

Building an ESL Chatbot

Third Capstone Project Proposal

Robert Walker

The Objective

The goal of this project is to create a chatbot that can carry on a conversation with a user who is learning English in order to help the user practice. The bot should be able to ask questions that elicit responses from the user, and then correct any spelling or grammar mistakes in the user's answer. The bot should also respond to the user's answers and ask follow-up questions in order to continue the conversation.

The Data

The dataset for this project will be created by scraping ESL dialogues from the following website: <https://www.eslfast.com/easydialogs/>, which contains simple conversations on a wide variety of topics. Those dialogues will be used to train the model, however the model also needs a list of starter questions to begin the conversations. Such a list can be found at this website: <http://iteslj.org/questions/getting.html>, which will also be scraped.

Methodology

Web Scraping

As mentioned above, the first stage of this project will be to use web scraping to acquire data needed to train the chatbot. The data will be scraped from the two websites using the requests and BeautifulSoup libraries. First, the requests library is used to make a request to each website to obtain the HTML source code. Once the source code is obtained, it can be parsed with BeautifulSoup. After that the content can be extracted by searching for tags that contain it. For the first website (<https://www.eslfast.com/easydialogs/>), this will be accomplished by writing a function that can be used in a loop to scrape many pages of dialogues in succession. For the second website, there is only one page to scrape, so there is no need to use a loop. In both cases the results will be returned as a list, which will be converted into a pandas series. Afterward, both of the series will be saved as csv files.

Building the Chatbot

The next stage will be to actually build the chatbot. For this project, this will be done using Chatterbot, a Python library designed to create chatbots. This algorithm starts by creating a

chatbot which can be trained on the Chatterbot corpus (which includes multiple languages), as well as other data. This gives the bot basic competency, which improves as the bot communicates with users. Chatterbot produces a response by comparing a request (statement from user) with statements it has seen before, and basically gives a response for the nearest match. The approach is not perfect, as the bot sometimes gives nonsensical responses, but it will work fine for this project.

In order to build the bot, the first step is to install the Chatterbot library. It will also be necessary to install the Chatterbot corpus for training the bot. The next step is to install GingerIt, a Python library for proofreading, which can be used to correct spelling and grammar mistakes in the user's responses. Once the installations are finished, it is time to import the required libraries, as well as the data we are going to use. After that, an instance of the ChatBot() can be created along with certain parameters, including setting the list of questions as the default response when the bot isn't sure what to say (which is often). Then, the bot can be trained on the Chatterbot corpus, as well as the ESL dialogues that were scraped from the web. Now the bot will be ready to use. The easiest way is to create a while loop which begins with a random question.

Deployment

Finally the bot can be deployed on Facebook Messenger or some other platform. This last stage consists of three steps. The first is to create an HTTP server which can listen to messages from Facebook. This can be done using the Flask library. The second step is to create several functions which together handle the HTTP requests and send messages back to Facebook. The third step is to create an app on Facebook to act as the interface for the bot. These steps are not simple, but hopefully they will not pose too great a challenge, and the bot can communicate with users online.