Documentation and Usage Instructions
(For Assignment 2 Chatbot using RNN)

Methodology

We use the approach of first pre-processing and cleaning the data, then splitting it into train and test datasets. Then we choose the parameters for our model (which in turn is from a variety of choices of models using different libraries). Finally, we fit our model to our train dataset and then test it on our test dataset and see the results.

Text Preprocessing

It is done in various steps. *Tokenization*, the process of breaking down text into individual words or tokens, is performed using the spaCy library. It allows the model to understand the semantic structure of the text. Then, cleaning and normalization steps are applied to standardize the text and remove any irrelevant information. Stopword removal and *lemmatization* are then employed to reduce dimensionality and focus on the essential meaning of the words.

Feature Engineering

Feature engineering involves transforming the processed text into a format suitable for machine learning models. The Term Frequency-Inverse Document Frequency *(TF-IDF) vectorizer* is utilized to convert text data into numerical features. This vectorization technique captures the importance of words within the entire dataset, providing a basis for training the machine learning model.

Model Selection and Training

For the chosen model architecture, a Support Vector Machine *(SVM) classifier* is employed. SVMs are known for their effectiveness in handling high-dimensional data and are suitable for text classification tasks. The model is trained on the pre-processed and vectorized text data, learning the associations between the features and their respective labels.

Evaluation Metrics

The model's performance is assessed using standard evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive

understanding of the model's ability to correctly classify text into predefined labels. The evaluation is conducted on the test dataset to ensure the model's generalizability to unseen data.

Model Architecture

The chosen model architecture combines natural language processing (NLP) techniques with machine learning.

Instructions for Using the Model

1. **Load the Trained Model:**
    - Utilize the **joblib** library to load the trained model file (**trained_model.joblib**).
    - This step ensures that the model architecture and learned associations are available for making predictions.

2. **Predict Labels for New Text Entries:**
    - Preprocess any new text data using the same steps as applied to the training data.
    - Utilize the loaded model to predict labels for the preprocessed text entries.

3. **Interpret the Results:**
    - Analyze the model's predictions using evaluation metrics such as accuracy, precision, recall, and F1-score.
    - Understand the model's performance and adjust as necessary for specific use cases.