Individual Assignment

Chatbot API Implementations

Segundo Parra Jurado #44934693

COSC 310: Software Engineering

Dr. Shan Du

April 15th, 2022

Implemented API's

Wikipedia (Jwiki) API

- https://www.mediawiki.org/wiki/API:Client_code#Java
- https://github.com/fastily/jwiki

Google Translate API

- https://cloud.google.com/translate/docs/reference/libraries/v2/java
- https://cloud.google.com/translate/docs/basic/translate-text-basic

GitHub Repository

- https://github.com/smparra/COSC310IndividualAssignment

Wikipedia (Jwiki) API

Before the implementation of the Wikipedia API, the Chatbot would ask what the user's hobbies were, which ranged from sports to movies. However, these conversations were short and somewhat dull. In the event a user wanted to possibly find out more information regarding one of their hobbies or possibly share a brief summary of their hobby with a friend, I decided that the Wikipedia API would help solve this issue.

The MediaWiki API website had various API implementation methods depending on the programming language. One of the methods linked to a GitHub repo displaying simple yet effective functions used to interact with the Wikipedia API. Within the chatbot code I managed to implement both .exists() as well as .getTextExtract() in order to further improve dialogue. Prior to the implementation of the Wikipedia API, I had to ensure that the following Maven dependencies were added into the .pom file within the chatbot code in order to access certain functionality.

```
<dependency>
    <groupId>io.github.fastily</groupId>
    <artifactId>jwiki</artifactId>
    <version>1.10.0</version>
</dependency>
```

Before being able to use Wikipedia to search for their hobbies, I had to be able to extract the hobby name from within conversation. Therefore, I created the getHobbyText(String s) function.

```java
//Returns hobby, movie, sport in a sentence
public static  String getHobbyText(String s) {

    String word = "";
    word = s.substring(s.lastIndexOf(" ") + 1);

    return word;
}
```

Once the hobby was extracted from the conversation, the chatbot would check whether the hobby had a Wikipedia page using wkpdia.exists(). In the event it didn't, then the Chatbot would return a message stating that there was no additional information that could be found.

```java
// check if a user input includes some hobbies
public static String checkUserHobby(String s) {

    Wiki wkpdia = new Wiki.Builder().build();
    String hobby = getHobbyText(s);
    Scanner sc = new Scanner(System.in);

    String fav = "";
    String response = "";
    int situation = 0;

    for (int i = 0; i < verb.size(); i ++) {
        if (verb.get(i).toLowerCase().equals("like") || verb.get(i).toLowerCase().equals("love")) {

            System.out.println("*Checking to see if there is additional information available regarding: " + hobby + "*");

            if(wkpdia.exists(hobby)) {
                System.out.println("I was able to find more information regarding " + hobby);
                System.out.println("Would you like a brief summary of it to share with others?\nInput either 1 or 2 based off your decision");
                System.out.println("(1): Yes\n(2): No");

                int answer = sc.nextInt();
                if(answer == 1) {
                    System.out.println(wkpdia.getTextExtract(hobby));
                }
            }
            else {
                System.out.println("*There was no additional information regarding: " + hobby + "*");
            }

            response = userLikeHobby();
            situation = 1;
        }
    }
```

If there was additional information to be displayed, the user would be prompted to enter either 1 or 2. If they decided to read a short summary in order to share with others, or simply to gather more information regarding their hobby, then the user would input 1. Using wkpdia.getTextExtract() the API would gather the first paragraph of the Wikipedia page.

Google Translate API

I believed the implementation of a translate feature would be useful within the chatbot, being bilingual I find myself often having to translate certain words or phrases usually from Spanish to English. Therefore, I decided that having a feature where the chatbot could translate text into English would be helpful.

The Google Translate API was implemented using the official Google platform. In order to implement the Google Translate feature, I needed to setup a project within my google account, then add billing and finally generate an API key. Once this was completed, I had to add in the following Maven dependencies in order to allow for the chatbot to access certain features:

```
<dependency>
        <groupId>com.google.cloud</groupId>
        <artifactId>google-cloud-translate</artifactId>
        <version>2.1.12</version>
</dependency>
```

After the dependencies were added, I was able to use .translate(), .getDefaultInstance() and .getService() within the chatbot code. I had to add my API key to my environment prior to coding in order to make sure that Google services would work. I decided to add another line of user input that the chatbot could read:

```
//ask to translate
{"Can you translate for me", "I need a translator", "Can u translate", "Can you translate", "I need a translation", "Please translate"},
//Bot-to-bot communication
```

Once the chatbot recognizes one of these phrases, the chatbot will return the function I created called askTranslation(). The chatbot will then ask for the user to input text, once the text is inputted it will be translated and outputted back to the user.

```
public static  String askTranslation(String s) {
    System.out.println("Sure, I can auto translate any inputted language to English!");

    Scanner sc = new Scanner(System.in);
    String response = sc.nextLine().toLowerCase();

    System.setProperty("GOOGLE_API_KEY", "");
    Translate tLate = TranslateOptions.getDefaultInstance().getService();
    Translation tLation = tLate.translate(response);

    response = tLation.getTranslatedText();
    return response;
}
```

(I've omitted my API key from the screenshot)