Lab 06: Chatbot Creation

Fabian Haas, Markus Reichl, Florian Weingartshofer

Dataset

The dataset used for this project is the Python Questions dataset from Stack Overflow, which was downloaded from [Kaggle](https://www.kaggle.com/stackoverflow/pythonquestions). This dataset consists of three CSV files:

1. `Questions.csv`: Contains information about the questions asked on Stack Overflow. The 'Body' field contains the HTML of the answer.
2. `Answers.csv`: Contains information about the answers to the questions. The 'ParentId' field maps to a question.
3. `Tags.csv`: Contains the tags associated with each question. The 'Id' field here corresponds to the 'Id' in the Questions.csv file.

The 'Body' fields in the Questions and Answers CSV files were cleaned by - removing all HTML using the BeautifulSoup library, - making all text lowercase, - removing all punctuation and fill words, - and only allowing common ASCII characters.

Also, the tags were simplified by only keeping the first tag for each question.

Finally, the data was split into a training set and a test set. The training set contains 80% of the data, and the test set contains the remaining 20%.

The Models

Two models were created for this project: a chatbot as a classifier and a chatbot as a generator.

Chatbot as a Classifier

The chatbot as a classifier was created using a vectorizer and logistic regression. The vectorizer was used to transform the text data into numerical data, and logistic regression was used as the classifier. The classifier was trained on a subset of 10000 questions and 15000 answers. Training with the full dataset took dozens of hours and was very hard to debug, so a subset was used instead.

Chatbot as a Generator

The chatbot as a generator was created using a sequence-to-sequence (seq2seq) network with LSTM layers. The seq2seq model was trained on the entire dataset and can generate a new tag based on the input text.

Results

The chatbot as a classifier achieved an accuracy of 94.8% on the test set. This indicates that the classifier can accurately predict the tag for a given input text most of the time.

The chatbot as a generator was able to generate coherent and relevant responses to the input text. However, the quality of the generated responses can vary depending on the complexity and ambiguity of the input text.

Conclusions

The chatbot created in this project can classify input text into different categories and generate relevant responses. The classifier and generator models provide different approaches to handling user input and generating responses, and both models achieved satisfactory results.

The models could be further improved with better preprocessing, fine-tuning the model parameters, and by improving the grouping of tags.