Towards Building A Neural Conversation Chatbot Through Seq2Seq Model

J.Prassanna, Khadar Nawas K, Christy Jackson J, Prabakaran R, Sakkaravarthi Ramanath

Abstract: Improvements in computation and processing power paved a way for Machine learning to be applied more efficiently in real-time and in a lot of applications. In which most prominent area is Natural Language Processing and Natural Language Understanding, which helps the computer to process and understands the natural language used by people. Thanks to deep learning models and architectures which made this process of making the system process and understand natural language, which makes the system more intelligent. Chatting agent's AKA-Chatbot is one of the major use cases of Natural Language Processing and Natural Language Understanding, which can be used in different domains to engage customers and provide a response to customer's queries. Though many chatbots use a retrieval-based model with the recent advancement of Deep Learning, we in this work use Neural Networks to train a chat model with a question and answer datasets that make models understand the patterns in it and behave intelligently. Here we build a domain-specific generative chatbot using Neural Networks to train a conversational Model which reads the pattern of data and reply answer when a new question is asked. Finally, we conclude by validating how relevant the response generated by the model to test data or test question and provide a further area of improvements to make the system more efficient and intelligent.

Index Terms: Chatbots, Seq2Seq, Word2vec.

1 INTRODUCTION

Deep Neural Networks (DNN is unrestrained hyperactive machine learning model which helps to solve the most complex problem and showing efficient and promising result in problems which it is applied. The complex problems which it is applied are speech recognition [1-3], object recognition. The thing which makes them powerful is, they fundamentally gain abstract parallel computation for a reasonable number of steps. One best example of DNN is handwritten digit recognition using Convolutional Neural Networks (CNN). Ultimately neural networks are an interconnected group of nodes waiting upon to the propertied statistical model, they learn extremely complex computation. A furthermore very deep neural network can be trained with back propagation with good network learning parameters. Recent advancement in Neural Networks to handle sequential data has improved the applications of DNN in Natural language processing and Natural Language Understanding. But DNN is not capable of dealing with sequential data; it can deal only with inputs and outputs with fixed dimensional vectors. But many important problems like machine translation, text summarization, question answering system etc. depends on sequential data. However, Seq2Seq model by Ilya Sutskever [4] is one which is dedicated to handling sequential data, which has remarkable success in Machine Translation task. And it shows great promise in other NLP problems like Text Summarization, Speech Recognition, Keyword Extraction etc. In which one of prominent is Conversation System.

- *J.Prassanna, SCOPE, Vellore Institute of Technology, Chennai, India, E-mail: prassanna.j@vit.ac.in*
- *Khadar Nawas K, SCOPE, Vellore Institute of Technology, Chennai, India.*
- *Christy Jackson J, SCOPE, Vellore Institute of Technology, Chennai, India.*
- *Prabakaran R, SCOPE, Vellore Institute of Technology, Chennai, India.*
- *Sakkaravarthi Ramanathan, Department of Computer Science, Cegep Gaspesie, Canada.*

Using the concept of seq2seq model, the word sequences representing question are mapped to a fetter of words representing reply. And again, it is possible only of the problem is domain independent. In this paper, we show the application of Seq2Seq model in Chatbot. Seq2seq model use architecture called Long Short-Term Memory (LSTM) which is a specific type of Recurrent Neural network (RNN) architecture. The major use of its design is to take care of temporal sequence and their long-range dependencies more accurately then vanilla RNN [5]. The general idea is to use the idea of seq2seq for machine translation in chatbot where both question and answer are in the same language. The fundamental thing is to utilize one LSTM to peruse the input sequence, one step at once, to acquire substantially huge fixed-dimensional vector representation and use another LSTM to abstract the output sequence form vector. The first LSTM is called encoder, second LSTM is called decoder which is an RNN language model except that it is conditioned on the input sequence. The ability to handle the large sequence of inputs and ability to handle long-range temporal dependencies make seq2seq model a standalone model for natural language processing problems which majorly has sequential inputs. There are many ways suggested by researchers to solve general sequence to sequence issue. Our strategy is firmly identified with [4] Ilya Sutskever, the main benefits of this framework are that it requires less feature engineering process, less domain specific Wishlist matching, no need for domain Wishlist matching if dataset is well formed and it of single domain. Conversational modeling directly benefits from this framework because the simplest procedure for the sequence to sequence learning is to align the input sequence to fixed size vector utilizing one LSTM to the objective sequence with another LSTM. Here I am going to experiment with conversation modeling by given the previous sequence of input and predicting the next sequence.

