**Chatbot Report**

**Description**

The purpose of this project was to put to use the concepts that we learned in this class. We were given freedom over what kind of application or project we wanted to create as long as it included deep learning and natural language processing like we learned in this class.

**Objective**

The objective was to create a project using NLP. The requirements for this project were for the project to be made using NLP concepts and modeling, using unstructured data, using python deep learning techniques, not using pre-built models, and to ensure that the project scope is not narrow. There were a few suggestions given for projects including a customer support bot, a language identifier, an ML-powered autocomplete feature, a predictive text generator, or a media monitor. The possibilities were pretty much endless, but for the purpose of this project, the chatbot was chosen as it seemed the most interesting and fun.

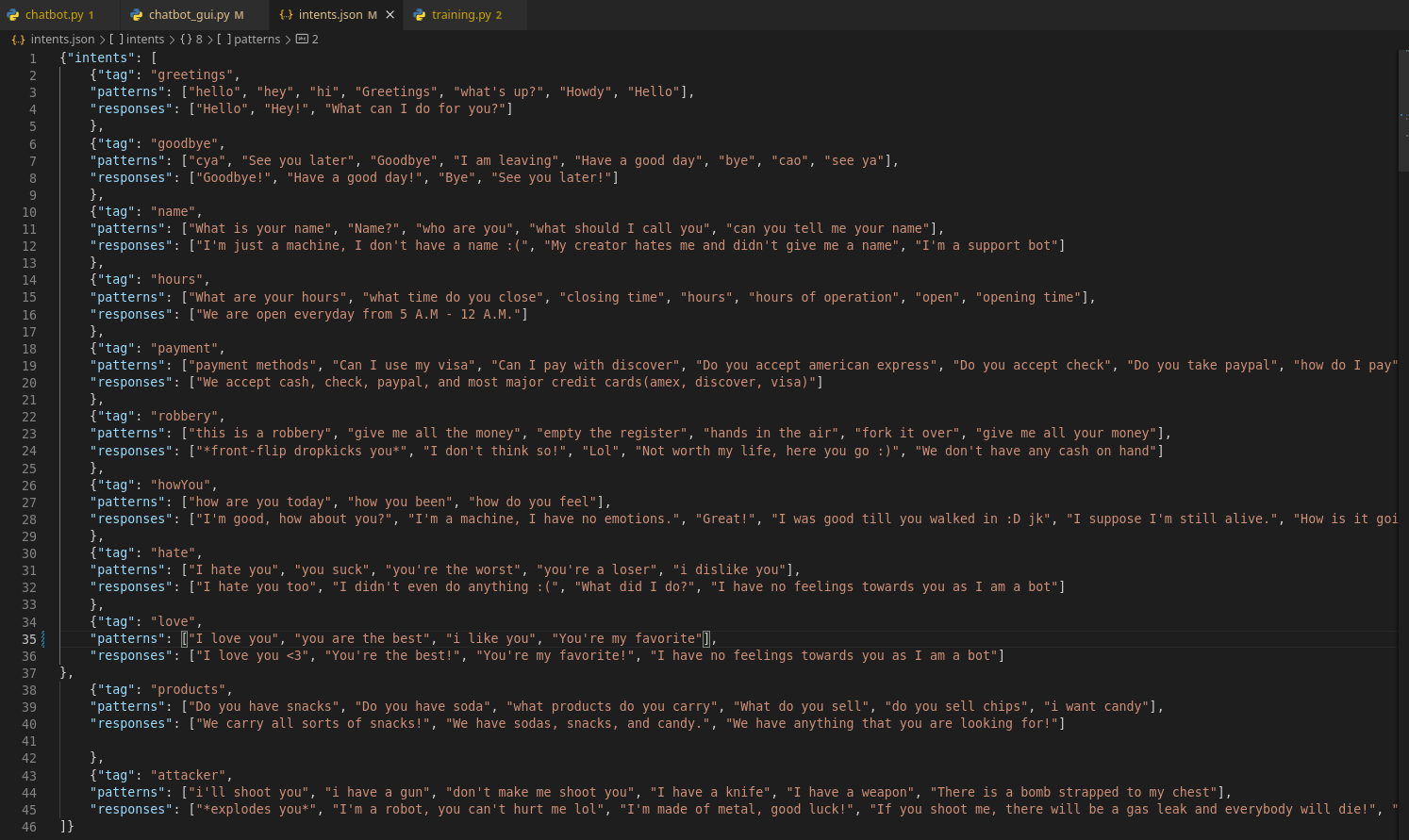
**Design/Implementation**

**a. Brief Overview**

It was a little tough to get started on the bot especially to know where to start since this was a new concept for us so we started off by doing some research and figuring out the basics to building a smart chatbot. It wasn’t hard to figure out what exactly was needed in order to make a simple ai powered chatbot since this is a project which has been completed time and time again. The basic things needed for a smart chatbot were data to train the bot on, a neural network model to train the bot, and then some functions for the bot to take in the user’s input and interpret what the user is saying.

**b. Training Data**

The training data was fairly simple, essentially it’s just a small database where the tag is what the pattern is identified as, the pattern are things that users may say that the bot will be trained on, and the responses are responses that the bot will give if the neural network determines that the user’s input falls under that tag.

