Team Members: **Vivek Kumar Maheshwar**& **Kaung Min Sett**

Maji

A basic version of modern-day chatbots such as ChatGPT

What is a chatbot?
A chatbot is an artificial intelligence (AI) software that can start a conversation with a user through messaging applications, websites, mobile apps or the telephone.

Pytorch will be utilized to create the neural network and Python programming language will be used along with Numpy and Pandas library.

Cover Page

- Maji Chatbot Application
- URL of the project's website or the project's wiki page
- Vivek Kumar Maheshwari: v_kumarmaheshwari@u.pacific.edu (Programmer)
- Kaung Min Sett: k_sett@u.pacific.edu (Product Manager)
- Feb 16, 2023,

System Architecture

This AI chatbot will use recurrent neural networks (RNNs), which require more memory because current output depends on historical inputs. RNNs are based on the Seq2Seq architecture, a more sophisticated LSTM implementation. Long short term memory gives neural networks a mechanism to transition between recalling recent information and information from a long time ago.

Step1: Building of vocabulary and creating word embeddings for better performance
- Creating a vocabulary object using a corpus of data drawn from Torch Text

Step 2: Creating the Encoder
- A Seq2Seq architecture consists of an encoder and a decoder unit.Will be using Pytorch to build a full Seq2Seq model.
- Creating an encoder with an LSTM unit
- Loading pretrained embeddings into the LSTM unit

Step 3: Creating the Decoder
Step 4: Combining them into a Seq2Seq architecture
- Combining encoder and decoder units into a working model

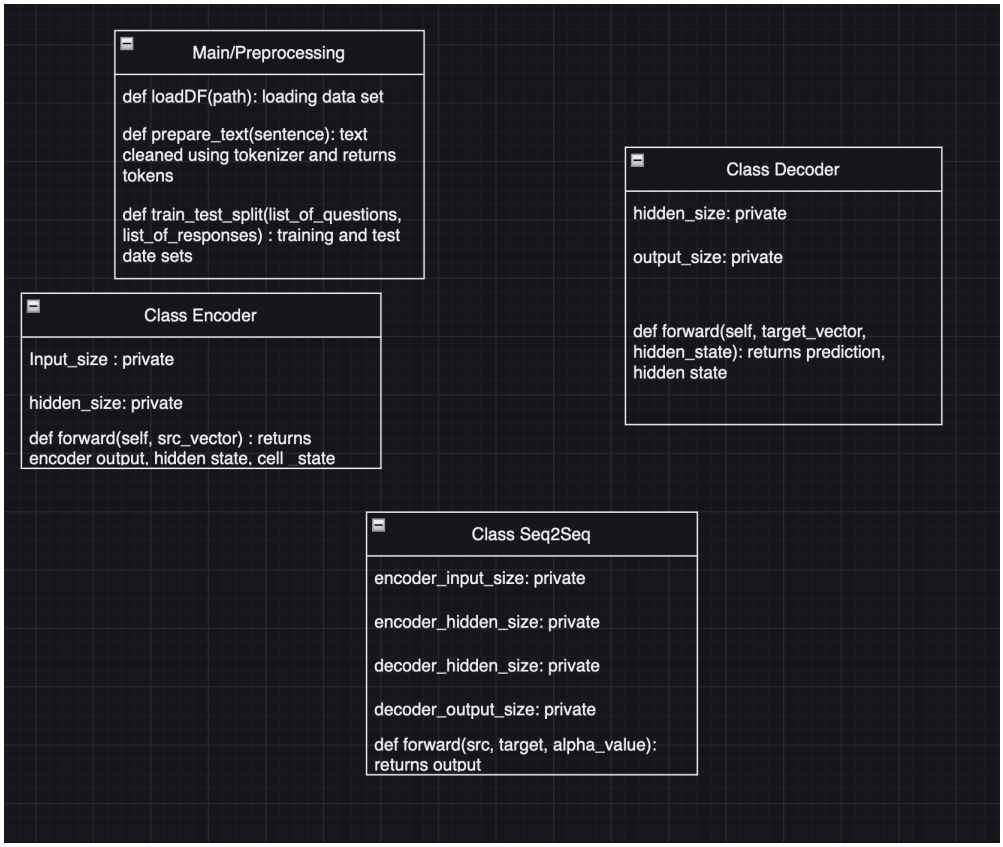
- The Seq2seq will instantiate the encoder and decoder. It will then accept the inputs from these units and manage their interaction to get an output using forward pass function

