally, *SmallTalk( )* will accept this tag where a corresponding answer (You are welcome/My pleasure) will be given. It is worth to mention that these tasks can handle small variations in casing and use of punctuation by using the text processing methods mentioned above. Like 'Hi!', 'hi', 'Hi' are all processed into 'hi'.

Table 2. Sub-tasks for small talk intent progress

Sub-tasks	Corresponding Tag	Answer
Daily greetings	small_talk_1	'Hello!','Hi!','I am glad!'
Thankfulness expression	small_talk_2	'You are welcome.','My pleasure'
Time Asking	small_talk_3	local time

```
Huey: My name is Huey.
Huey: I will try my best to answer your question.
Huey: If you want to exit, you can type < bye >.
Huey: By the way, what is your name? My friend.
User:My name is Anthony.
Huey: Hello! Anthony.
Anthony:Hello!
Huey: I am glad! You are talking to me.
Anthony:How are you?
Huey: Hi! I am good!
Anthony:hello
Huey: Hi!
Anthony:Hi!
Huey: Good day, How may i of help?
Anthony:What time is it now?
Huey: Wed Dec  2 23:59:40 2020
Anthony:Thank you!
Huey: My pleasure.
```

Figure 5. More examples of small talk intent

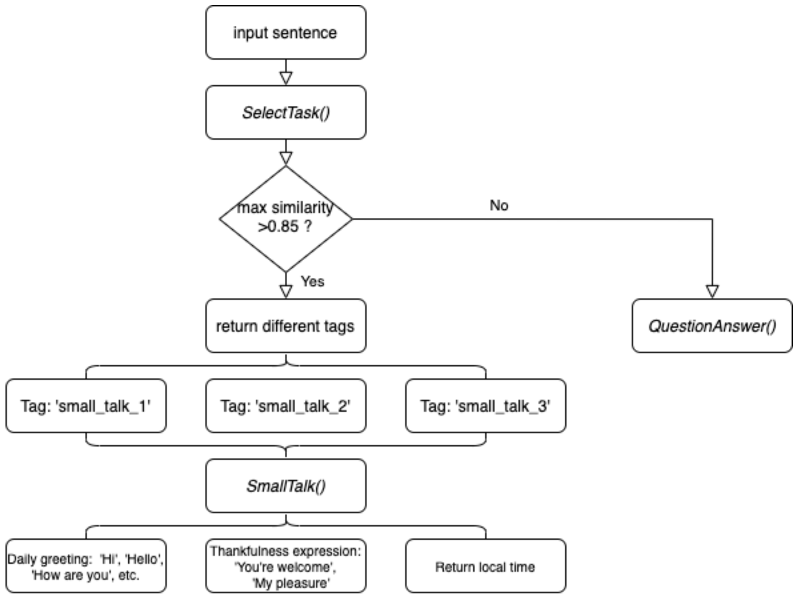

Figure 6. Flowchart of small talk intent progress

1.5 Question Answering

As stated in the previous part, the similarity between input sentence and small talk& name intent related words is the yardstick to determine the chatbot's response. If the similarity value is too low after using *SelectTask( )* function, it means that the content of input is not within the scope of *Name Management* and *Small Talk*, the chatbot will then go to the question answering part and compare the input and question-answer intent file. At this stage, the stop words filter method is included because it is time consuming to use the whole query in similarity calculation. Then, the chatbot will find the index with the highest similarity score and extracts a corresponding answer based on the index. If the similarity of all matches is 0, that is to say, the user query does not match any stored question, it will send an understandable error message: Sorry, I am not able to answer this question at the moment. Here I give an example of how chatbot will reply when facing an unrecognisable question.

