Developing a Chatbot using Sequence Modelling

Problem Statement

Chatbots are proving themselves quite useful now a day's. It has become a way of interacting to the system. Chatbots like Siri(iOS), Google Assistant(Android), Google Allo (Cross Platform), Cortona (Windows), Amazon Alexa and many more, are used daily by many people, in various parts of world. These chatbots help us accomplish us many daily tasks like finding a location, booking a table at restaurant, calling to someone, etc.

These chatbots are all open domain chatbots, I.e. they can be used for genal purpose daily tasks. But they are unable to perform specific tasks like answering questions which are specific to industry, education, or an organization. Such chatbots are called as closed domain chatbots. Many companies deploy such chatbots on their website to answer specific questions related to them. Most of the replies are pre-fed and if any new query comes to the chatbot, it either sends the query to the companies concerned authority to reply the query or straight away ignores the question.

We will be focusing on how to build an open domain end-to-end conversation model for chatbots. The method used for open domain chatbots can be applied for closed domain chatbots as well if there is relevant data available for that domain. The only problem faced by closed domain chatbots is they lack datasets for development, whereas lots of data is available for open domain chatbots. In the next section I will be discuss some background of Chatbots.

Background

In 1966, Joseph Weizenbaum at MIT created the first chatbot that, arguably, came close to imitating a human: ELIZA. Given an input sentence, ELIZA would identify keywords and pattern match those keywords against a set of pre-programmed rules to generate appropriate responses. Since ELIZA, there has been progress in the development of increasingly intelligent chatbots. In 1972, Kenneth Colby at Stanford created PARRY, a bot the impersonated a paranoid schizophrenic. In 1995, Richard Wallace created A.L.I.C.E, a significantly more complex bot that generated responses by pattern matching inputs against <pattern> (input) <template> (output) pairs stored in documents in a knowledge base. These documents were written in Artificial Intelligence Markup Language (AIML), an extension of XML, which is still in use today. ALICE is a three-time winner of the Loebner prize, a competition held each year which attempts to run the Turing Test, and awards the most intelligent chatbot.

Modern chatbots include: Amazon's Echo and Alexa, Apple's Siri, and Microsoft's Cortana. The architectures and retrieval processes of these bots take advantage of advances in machine learning to provide advanced "information retrieval" processes, in which responses are generated based on analysis of the results of web searches. Others have adopted "generative" models to respond; they use statistical machine translation (SMT) techniques to "translate" input phrases into output responses. Seq2Seq, an SMT algorithm that used recurrent neural networks (RNNs) to encode and decode inputs into responses is a current best practice.

Methodology

Most of the methods follow Machine Translation models for Chatbot applications. The only reason behind using Machine translation models is that chatbot works quite similar to Machine translations, only instead of translating one language to another, it will produce a response of the input query. Following are the steps that we can follow to build a chatbot:

Step 1: Collect the data and pre-process it. The preprocessing is done based on the requirement of model. We also divide the data into training and testing data. We can also perform cross validation to gain better results.

Step 2: We perform word embedding to convert the words into vector form.

Step 3: We design Bi-Directional Encoder layer with Long Short-Term Memory(LSTM) cells. These encoders represent the input vector into its own representation.

Step 4: Then we can apply attention mechanism at decoding layer. Attention mechanism helps to remember long-term dependencies for LSTM. In Attention mechanism larger weights are assigned to the words which are more important to context.

Step 5: Loss is calculated by comparing the resultant response to the targeted response. And weights are updated according to that in the next iteration.

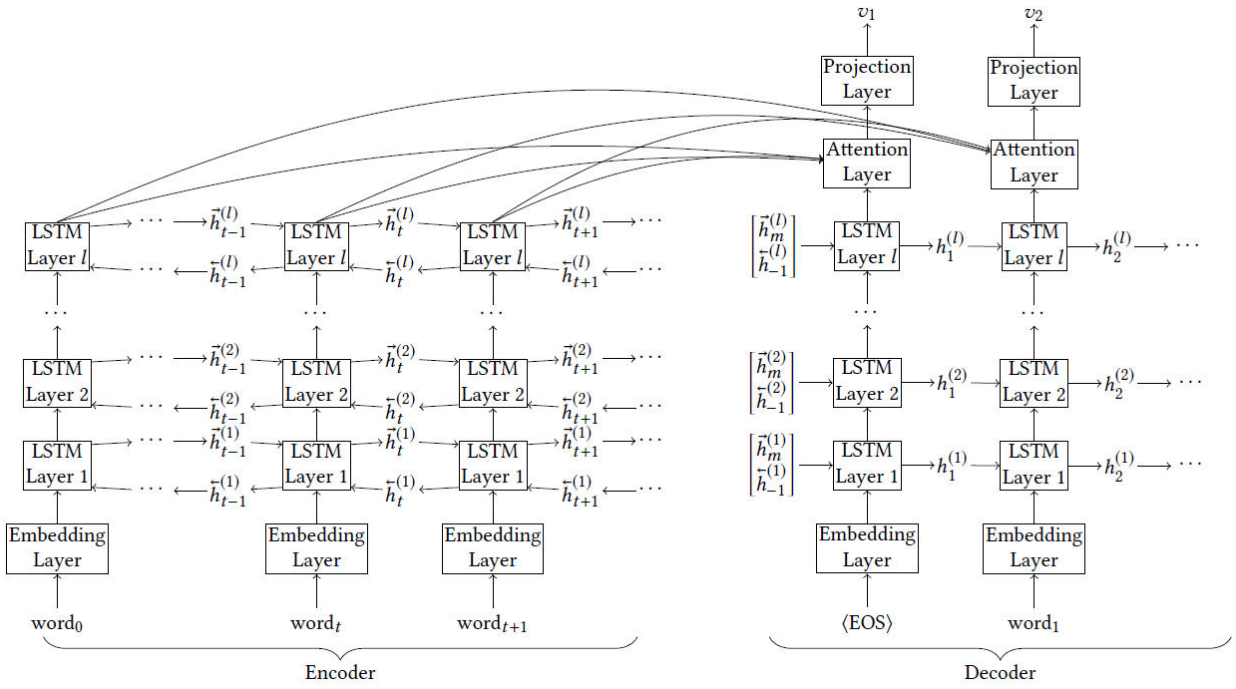

Figure 1: Bi-directional Multilayer LSTM Encoder + LSTM Attention Decoder ["DeepProbe: Information Directed Sequence Understanding and Chatbot Design via Recurrent Neural Networks" by Zi Yin, Keng-hao Chang, Ruofei Zhang. In KDD'17, August 13–17, 2017, Canada]

Experimental Design

Dataset: Dataset is the main attribute of a chatbot, as its accuracy and ability is very much dependent on the data it is trained. Here Since we are discussing about open domain chatbot, we can use Cornell Movie Dialogue Corpus, Reddit conversation datasets or Maluuba Frames Dataset which realised by Microsoft recently. There are many other open dialog datasets which can be used to train a chatbot model.

Evaluation Measures: The evaluation can be done based on F1 scores, this measurement can help to understand whether the model will work properly in real world scenario. There are many other evolutionary methods like BLUE Matrix, Dialogue Efficiency/Quality matrix (oldest method for evaluating chatbots), last but not the least user satisfaction.

Software and Hardware Requirements: Python based Deep Learning libraries will be exploited for the development and experimentation of the project. Tools such as Anaconda Python, and libraries such as Gensim, Tensorflow, and Keras will be utilized for this process.