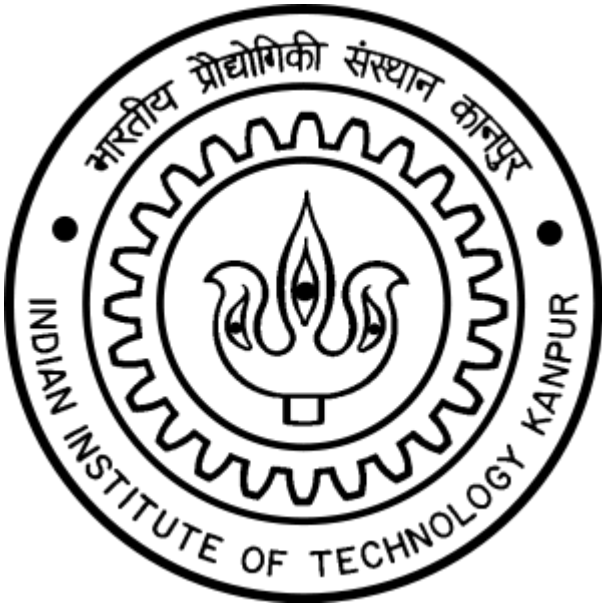
A Neural Conversational Model

CS671A: Introduction to Natural Language Processing

Advisor: Prof. Amitabha Mukherjee



Aadil Hayat (13002)

Masare Akshay Sunil (12403)

1 Abstract

Conversational modelling is one of the most exciting problems in the field of Natural Language Processing. Since Eliza[1], many attempts have been made to improve the conversation model. But, most of these attempts were restricted to specific domains[2] and required hand-crafted rules. The Neural Conversation Model[3] tries to model our agent using just previous sentence or sentences. It is trained end-to-end and hence, require less hand crafted rules. Our agent can have simple conversations if trained with large enough dataset even if it is as generic and noisy as a movie subtitle dataset.

2 Traditional ChatBots

Traditionally, most chatbots that we have seen have used hand-crafted rules for conversation. But, making all these rules can be quite tedious. Also, this also made making a open domain chatbot very difficult. Neural Networks still existed but, they were mostly overlooked by the chatbots. The main reason for this being that the Neural Networks require input and output to be of fixed dimension . But, in a conversation model, we deal with variable input and output length, and hence its dimensionality cannot be known a-priori. But, in recent years we have seen many workarounds for this.

3 Neural Networks in ChatBots

Almost all of the work in Neural Conversation Modelling has been done in the last few years. It is largely based on the work of Sutskever, et al.[4] which uses neural networks to map sequences to sequences. This framework was first used for neural machine translation and archival. As RNN, by itself, suffers from vanishing gradients, a variant of Long Short Term Memory (LSTM) RNN based on the works of Hochreiter et al[5] . The works of Sordoni et al. [6] and Shang et al.[7] also used RNN to model dialogue in short conversations. Our approach is based on producing answers given by a probabilistic approach to maximize the correctness of answer in the given context.

4 Our Approach

We are using the approach suggested in the Neural Conversation Model by Vinyals, et al[3]. It is based on the Seq2Seq[4] framework described above. It uses two LSTMs : one for encoding and one for decoding. To preser e context, the input sequence is the concatenation of what has been conversed so far, and the output sequence is the reply.

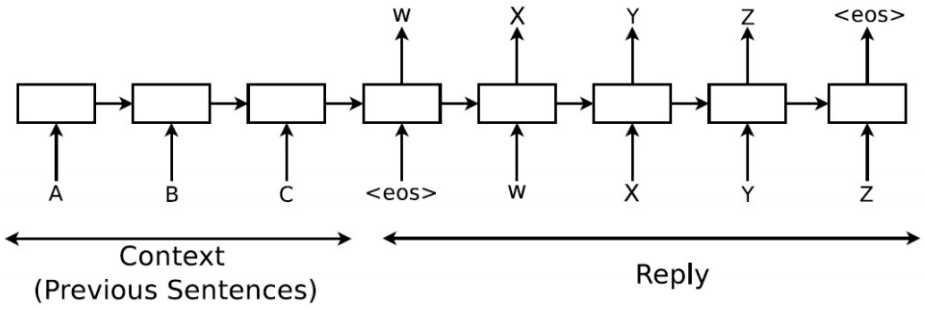


Figure 1: The Neural Conversational Model

Let us look at a example given in [3]. Suppose we observe a conversation between two persons and the first person utters "ABC", and the second person replies "WXYZ". This segment of conversation is trained with to produce a map from "ABC" to "WXYZ". The encoder reads the input "ABC" until it gets the end of statement flag, in reserve order to generate a vector embedding. This becomes the input of the decoder to train the first word "W", and then, "W" is again put back in the decoder along with the input "ABC" to generate next word.

5 Dataset

We trained our model on the OpenSubtitles dataset by Tiedemann[8] . This dataset consists of movie conversations in XML format. Our training and validation split has 62M sentences (923M tokens) as training examples, and the validation set has 26M sentences (395M tokens). The split is done in such a way that each sentence either appear together in the training set or test set but not both. The OpenSubtitles dataset is quite large, and rather noisy because consecutive sentences may be uttered by the same character. Given the broad scope of movies, this is an open-domain conversation dataset.

5.1 Pre-processing

We have the dataset in XML format. We did a simple parsing to remove the XML tags and add an end of statement indicator at the end of each statement. We also removed obvious non-conversational text like hyperlinks, movie title and names of the creators. The speaker of a statement was not indicated in the dataset, so we just assumed consecutive statements were made in response to the previous one.

6 Implementation

We tried to use Keras library for implementation of our model. But, due to the sequencial nature of our input, it failed to create a proper LSTM for our purpose. So, we shifted to TensorFlow.

6.1 Tensor Flow

TensorFlow is the open source library for machine intelligence by Google. It provided for an easy implementation of the Seq2Seq model using a custom python wrapping called Bazel.

6.2 System

We training out model on a GPU. The machine we are using has Nvidia GTX 760 GPU, which will handle the training of our model. This GPU has CUDA support with 1664 CUDA cores.

7 Future Work

This model is currently trained on a very noisy OpenSubtitles Dataset. In the future we would like to test on a more structured dataset with a closed domain for better results. Also, currently the model is completely unsupervised. Hence, with little supervision we can achieve a lot better results.

Refrences

[1] Weizenbaum, J. *ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine.* Communications of the ACM, Vol 9, 1966. [

2] Huang, J., Zhou, M., Yang, D. *Extracting Chatbot Knowledge from Online Discussion Forums*. IJCAI07-066, 2007

[3] Vinyals, O., Le, Q. V. *A Neural Conversational Model*. arXiv:1506.05869v3, 2015.

[4] Sutskever, I., Vinyals, O., and Le, Q. V. S*equence to sequence learning with neural networks*. In NIPS, 2014.

[5] Hochreiter, S. and Schmidhuber, J. *Long short-term memory.* Neural Computation, 1997.

[6] Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Gao, J., Dolan, B., and Nie, J.-Y. *A neural network approachto context-sensitive generation of conversational responses*. In Proceedings of NAACL, 2015.

[7] Shang, L., Lu, Z., and Li, H. *Neural responding machine for short-textconversation*. In Proceedings of ACL, 2015.

[8] Tiedemann, J. *News from OPUS - A collection of multilingual parallel corpora with tools and interfaces*. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R. (eds.), Recent Advances in Natural Language Processing, volume V, pp. 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, 2009. ISBN 978 90 272 4825 1.