2 MODEL

Recurrent neural network is feed forward neural network which helps greater in sequence data, the standard RNN computes the output sequence $(u_1 \ldots\ldots u_t)$ given the input sequence $(v_1 \ldots\ldots v_t)$ using iteration of this following equation.

$$z_t = xw_t + b_1$$

$$a_t = tanh(z_t)$$

It is easy to map sequence to sequence at whatever point the arrangement between the sources of info and yields is known early. Be that as it may, it impractical to apply RNN when inputs and outputs have different lengths, so it is hard to prepare a RNN because of coming about long term dependencies. However, LSTM long short-term memory learn problem with long range temporal dependence it goes well for our problem. The below figure 1 shows the working of RNN.

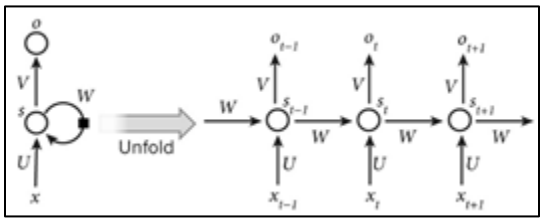

Fig 1 *Unrolling RNN.*

Here the main objective of LSTM is to estimate the conditional probability $p\langle u_1 \ldots\ldots u_t | v_1 \ldots\ldots v_t \rangle$ where $(v_1 \ldots\ldots\ldots v_t)$ is the input group sequence and $(u_1 \ldots\ldots\ldots u_t)$ is the relating output group sequence in which their range differs. LSTM figure out the restrictive probability by first getting the fixed dimensional vector V of input group sequence $(v_1 \ldots\ldots\ldots v_t)$ given the last concealed condition state of the LSTM. And after that process compute the likelihood of output group sequence $(u_1 \ldots\ldots\ldots u_t)$.

$$p\langle u_1 \ldots\ldots u_1 | v_1 \ldots\ldots v_t \rangle = \prod_{t=1}^{T} p\,\langle u_t | V, v_1 \ldots\ldots v_{t-1} \rangle \quad (1)$$

We use SoftMax [6-7] over all the words in vocabulary to represent this equation (1). To train this network we need the <EOS> tag at each end of the given sentence so that the distribution could happens over different length sentences, the figure 2, given below represents the model.

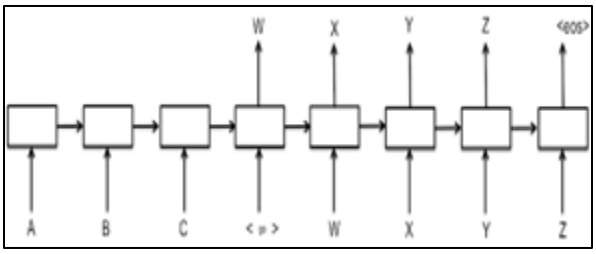

Fig 2 *Seq2Seq model.*

Here the A, B, C are inputs where W is the context vector which holds mapping vectors information's of all the inputs using the context vector output sequence X, Y, Z are mapped and predicted. Here we use two LSTM, one for input and other for outputs. Ilya Sutskever [4] suggested that training input sequence in reverse order decreased the word perplexity and increased the accuracy of the model.

3 DATASET

In our experiment, we used Message Chat Data Sets between different people, which have more than 3000 conversations, which we find is clean and spotless for training better than other different datasets. You can use other very big datasets like Ubuntu dialogue corpus, standard question answering dataset and Microsoft maluuba chat frames dataset, however, all this require high computation to train. It would be easy to train a model if you have Nvidia TITAN X GPU [7-8]. Another issue in Ubuntu dialogue corpus is very noisy data which need a lot of preprocessing, however, all these datasets have negatives which make your system, not context means. So, it is better you collect your own data, however it very tedious process, but a manual collection of data makes a machine learning model efficient and it gives the context of business. You won't believe even big players link Google, Microsoft, Facebook etc. relies on collecting data manually for their research and application.

