Toxic Comments Classification using LSTM.

(The data was taken from one of the Kaggle challenges and Long term time memory networks.)

To study more about LSTM go to this link - <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>

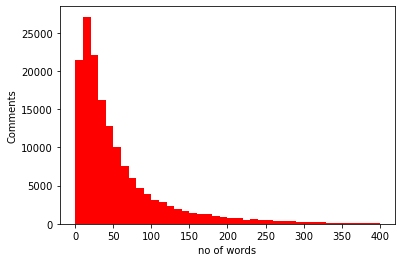
The train data was classified in 6 categories.

["toxic", "severe\_toxic", "obscene", "threat", "insult", "identity\_hate"]

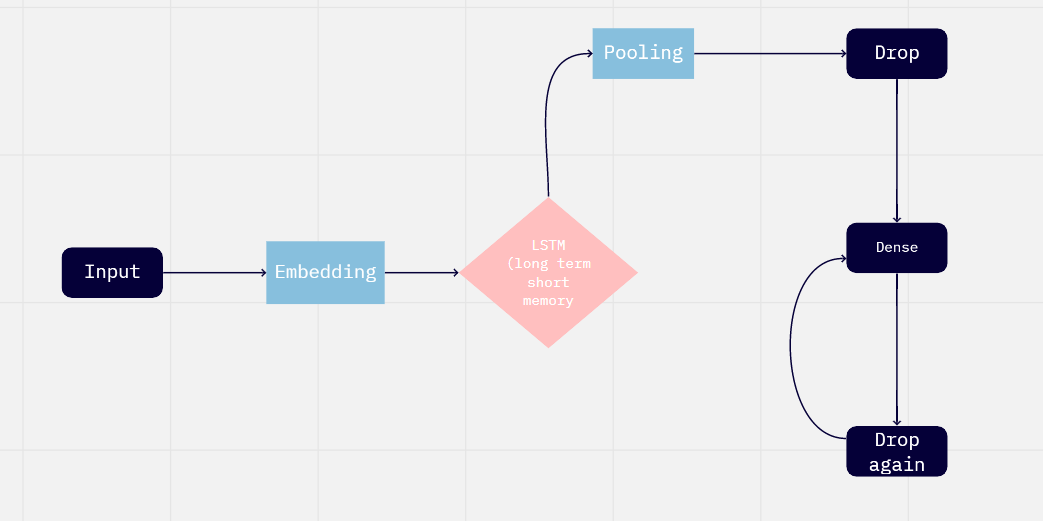
The next step was to Tokenize the comments of dataset.  
Both train and test datasets were tokenized using Keras tokenizer function.

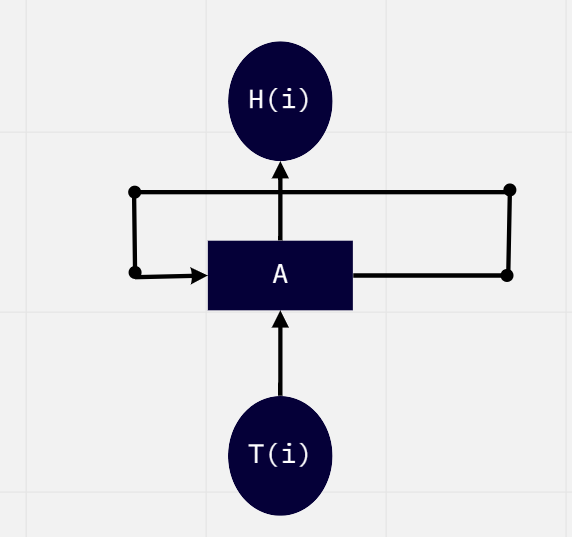
Since every comment was not of equal size padding was done to make every comment of equal length.

Following is the histogram to visualize the number of words each comment has-



**System Architecture**





In the above picture, the LSTM will be unrolled and fed to the next A and give us a set of h0,h1,h2 until the last h.

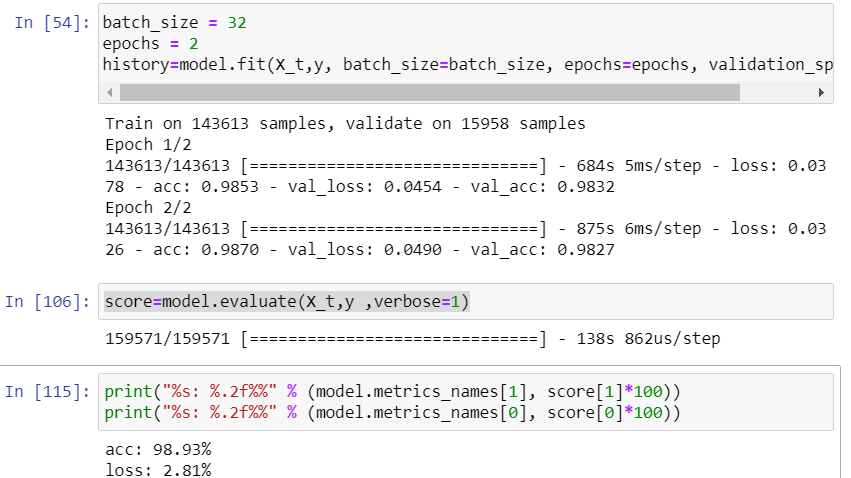
 LSTM takes in a tensor of [Batch Size, Time Steps, Number of Inputs].

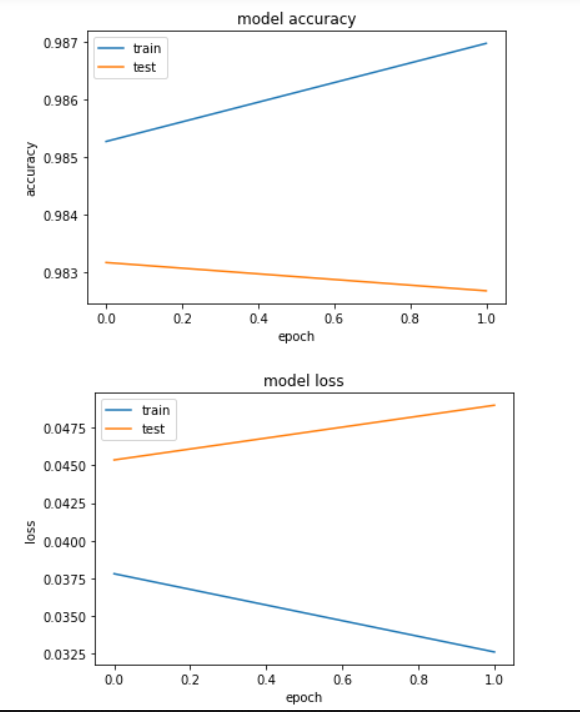
After embedding, the 3d tensor will be converted to 1D using the Global Max pool and drop 10% of the neural nodes every time.

First, the dropout layer is fed into a Relu function and then that output is again fed into a sigmoid function.

Now the model is trained using Keras model.compile().

The following outputs were recorded.





Tools and Library Used

Language-Python

Interface- Jupyter Notebook

Libraries- Tensorflow and Keras for backend.

System Architecture – Miro for designing.

Other Libraries – pandas,Matplotlib