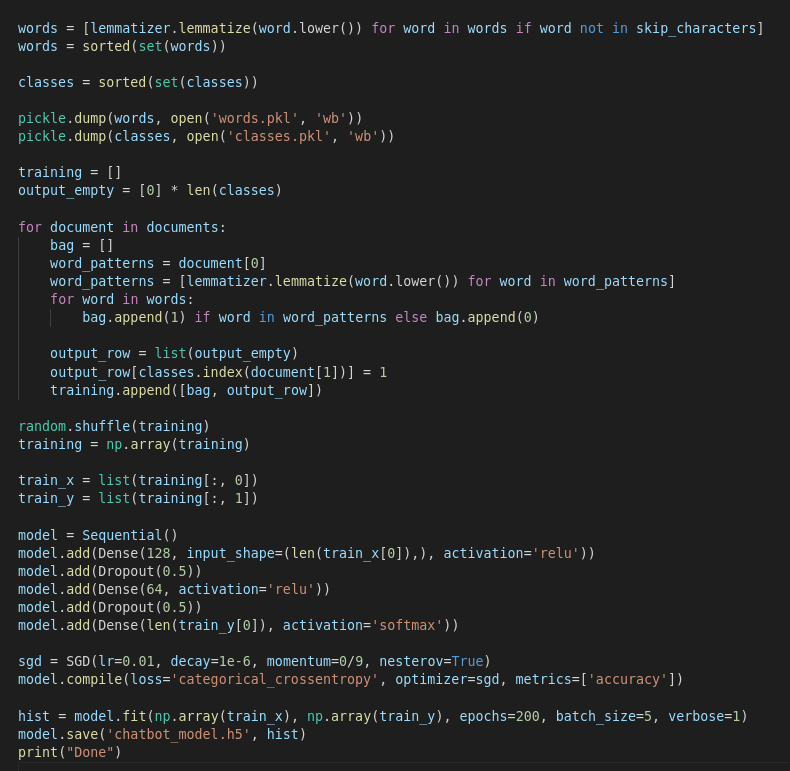
Figure 1: The training data used for Botty the Bot.

**c. Training Model**

For the training model, the words must first be tokenized and lemmatized from the data file. The words were then separated by class and added into into lists based on which class they belonged to. After the data was organized and structured, files were made containing the words as well as the classes. The next step was the most important step which was actually setting up and training the neural network model. The first step was to convert the words into numbers so that the model could interpret them. This was done by converting words to 1s if they were in the word patterns list, or keeping them as 0s if they weren’t in there. Then the bag of words was added to a training list and shuffled in preparation to train the neural network.

Figure 2: Prepping data to train neural network.

After the training data was prepped, the next step was to actually build the sequential model. A simple training model was used and adapted to fit the training data. The training model consisted of a few layers and some dropout layers to prevent overfitting. An sgd optimizer was used and the only metric returned from the model was accuracy, since that’s all that the bot will be interested in. Model is then trained and saved to file for the bot to use later.

Figure 3: Training model setup.

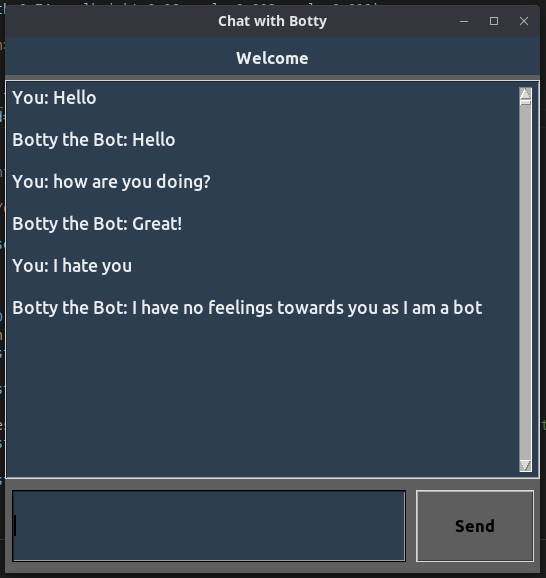
**d. Actual Chatbot**

After building the training model and getting the data ready, the next step was to put everything to use. The most important part was arguably giving the bot a cool name, which in this case was definitely successful. After that the bot was given a few functions. The purpose of these functions was to tokenize and lemmatize the user’s input, convert into a bag of words of 0s and 1s so that it can be pushed to the training model, a function to predict the class that the user input would fall under, and then choosing a random response which matches the class of the user input.

Figure 4: Chatbot functions

**e. Adding a GUI**

Initially, the chatbot was just running in the console, which wasn’t necessarily a fun or engaging way to interact with the bot. The solution was to either add the bot to a website, which doesn’t exist, or to create a simple GUI. The obvious choice was to create the GUI

Figure 5: A simple GUI for running the bot in.

**Video Link**

https://www.youtube.com/watch?v=yO9aSrUDkd4&ab\_channel=avasharif

**Conclusion**

This chatbot is a fairly simple application using the tools which we have learned in this class. It was a very good way to use these tools in real life and while the scope of the chatbot created for this project isn’t very broad, with more experience and more skills, it is possible to create bots that are capable of much more in terms of interaction as well as bots that are capable of even formulating their own responses.

Struggles that we faced with this project were initially just figuring out what to do and what sounded the most interesting while still seeming feasible. The main struggle we ran into was having to recollect knowledge of tools that we used and also seeking information from other sources on the web to figure out how much was possible with these tools. Overall, this project was really helpful because it forced us to learn and bring together everything that we have learned so far.