**Portfolio Project**

Viet Ngo

Colorado State University Global

CSC525: Principles of Machine Learning

Dong Nguyen

08/04/2024

**Portfolio Project Milestone 3**

The primary goal of this chatbot project is to create an intelligent, adaptive conversational agent capable of understanding and responding to a wide variety of user inputs in real-time. The chatbot will be designed to operate in an uncertain environment, meaning it will encounter diverse and unpredictable user inputs. To handle this effectively, the chatbot will demonstrate an adaptive control solution, adjusting its responses based on the context, user sentiment, and evolving conversation dynamics. The first of four main goals for this chatbot is natural language understanding. The chatbot will accurately comprehend user inputs, including context and sentiment. The second goal is appropriate response generation. The chatbot will produce contextually appropriate and coherent responses. The third goal is adaptability. The chatbot will continuously learn and adapt to new inputs and user feedback. Finally, the last goal is proper user engagement. The chatbot will maintain engaging and relevant conversations across various topics.

There are five libraries that will be necessary for this chatbot. The first is Natural Language Toolkit (NLTK). For initial text processing tasks such as tokenization, stemming, lemmatization, and stop words removal. NLTK's extensive suite of text processing libraries will ensure robust preprocessing of user inputs. The second library is Chatterbot. This library will be used to manage conversation flow and context. While Chatterbot is typically used for rule-based or retrieval-based systems, it will serve as the framework for integrating and managing the generative model's responses within a coherent conversational structure. The third library is TextBlob. This library will be used for additional text processing and sentiment analysis. TextBlob's simple API for noun phrase extraction, part-of-speech tagging, and sentiment analysis will be needed in understanding the user's emotional tone and adjusting responses accordingly. The fourth library is spaCy. spaCy will be needed for high-performance NLP tasks such as named entity recognition (NER), dependency parsing, and generating word vector embeddings. This will help with complex user inputs. The last necessary library is Hugging Face, a transformer library. This is the backbone of the generative response mechanism, capable of producing high-quality, human-like text based on input context.

The adaptive control solution involves three key components. The first is context management. Chatterbot will be used to maintain conversation state and context, ensuring that the chatbot's responses are relevant to the ongoing dialogue. The second component is dynamic response adjustment. TextBlob will be necessary for sentiment analysis to change the tone and content of responses when needed. For example, positive sentiments might yield more enthusiastic responses, while negative sentiments could prompt empathetic replies. The last component and most important component is continuous learning. The chatbot will adapt by leveraging user feedback and additional conversational data to fine-tune the transformer model periodically. This involves retraining the model with new data to adapt to evolving user interactions and preferences.

The general structure of the program begins with initialization. This step includes loading the necessary libraries mentioned above, initializing the conversation framework with Chatterbot, and loading the pre-trained transformer model for response generation. The next step of the program is data preprocessing. NLTK and spaCy will be used to preprocess user inputs. TextBlob will be used for sentiment analysis and additional processing. The third step is understanding user input with the chatbot. spaCy will be used for extracting syntactic and semantic features of the user input. The fourth step will be response generation. The transformer model will be used to generate a response based on the processed input and context will be maintained by Chatterbot. If necessary, adjust the generated response using sentiment analysis results from TextBlob to match the emotional tone of the conversation. The next step will be user interaction and user interface. This step displays the output on some interface and collects user feedback on the response for continuous learning. The last step of the program uses that feedback to create a feedback loop and implement continuous learning. The key is to periodically retrain the transformer model with new data to improve performance and adaptability.

**References**

*Akash. (2024, July 19). Making natural language processing easy with textblob. Analytics Vidhya.* [*https://www.analyticsvidhya.com/blog/2021/10/making-natural-language-processing-easy-with-textblob/*](https://www.analyticsvidhya.com/blog/2021/10/making-natural-language-processing-easy-with-textblob/)

*Mondal, A. (2024, May 4). How to build your AI chatbot with NLP in python?. Analytics Vidhya.* [*https://www.analyticsvidhya.com/blog/2021/10/complete-guide-to-build-your-ai-chatbot-with-nlp-in-python/*](https://www.analyticsvidhya.com/blog/2021/10/complete-guide-to-build-your-ai-chatbot-with-nlp-in-python/)

*Real Python. (2023, October 21). Natural language processing with python’s NLTK package.* [*https://realpython.com/nltk-nlp-python/*](https://realpython.com/nltk-nlp-python/)

*Real Python. (2023a, January 2). Natural language processing with spacy in python.* [*https://realpython.com/natural-language-processing-spacy-python/*](https://realpython.com/natural-language-processing-spacy-python/)