**Sentence Correction using RNN’s(Deep learning)**



**Sentence Image**

**Contents:**

1. Introduction
2. Business problem
3. Map to Machine Learning problem
4. Source Dataset
5. Exploratory Data Analysis (EDA)
6. Preprocessing
7. Tokenization and Padding
8. Deep Learning Models
9. Beam Search Decoder and Inference Setup
10. Final Pipeline

**1.Introduction**

The information exchange between people rapidly increasing through different social media channels (Facebook, Twitter etc.).People started texting short form of messages and using different terminologies to convey the information. This short form of messages(Fast text) creating big impact and drastically reducing the performance of machine learning models which runs on this data. And it slowly led to make wrong predictions and wrong interpretation from ML models. **Gene Lewis** published a research paper from Stanford University and proposed an approach to overcome this problem in the field of AI.

***Research paper:*** <https://cs224d.stanford.edu/reports/Lewis.pdf>

**2.Business Problem**

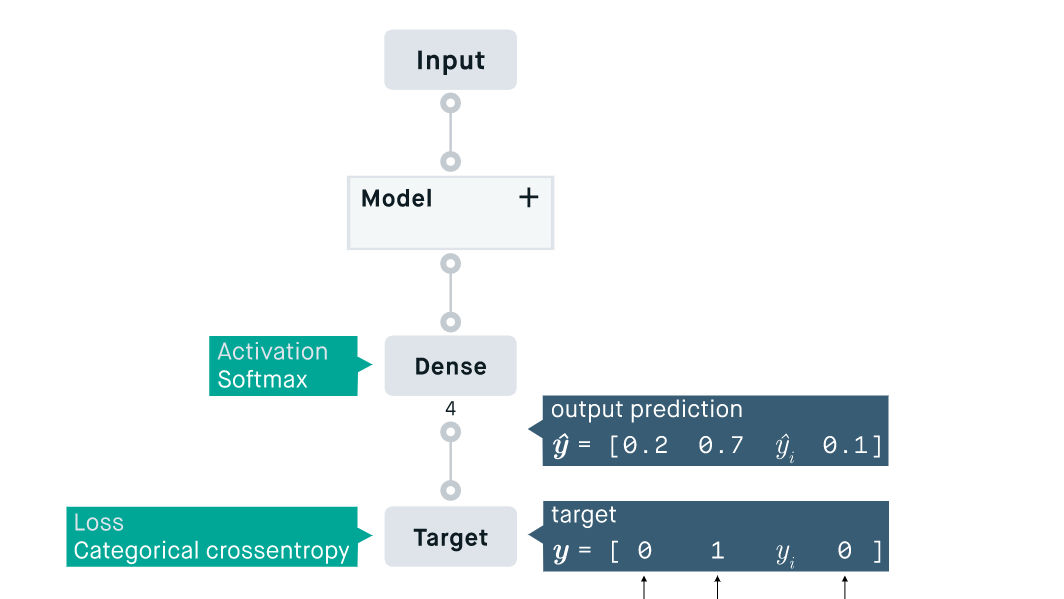
Building a deep learning model that helps in correcting the corrupted text in social media by preserving the semantic meaning of the sentence. Once the correction made ,it helps in increases the performance of the machine learning models which runs on the social media data. Example: Sentiment analysis, search engine building on social media text data etc.

**3.Map to Machine Learning Problem**

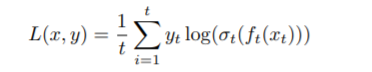
In this case study i am going to use different types of recurrent neural networks (RNN’s) for translation of corrupted or fast text to normal English text. This is a type of Neural machine translation problem.

***Loss Function:***

Used categorical cross entropy as loss function. Cross-entropy loss increases as the predicted probability diverges from the actual class. As i am using soft max classifier as last layer of neural network, using categorical cross entropy helps in interpreting the predicted probabilities with actual probabilities.



