Analysis of Chatbot Data Generation using LSTM

Deepanshu Sharma, Ayaan Samad, Deepali Dev

Abstract: ***Deep Neural Networks (DNNs) have provided admirable results on difficult learning tasks because of its powerful model. Whenever large data sets are available for preparing training set DNNs works very well as they can be used to process and generate sequences (questions) to sequences (answers). In this paper, we have worked on end-to-end approach related to a regular sequence-learning model that makes minimal presumption on the sequence structure and makes use of our processed data set. Our method makes use of multi-layered Long Short-Term Memory (LSTM) and attention mechanism. Our experimental result is based on a particular set of chat implementation from twitter data set and Cornell movie dialog corpus. The available size of meaningful data is confined, therefore the response time is limited, however the LSTM did not find any difficulty in handling long sentences.***

I. INTRODUCTION

Recurrent Neural Networks (RNN), are uncommon neural networks that assist in dealing with data which has sequential anatomy, like videos(sequence of frames) and more generally, text sequences or basically any chain of pattern. The beauty of RNN is, even if it is not aware about the meaning of symbol is will work completely, therefore RNN will not face any difficulty in deducing the elucidation of symbols, as they look at the form of the text and relative positions of symbols. LSTM networks is a standard of RNN architecture that is designed and programmed to "remember" previously interpreted input values for a small programmed period of time. A basic LSTMs minimum contains three gates that control the flow towards and from their memories and forward to the next layer. The "input gate" controls the input of new information to the memory. The "forget gate" controls how long certain values are held in memory. Finally, the "output gate" controls how much the value stored in memory affects the output activation of the block.

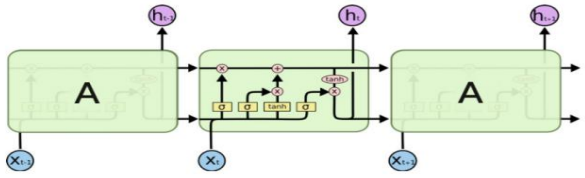

Fig. 1 Unrolled LSTM network [5]

In our work, we investigate with the data by breaking it down word-by-word and transforming and modelling by pitching it to a assignment of forecasting the succeeding order given the preceding sequence to sequences with the help of recurrent networks.

II. RELATED WORK

LSTM was proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber[1]. RNN's are relatively old, like many other deep learning algorithms. They were initially created in the 1980's, but can only show their real potential since a few years, because of the increase in available computational power, the massive amounts of data that we have nowadays and the invention of LSTM in the 1990's[2]. Our methodology is scrupulously connected to Kalchbrenner and Blunsom[3], the problem they face when they first try to plot the input sentence into a vector and then back to a sentence is losing the arrangement of the words, although the procedure they used to plot sentences to vectors with the Convolutional NN. Resemblance to the research, Cho et al. used LSTM like RNN design to plot sentences into vectors and back, although their priority is to integrate their neural network into a SMT system [4]. Since it is widely known that vanilla RNN suffer from vanishing gradients, the majority researchers use variants of LSTM RNN (Hochreiter & Schmidhuber, 1997). Many analyst have pursued raising bots and conversational agents over the last decades, and it is out of the range of this paper to provide an exhaustive list of mention.

III. MODEL

We make use of sequence2sequence model which consists of two RNNs: An Encoder and a Decoder programmed input set. Input is a arrangement of a sentence which is feeded in the encoder and that develops one chat (word) at each timestamp. The primary objective of the encoder is to convert a sequence of patterns into a fixed size feature vector that encodes only the crucial information in the sequence while discarding the unnecessary data saving both space and time. Each concealed state effects the next state and the final hidden state can be looked at as a summary of the arrangement. This concealed state is called the thought vector, as it represents the intention of the sequence. From the context, another sequence is generated by the decoder producing one data part (word) at a time. In this model, at each timestamp, the context and the preceding generated symbols influences the output of decoder. The LSTM has the ability to modify information to the cell state, carefully controlled by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a point wise multiplication operation. The sigmoid layer outputs between zero and one, describing how much of each component should be passed. A value of zero means nothing to be passed, while a value of one means everything to be through.

Revised Manuscript Received on December 22, 2018.
Deepanshu Sharma, Student, ABESEC, Ghaziabad, India
Ayaan Samad, Student, ABESEC, Ghaziabad, India
Deepali Dev, Sr. Assistant Professor, ABESEC, Ghaziabad, India

A. Preprocessing

We filter out all the words, which have been rarely used in the data set. We apply padding on the dataset to modify the irregular length sequences into fixed length sequences. We use special symbols to fill the sequence - '<PAD>', '<EOS>', '<OUT>', '<SOS>'. We clean the text by converting the words to simpler forms egi'm to i am, he's to he is etc. I excluded all the sequences with 0 or 1 character. Also filtering out all the questions and answers that are too short or too long was done. To avoid the wastage of large amount of space due to small size questions we use bucketing and save space by aligning sentences into buckets of different range of sizes. Let the list of buckets: [(10,20), (20,30), (30,40), (40,50)]. Suppose the length of a question is 7 and the length of its generated response is 8 (as in our previous example), we put this sentence in the bucket (10, 20).