Step 5: Training and Evaluating the model
Step 6: Interacting with the Chatbot on the command line

Hardware, Software and System Requirements

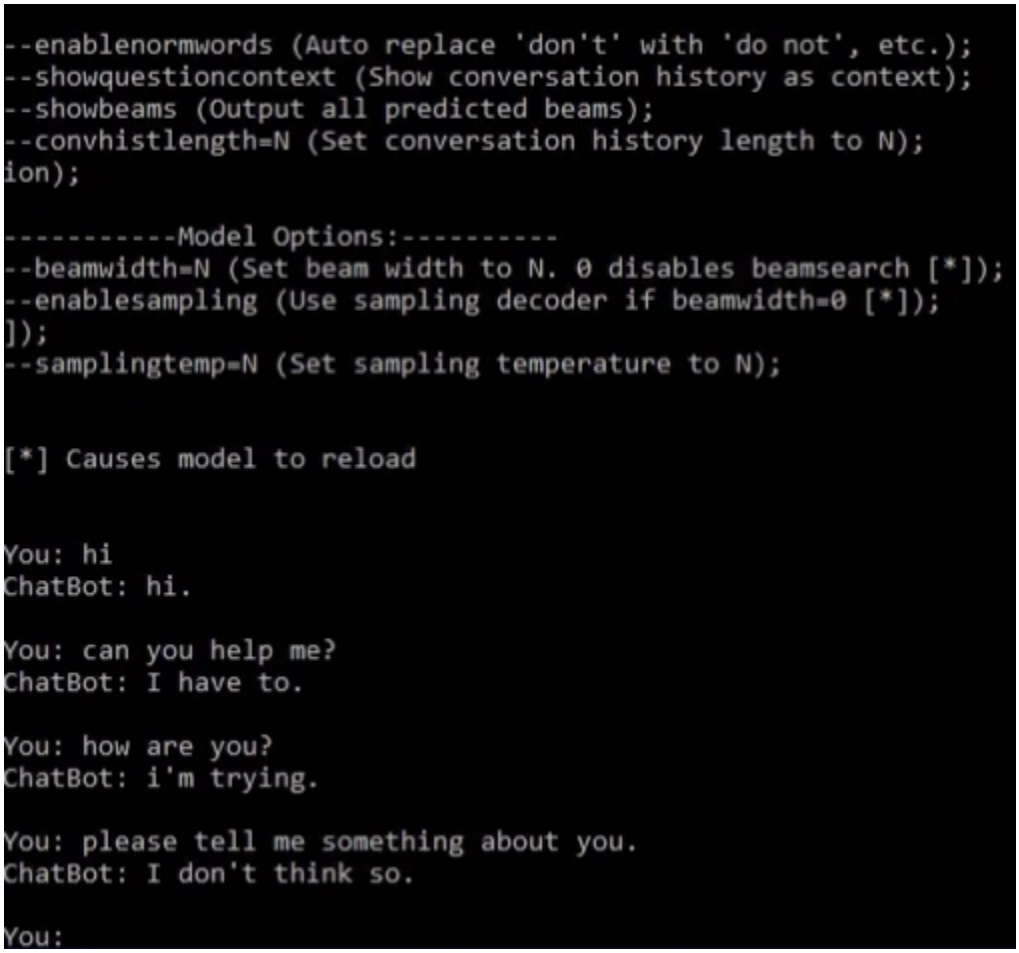
- Hardware requirements: 10 Core CPU, 16GB Ram,Software requirements/Dependencies: Pytorch, Numpy, Pandas, NTLK, Gzip, Gensim
- System requirements: Mac OS, Linux, Windows

Software Design



User Interface Design

Currently, the plan is to use the command line to interact with users. However, a UI would give a good user experience. Therefore, Python's Tkinter framework might be leveraged for a simplistic chatting interface.



Glossary of Terms

List important terms and their definitions used in the SDD, to ensure consistency and avoid ambiguity in the system specification. Use the language of the software and hardware developers. Avoid uncommon terms or define these as well.

Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to process sequential data, where the output of the network depends not only on the current input, but also on the previous inputs that have been fed into the network. Unlike feedforward neural networks, which process a fixed input size and have no memory of past inputs, RNNs have a "memory" that allows them to take into account past inputs and make decisions based on that information. This makes them well-suited for a variety of applications such as language modeling, speech recognition, and time series

prediction. The key feature of RNNs is the presence of recurrent connections, which allow information to be passed from one step of the sequence to the next. This enables the network to maintain a memory of the previous inputs and use that information to influence the processing of the current input. RNNs can be implemented in several different architectures, including simple RNNs, Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs), each with its own strengths and weaknesses. Overall, RNNs are a powerful tool for processing sequential data and have been used in a wide variety of applications.

The Sequence-to-Sequence (Seq2Seq) architecture is a type of neural network architecture that is designed to handle sequential data of variable length. It is commonly used in natural language processing (NLP) tasks such as machine translation, summarization, and chatbot development. The Seq2Seq architecture consists of two main components: an encoder and a decoder. The encoder processes the input sequence and generates a fixed-length vector representation that captures the important information contained in the input. The decoder takes the vector representation generated by the encoder and uses it to generate the output sequence, one element at a time.The encoder typically consists of a Recurrent Neural Network (RNN), such as a Long Short-Term Memory (LSTM) or a Gated Recurrent Unit (GRU), that processes the input sequence element by element and generates a final state vector that summarizes the entire sequence.The decoder is also typically an RNN that takes the final state vector from the encoder as input and generates the output sequence element by element. At each step, the decoder generates a probability distribution over all possible output symbols, and the symbol with the highest probability is selected as the next element in the output sequence. Seq2Seq models are trained using a technique called teacher forcing, where during training the model is given the correct output sequence as input for each decoder time step. During inference, the model is fed its own output from the previous time step as input for the current time step, which is known as the autoregressive decoding strategy. Overall, the Seq2Seq architecture is a powerful tool for handling variable-length sequential data, and has achieved state-of-the-art results in many NLP tasks.

Gensim is an open-source Python library for natural language processing (NLP) and topic modeling. It is designed to handle large-scale text data and includes implementations of several popular NLP algorithms, including topic modeling algorithms such as Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA), as well as word embedding models such as Word2Vec and FastText.

NLTK stands for Natural Language Toolkit. It is an open-source Python library for natural language processing (NLP) that provides a range of tools and resources for processing and analyzing text data. NLTK includes a wide variety of modules and

sub-packages that allow users to perform a range of NLP tasks such as tokenization, stemming, part-of-speech tagging, named entity recognition, sentiment analysis, and more. It also includes several pre-built datasets, such as the Brown Corpus and the Penn Treebank, that can be used for training and evaluation of NLP models.

References

- L, P. (2018, December 27). Build a Chatbot by Seq2Seq and attention in Pytorch V1. Retrieved February 17, 2023, from Medium website: https://chatbotslife.com/build-a-chatbot-by-seq2seq-and-attention-in-pytorch-v1-3cb296dd2a41

- Viraj, A. (2020, October 31). *How To Build Your Own Chatbot Using Deep Learning*. Medium. https://towardsdatascience.com/how-to-build-your-own-chatbot-using-deep-learning-bb41f970e281

- https://github.com/bentrevett/pytorch-seq2seq/blob/master/1%20-%20Sequence%20to%20Sequence%20Learning%20with%20Neural%20Networks.ipynb

- *Chatbot Tutorial — PyTorch Tutorials 1.10.1+cu102 documentation*. (n.d.). Pytorch.org. https://pytorch.org/tutorials/beginner/chatbot_tutorial.html

- Bhashkar, K. (2019, January 20). *Conversational AI Chatbot using Deep Learning: How Bi-directional LSTM, Machine Reading…*. Medium. https://bhashkarkunal.medium.com/conversational-ai-chatbot-using-deep-learning-how-bi-directional-lstm-machine-reading-38dc5cf5a5a3

- *Deep Learning*. (n.d.). Retrieved February 17, 2023, from https://edu.anarcho-copy.org/Algorithm/grokking-deep-learning.pdf

- http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/