Technical Design

Summary

This project is using Python to build a chatbot service with RESTful API protocol applied, which could chat with people by text. Moreover, in order to make this service more personalized and, more importantly, fun, a training portal will be provided for the users to train their chatbot to be more characteristic.

Functionality

There are two main services provides, one is chat service, and the other is train service. And I classify the chat sentences into three types: greeting, question, and statement.

1. **Chat service (<root URL>/chat):**
   For the chat service, there are three main processes:
   a. Receive the chat text from RESTful API and classify the text into one of three types: Greeting, Question, and Statement
   b. Parse out the text using TextBlob library, mainly use the Noun Phrases and Sentiment models.
   c. Generate the TextIdentity based on the text type and the analysis results, and retrieve the chat response from the database using this TextIdentity and the send back to the client
2. **Train service (<root URL>/train):**
   For the train service, there are two sub-models, one is to train the chat response, the other is to train the text classification.
   a. **Chat Training Model**
      There definitely will be cases that the IChatBot doesn't know how to respond to the certain chat sentence, but what I'd like to do is to provide this training model to help decrease the time it happens. Also and probably more importantly, I'd like to make this chatbot more personalized by the user and hence it could be more fun.
      So, in order to achieve this, I will store the chat text to which the IChatBot has no idea how to response and when the user comes into the training mode, the user could response to these chat and hence teach the IChatBot in their own ways.
   b. **Classification Training Model**
      This model will use the classifiers model of Textblob library to train the service to be accurate for the three of the chat text. This model may not be released to users since I don't think users should care it.

Technical Description

1. **RESTful APIs:**
   Use Flask RESTful API to build three endpoints:
   a. <root URI>/chat

This endpoint is the main entry for users to chat with IChatBot, and the request format should follow the HTTP protocol

b. <root URI>/train/classification
This endpoint is for admin to train the chatbot how to classify the text sentence into three categories, Greeting, Question and Statement. The training data set will be sent to the service in JSON format.

c. <root URI>/train/chat
This endpoint is for users/admin to train how to respond to a certain text. With this service, chatbot can learn from the user and improve the intelligence to chat. Basically, this service will send to client with a list of the chat sentences that the chatbot don't know how to respond in previous chat, and the service will store the responses from user/admin after.

2. **Textblob Library:**
The Textblob library is a third-party tool to help analyze the text content. There are three main modules that will be used in this project, Noun Phrases, Sentiment, and Classification. The Noun Phrases and Sentiment are used to analyze the chat text and then help to generate the response text. The Classification module is to help train the chatbot to classify the chat text into three different types.
One more thing to mention is that the Textblob library doesn't support the Chinese language, so this project will only work with English for now.

3. **Database**
There are mainly four tables to store all the data in this project, one for each text type, one for overall word frequency, and another one for training data. Below are table structures for these three type of tables:

a. greeting, question, and statement

ID	Text	Noun1	Nount2	Noun3	Response1	Response2	Response3

b. word frequency

ID	Word	Frequency

c. training data

ID	Text	Noun1	Noun2	Noun3

Future Work

This project focuses on the implementation of the server side service and there is no user interface provided in this work. In the future, a client application will be built for users to use with

this service, I'd like to use the API of WeChat or Messenger to make the IChatBot available on those platforms. And I'd like to support the Chinese language in the next work, so other text analyzer libraries will be considered. such as the jieba and NLTK. Another future improvement could lie in the database structure. In this project, I am using a relational table structure, but I do think other structure could potentially improve the overall performance.