Deep Learning Project

Group member: Yinjia Liu & Chenwei Cao

November 25, 2020

* Summary

A chatbot is a software that provides a real conversational experience to the user, here for our project, we build an open domain chatbot which can generate a response as the name implies.

* Tools and Datasets Overview

We used Python and Keras API to build our deep learning models, and google colab; the datasets we used are collected from Kaggle which consists of 2363 entries records conversations between a human and other human acting as a companion bot.

* Data Pre-processing

We used text mining technique to do text corpus preprocessing. We remove punctuation and space with the help of regular expression. After that, we grouped human response with robot response as pairs of sentences since we need to use human response as input sequence and robot response as target sequence, then we shuffle the orders of theses sentence pairs :

Text

Description automatically generated

Here is output for first 10 sentence pairs

Text, letter

Description automatically generated

For target sequences, we have to add <START> at the beginning of the target sentence and <END> at the end of the target sentence.

Then we need to do tokenization, we created separate lists for unique input tokens and unique target tokens.

After that, we created input features dictionary and target features dictionary to store tokens as key-value pairs.

For our model to understand out human and robot language, we need to do Natural language processing (NLP). First, we use one-hot vectors for encoder input, decoder input and decoder output. However, considering one-hot encoding will ignore the inner meaning of the word in a sentence, so next, we use word embedding to help improving model performance.

* Model Selection

We selected LSTM seq2seq model as our training model. We use Keras Functional API as our model structure.

The seq2seq model also called encoder-decoder model which use long-short term memory for text generation from the training sets. It predicts a word given in the user input and then each of the next words is predicted using the probability of likelihood of that word to occur.

First of all, we use the LSTM model with one-hot method

Table

Description automatically generated

* Model Evaluation
* Conclusion
* References