Next word prediction using LSTM RNN

1. Data collection :nltk.download(‘gutenberg’) -🡪shakespeare-hamlet
2. Intialize tokenizer(keras pre processing text)🡪 tokenizer.fit\_on([text]) 🡪 text is “hamlet.txt”
3. Total\_words = len(tokenizer.word\_index)🡪 tokenizer.word\_index( gives me index for each word in vocab)
4. Creating an input\_sequence using texts\_to\_sequence for each line of text( on \n). Then add a pad sequence, so that all sent are equal len.
5. X= leaving last element , Y = last element ,training and test 🡪 since this is next word prediction unlike sentiment analysis , we cant keep just the last word as one number, we have to match it to embeddings of other words, hence we add zeroes( as in whereever that index id present will be 1, rest will be 0)
6. y=tf.keras.utils.to\_categorical(y,num\_classes=total\_words)
7. total\_words=len(tokenizer.word\_index)+1
8. model=Sequential()
9. model.add(Embedding(total\_words,100,input\_length=max\_sequence\_len-1))
10. model.add(LSTM(150,return\_sequences=True))
11. model.add(Dropout(0.2))
12. model.add(LSTM(100))
13. model.add(Dense(total\_words,activation="softmax"))

100 = dimension of output vector input length( max length of one sentences) , max\_sequence\_len in here could be ki how much of sentence len of max could be passed as input for next word pred.