**Categorical Cross Entropy**

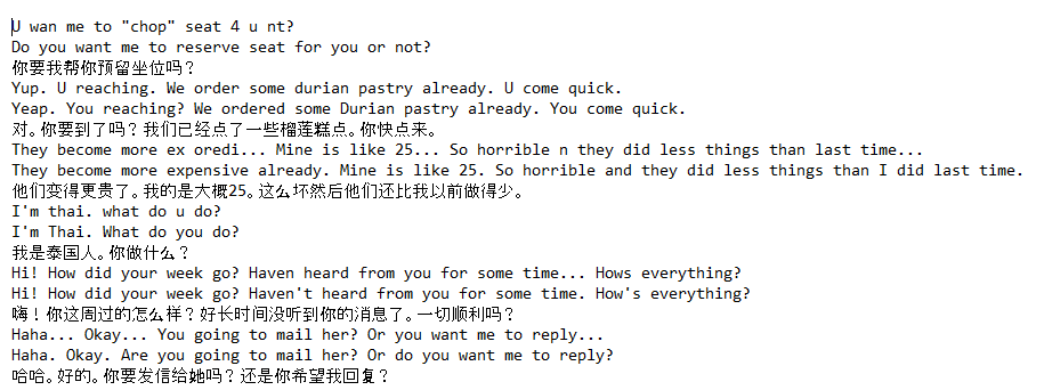


**Formula**

**4. Source Dataset :**

The Source dataset contains three types of texts corrupted text, Chinese text and Original English Text. The Overall points present in source data file is 2K points. This data may not be sufficient to get good accuracy of the model. To increase the dataset size, i am going to perform Synthetic data generation .

***Source Dataset:*** <http://www.comp.nus.edu.sg/~nlp/sw/sm_norm_mt.tar.gz>

**Source Dataset**

***Synthetic Data Generation (Artificial Data)***

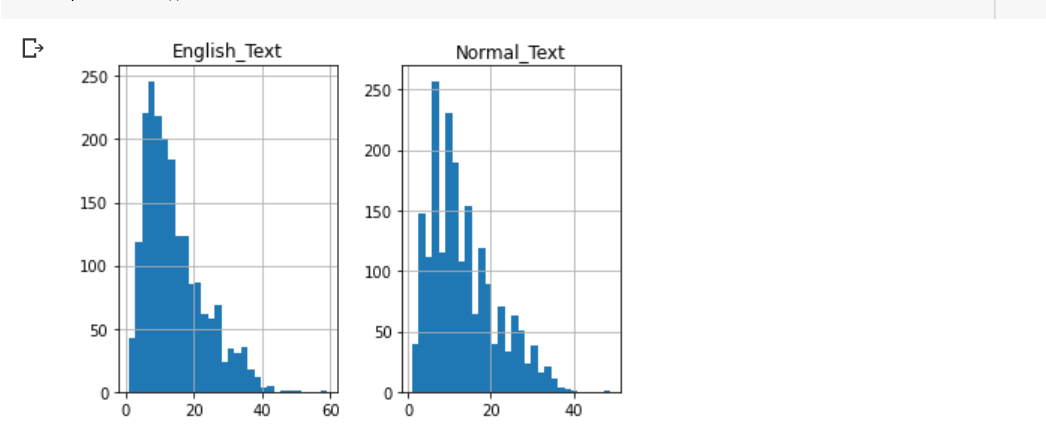
Training a neural network with insufficient data may not help in achieving the good accuracy. As dataset size is very small, generated synthetic data using nlpaug library. Used synonym augmentation to keep the semantic meaning of the sentence along with fast text word embedding model to generate the artificial data.

