

Methods and techniques used

Long short-term memory are the upgraded versions of RNNs and help solve the vanishing gradient problem by making use of gates in each cell of the network. These gates help differentiate which data in the long sequence is important to retain and which one to forget and hence uses the relevant information to help make the prediction. we will be making use of LSTM model for text generation to create poetry as only this architecture helps to make use of the words trained for the neural network and generate meaningful poetry using the gates hidden inside in the LSTM model. It helps retain the necessary information from the long sequence of inputs to train it. The model has been implemented using Tensorflow and keras library in google colab.

This class allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count.

- 1) `texts_to_sequences` -Transforms each text in texts to a sequence of integers.
- 2) `pad_sequences`-This function transforms a list of sequences into a 2D Numpy array.
- 3) `to_categorical`-Converts a class vector (integers) to a binary class matrix.
- 4) `Embedding`-Turns positive integers (indexes) into dense vectors of fixed size.

The loss function used computes the categorical crossentropy loss which is for multiclass classification.

The optimizer function that we used implements the RMSprop algorithm.