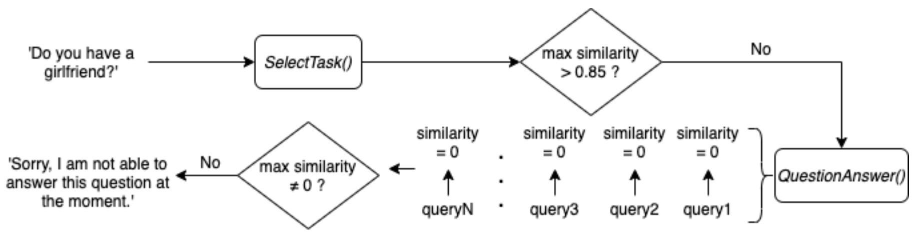

Figure 7. Flowchart of sending an understandable error message

2. Evaluation

Below is the dialogue after using the test queries to run on the system.

```
Huey: My name is Huey.
Huey: I will try my best to answer your question.
Huey: If you want to exit, you can type < bye >.
Huey: By the way, what is your name? My friend.
User:My name is Anthony
Huey: Hello! Anthony.
Anthony:Hello, How are you?
Huey: I am glad! You are talking to me.
Anthony:What's my name?
Huey: Your name is Anthony.
Anthony:Change my name to Handsome
Huey: Hello! Handsome.
Handsome:Thanks!
Huey: You are welcome!
Handsome:How many people play in a Hockey team?
Huey: It started with four teams and, through a series of expansions, contractions, and relocations, the league is no
w composed of 30 active franchises.
Handsome:How mant people live in Atlantis?
Huey: Atlanta (, stressed , locally ) is the capital of and the most populous city in the U.S. state of Georgia , wit
h an estimated 2011 population of 432,427.
Handsome:What are the big ten?
Huey: Its twelve member institutions (which are primarily flagship research universities in their respective states,
well-regarded academically, and with relatively large student enrollment) are located primarily in the Midwest , stre
tching from Nebraska in the west to Penn State in the east.
Handsome:What is single malt scotch?
Huey: Single Malt Scotch is single malt whisky made in Scotland using a pot still distillation process at a single di
stillery , with malted barley as the only grain ingredient.
Handsome:Who is Issac Newton?
Huey: Sir Isaac Newton (25 December 164220 March 1727) was an English physicist and mathematician who is widely regar
ded as one of the most influential scientists of all time and as a key figure in the scientific revolution .
Handsome:What is the best Dim Sum?
Huey: Dim sum () refers to a style of Cantonese food prepared as small bite-sized or individual portions of food trad
itionally served in small steamer baskets or on small plates.
Handsome:What is mustard made of?
Huey: Mustard (or yellow sauce) is a condiment made from the seeds of a mustard plant (white or yellow mustard, Sinap
is hirta ; brown or Indian mustard, Brassica juncea ; or black mustard, B. nigra ).
Handsome:What is 奶茶?
Huey: Sorry, I am not able to answer this question at the moment.
Handsome:Do you have a girlfriend?
Huey: Sorry, I am not able to answer this question at the moment.
Handsome:What time is it now?
Huey: Sun Nov 29 10:32:30 2020
Handsome:Thank you!
Huey: My pleasure.
Handsome:bye
Huey: Bye! Hope that I helped you out!
```

2.1 Functionality

It can be seen from the screenshot that the chatbot gives appropriate answers to different tasks (Name Management, Small Talk, and Question Answering). It also returns an error message ('Sorry, I am not able to answer this question at the moment') to questions not in the corpus.

2.2 Performance

```
User:hello
Huey: Hello!
User:Hello
Huey: I am glad! You are talking to me.
User:Hello!
Huey: Good day, How may i of help?
User:Hi!
Huey: Hello, How can i help?
```

This chatbot is able to handle small variations no matter the use of punctuation, e.g., Hello, hello, hello! Hi! There is also more than one response for small talk and some randomness in the response chosen.

```
User:What is single malt scotch?
Huey: Single Malt Scotch is single malt whisky made in Scotland using a pot still distillation process at a single di
stillery , with malted barley as the only grain ingredient.
User:What is single male scotch?
Huey: Single Malt Scotch is single malt whisky made in Scotland using a pot still distillation process at a single di
stillery , with malted barley as the only grain ingredient.
User:What is signal malt scotch?
Huey: Single Malt Scotch is single malt whisky made in Scotland using a pot still distillation process at a single di
stillery , with malted barley as the only grain ingredient.
User:What is singlemalt scotch?
Huey: Single Malt Scotch is single malt whisky made in Scotland using a pot still distillation process at a single di
stillery , with malted barley as the only grain ingredient.
User:What is signal male?
Huey: Sorry, I am not able to answer this question at the moment.
```

There may be typos and grammatical mistakes in the input. Keywords are essential for TF-IDF methods and similarity judgment. Here, I gave an example above of how the system response if the input contains some typos and grammatical mistakes. It is clear to see that the system can still give the correct answer even if there are several typos or grammatical errors in the input. The following figure gives details about how typos affect system performance.

Sentence 1	Sentence 2	Keywords Matched	Grammatical Mistakes	Similarity
What is single malt scotch?	how is single malt scotch made	3	No	0.9464409436738579
What are single malt scotch?	how is single malt scotch made	3	Yes	0.9464409436738579
What is single male scotch?	how is single malt scotch made	2	No	0.7727657945597155
What is single scotch?	how is single malt scotch made	2	No	0.7727657945597155
What is singlemale scotch?	how is single malt scotch made	1	No	0.5464279336021852
What is single?	how is single malt scotch made	1	No	0.5464279336021852
What is malt?	how is single malt scotch made	1	No	0.5464279336021852
What is scotch?	how is single malt scotch made	1	No	0.5464279336021852
What is signal soctch?	how is single malt scotch made	0	No	0.0

Table 3. The effect of typos and grammatical mistakes on similarity

The system obtains the similarity of input and query list by extracting keywords of them. The more keywords that match, the greater the similarity between sentences. When all keywords are misspelled, the similarity drop to 0. However, some grammatical errors, such as the inappropriate use of 'be verbs', will not affect the text-similarity. Because the 'be verbs' in both sentences (sentence 1, sentence 2) are replaced with their root forms in the text pre-processing stage, leading to a same word and will not contribute to sentence differences.

2.3 Affect

Keeping the script natural can be a huge plus. In order to increase the naturalness of chatting, I reduced repetition and official replies. For example, there will be multiple answers to input 'hi' - hi, hello, hey. Besides, I also give my chatbot a personality by naming him Huey to make him feel like a unique character.

2.4 Shortcomings and how to improve it

Generally speaking, this system is a retrieval-based model can only select a suitable answer from the predefined "answer set". One disadvantage of this mechanism is that it cannot trace the context, such as an object mentioned already, which could lead to misunderstanding. In this situation, the Sequence-to-Sequence architecture in deep learning is very suitable for generating characters. It can learn everything from its data and human-to-human dialogue to generate a wiser response. In addition, to improve this system, I may also code a graphical user interface using the Tkinter library that already comes in python.

In sum, I will use deep learning methods to train chatbot and build system UI to optimize this system.