4 EXPERIMENT

4.1 Feature Extraction

Neural Networks can understand only numbers so first, we have to extract features [9] from our data and convert it to the corresponding word vector matrix so that we can train our model. There is a different feature extraction technique for differing data in sensible image data, sound data, and text data. Here our data is text data. Their features extraction for text data falls under two categories. 1) Frequency based 2) Prediction based.

Frequency Based Feature Extraction:

- count vectors.
- TF-IDF (Term frequency inverse document frequency).
- co-occurrence matrix.

Prediction Based Feature Extraction:

Word2Vec model [10]:

- Bag of words.
- Skip-gram model.
- This different technique has its own applications where count vectors are used in key word extraction problems, TF-IDF is used in document classification, information retrieval and keyword extraction. Co-occurrence matrix to identify relation between which co-occur in the document. Here in our experiment word2vec model is best suitable.
- Word2Vec model is using to extract features from our data and convert into vector matrix. Word2vec can be built by two ways one is building a separate model using NumPy and pandas or we can make use TensorFlow word2vec estimator.

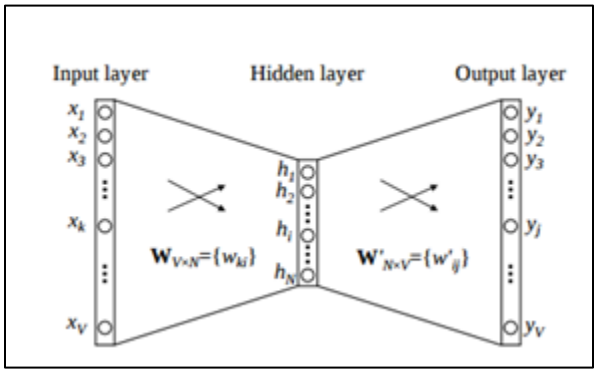

Fig 3 *Word2vec neural networks*

Word2vec is a two-layer neural network model which convert words or tokens into corresponding word vector by training this model we have word embedding matrix which we use further for training purpose of Seq2Seq model.

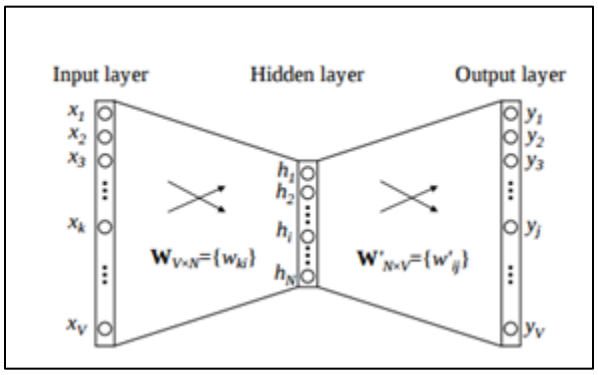

Fig 4 *Embedded matrix*

Then create the word list of the dataset, convert dataset into tokens and store in a separate file which we must use in training seq2seq model.