B. Attention Mechanism

The attention tool which we make use of in our research model in combination with Neural Machine transcription and by Jointly researching to Align and Translate [5], this allows the decoder part of the model to precisely look at the input pattern while decoding. This eases the workload of the encoder to process every advantageous material from the input. During each timestamp for which the decoder works, instead of using an established.

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j$$

where n is the length of input sequence and Hj is the hidden state at timestamp j

$$\alpha_{ij} = \exp(e_{ij}) / \sum_{k=1}^{n} \exp(e_{ik})$$

eij is objective of decoder's previous concealed state Si−1 and the jth concealed state of the encoder and represents alignment model. This alignment model is characterized as a feed forward neural network, which is collectively trained with the rest of model data.

IV. EXPERIMENTAL RESULTS

In this section, we describe the experimental results with the two datasets and show some samples of the interactions with the system that we trained. The questions which we need to ask should be similar to the data set on which we are working to get the best possible results. We have plotted graph of frequency vs words (most frequent to least frequent) based on the data. We have used vocabulary, which cover most the area under the curve.

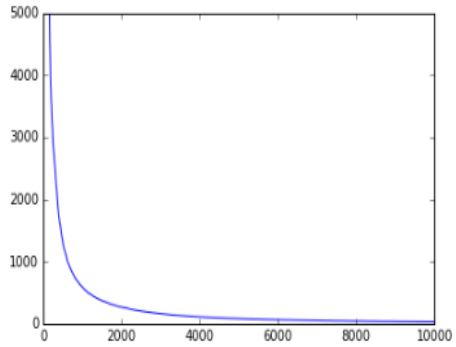

Fig. 2 Frequency vs Words Graph based on the Data

A. Cornell Movie Corpus Data Set

```
you: you can do it
chatbot: yeah i think so
you: life is so difficult
chatbot:it is meant to be
you: i am going to miss you
chatbot:no you are not
you: i wish life was fair
chatbot:why would you like to be
```

The important point is that the model can generate meaningful response to new questions. In other words, unlike the basic sequence-to-sequence model it does not simply look up for an answer by matching the question with the existing database and makes use of processed data. Most of the questions, which have been asked above, do not appear in the training set. One defect of this basic model is that it only gives transparent answers and when there are few or no references to the data for the questions asked, unsatisfying results to our questions are generated as can be seen above. Another drawback for our research model is that the model does not capture the true sense of data or conversations taking place and if we ask comparable or related questions, the output can sometimes be contrary to one another.

B. Twitter Data Set

```
You: the world cup had some really thrilling games and i
enjoyed it
ChatBot: how did you get to get a new one for the first
time i can get it

You: i am thinking of going to usa for my higher studies
the quality here is not very good
ChatBot: the only thing is a real woman and a woman who
can be a better job than a woman

You: chill out its over
ChatBot: its not a good time

You: lets have a get together on sunday havent met you
all in a long time
ChatBot: we will be there in the city in the city

You: the elections are coming near and the suspense of
who will be our next leader is making me crazy
ChatBot:  i love this show

You: hey happy birthday and have a nice day
ChatBot: thank you so much

You: i am very happy today as i met my idol
ChatBot: omg i love you
```

V. CHALLENGES AND DISCUSSION

The model faces difficulty in handling variable length sequences and the sentences need to be transformed. Another challenge faced by our model is that the decoder has to run softmax function has to run over a large vocabulary of words(approximately let us take 50000 words), for each set(word) in the output. That is going to reduce the speed of the coaching process, even if your hardware is good enough for handling it. In this paper, we have worked on a simple language model based on the sequence2sequence framework and making use of LSTM and attention technique can be used to train a conversational engine. The generated results show that it can generate simple and basic conversations, and extract knowledge from an open-domain dataset. This model can produce rather proper answers to many types of questions. However, the model needs large amount of sizable modifications to be able to deliver practical conversations. There are some obvious drawbacks, and we are actively pursuing this and other research directions. One of challenges left for the future is to better handle unknown or rare words. This will be required for the model to be more widely used and to match the performance of current state-of-the-art machine translation systems in all contexts.

VI. CONCLUSION AND FUTURE WORK

We can see that the responses which help to generate accurate answers' are only limited to the limit of the data set which we use and when we ask questions which have no relevance to the dataset the output generated is berserk.

```
You: which is the farthest star
ChatBot: the whole thing is the worst

You: who do you think will win the next elections
ChatBot: trump won the debate
```

We however believe that it remains an open research problem of designing a good metric to quickly measure the quality of a conversational model. However, there are obvious drawbacks on using those, and we are actively pursuing this and other research directions like combining image detection and deep natural language processing to automatically study the images and generate the response as a human.

REFERENCES

1. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 1997.
2. https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/
3. https://www.aclweb.org/anthology/D13-1176
4. https://www.aclweb.org/anthology/D14-1179
5. https://arxiv.org/pdf/1409.0473.pdf
6. Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In NIPS, 2014.
7. Graves. Generating sequences with recurrent neural networks. In Arxiv preprint arXiv:1308.0850, 2013.
8. Wu, Y., Wu, W., Xing, C., Xu, C., Li, Z. and Zhou, M. (2018). A Sequential Matching Framework for Multi-Turn Response Selection in Retrieval-Based Chatbots. Computational Linguistics, pp.1-35