4.2 Training

We build a Seq2Seq model and created training matrices with the help of word2vec matrix we created using the word embedding model and with the help of word list of the datasets. To train the model, we used Seq2seq model given by Google for machine translation for chatbot. However seq2seq model core structure is same for any of its application we apply but we must modify the model according to our problem. Since we have 3000 sentence there is 30000 unique words so embedding matrix dimensionality of data is very large. Here we must append word list with <EOS> to identify the ending of the sentence to enable model do probability distribution accordingly. Then set the hyperparameters:(batch Size = 24 maxEncoderLength = 15, maxDecoderLength = maxEncoderLength, lstmUnits = 112 embeddingDim = lstmUnits, numLayersLSTM = 3, numIterations = 500000) and create encoder and decoder estimator using TensorFlow.

```
('Current loss:', 7.775767, 'at iteration', 0)
hey how are you
['aga ']
('Current loss:', 7.3225894, 'at iteration', 50)
hey how are you
[]
that girl was really cute tho
[]
('Current loss:', 5.288179, 'at iteration', 100)
whats up bro
[]
that girl was really cute tho
[]
('Current loss:', 3.990014, 'at iteration', 150)
hey how are you
[]
that dodgers game was awesome
[]
('Current loss:', 2.505266, 'at iteration', 200)
that dodgers game was awesome
[]
hey how are you
[]
('Current loss:', 2.8571527, 'at iteration', 250)
hi
[]
that girl was really cute tho
[]
('Current loss:', 2.9098845, 'at iteration', 300)
hey how are you
[]
that girl was really cute tho
[]
('Current loss:', 1.577684, 'at iteration', 350)
that dodgers game was awesome
[]
that girl was really cute tho
[]
('Current loss:', 2.358489, 'at iteration', 400)
hi
[]
whats up bro
[]
('Current loss:', 1.8704377, 'at iteration', 450)
hey how are you
[]
that girl was really cute tho
[]
('Current loss:', 2.18371, 'at iteration', 500)
that girl was really cute tho
```

Figure 5 *Training of seq2seq model*

Here we used rectifier linear unit ReLU activation function to calculate loss in used sequences from TensorFlow seq2seq model estimator and used Adam optimizer to minimize the loss. This function done well in ([4] Ilya Sutskever) machine translation model so we went with it, trying with different function helps to know insights of the model. The hyperparameters of the model completely depends on what dataset you are using, your size of the datasets, size of the embedding matrix and size of the word list number of unique words in the datasets. Tweaking hyperparameter helps to know about model better and analyze its performance.

4 CONCLUSION

In this work, we showed how seq2seq model can be used for building chatbot with the explanation of each stage in building the model. However, still the model is not that much accurate we provided a simple way of approaching a method of building a chatbot but according to top researchers in field of the deep learning like Ilya Sutskever [4] and Oriol V [11] there is much hype in building intelligent system. But it is still not able to cope up with real time natural language when it is deployed. It is in the stage of how the internet was 20 years back. It needs to build with a structure where all-natural language things like syntax, semantics, context, intent etc. should be handled by the system in a more sophisticated way. As of now the best chatbot accuracy is achieved by Alibaba research group [12], however that model is a combination of IR(information retrieval) model and attentive seq2seq model where answers from both the model is ranked and if the answer rank for the given question is above the fixed threshold then that answer is given to the end user.

5 REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, "A Neural Probabilistic Language Model", J. Machine Learning Research, vol. 3, pp. 137-1155, 2003.

[2] Y. Bengio, P. Simard and P. Frasconi, "Learning long-

term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March 1994.

[3] Sak H, Senior A W, Beaufays F, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling", Proceedings of the 15th Annual Conference of the International Speech Communication Association. 2014, 338–342

[4] Ilya Sutskever , Oriol Vinyals , Quoc V. Le, "Sequence to sequence learning with neural networks", Proceedings of the 27th International Conference on Neural Information Processing Systems, p.3104-3112, December 08-13, 2014, Montreal, Canada

[5] Z. C. Lipton, "A Critical Review of Recurrent Neural Networks for Sequence Learning", The Computing Research Repository (CoRR), vol. abs/1506.00019, June 2015.

[6] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, “On using very large target vocabulary for neural machine translation,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015, pp. 1–10.

[7] K. Iwata, "Extending the Peak Bandwidth of Parameters for Softmax Selection in Reinforcement Learning", IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 8, pp. 1865-1877, Aug. 2017.

[8] Stroia, L. Itu, C. Niţă, L. Lazăr and C. Suciu, "GPU accelerated geometric multigrid method: Performance comparison on recent NVIDIA architectures", (2015) 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, 2015, pp. 175-179

[9] Chae-Gyun Lim, "A survey of temporal information extraction and language independent features," 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, 2016, pp. 447-449.

[10] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, "Linguistic regularities in continuous space word representations", HLT-NAACL (2013). Vol. 13, pages 746–751.

[11] Oriol Vinyals and Quoc V. Le. (2015), " A neural conversational model", CoRR, abs/1506.05869.

[12] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, Wei Chu, "AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine", Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. (2017)