Gits Link: https://gist.github.com/Madhusudhan77/5e9488bc551166323719ab1ca87b5d43

**5. Exploratory Data Analysis**

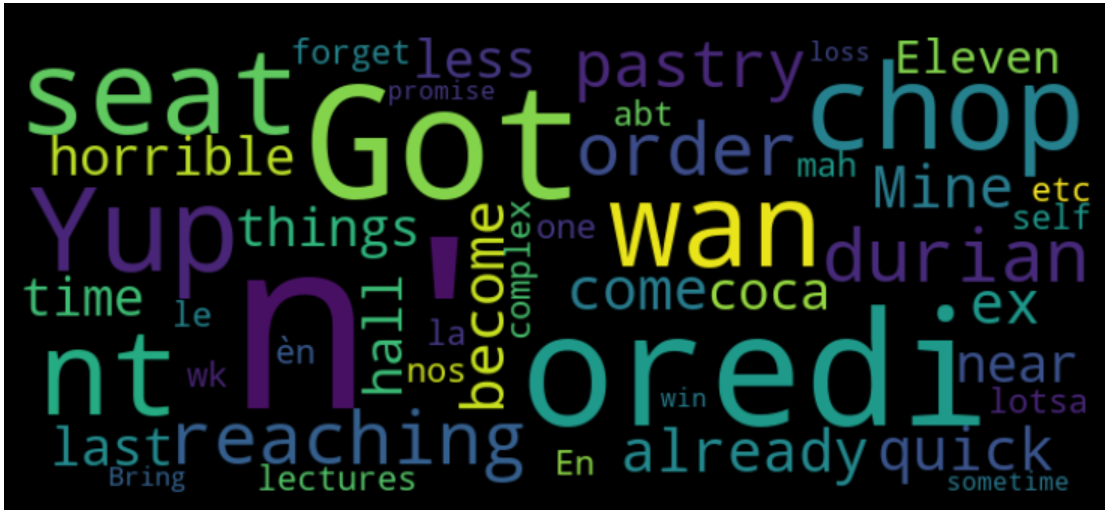
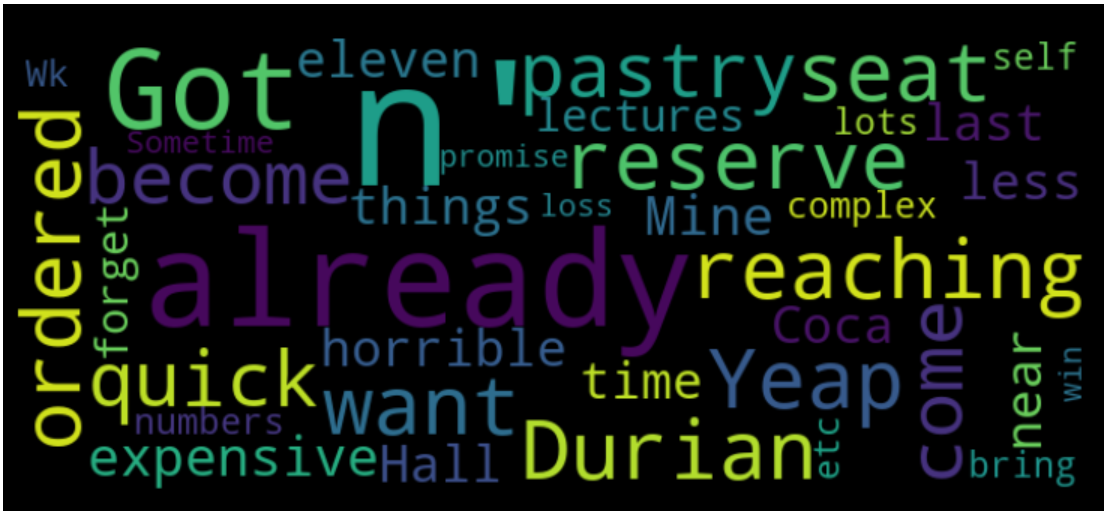
In this step i performed EDA on source dataset. Verified any missing values or duplicates present in source. The max length of the Normalized text and original English text is 49 and 50.

Gist Link: https://gist.github.com/Madhusudhan77/d507fd061ec4ce79361efe7ab7f0e78e

Max Length of the Sentence**Histogram Plot**

From the above plot most of the sentences lies between 1 to 40.The region between 1 to 40 is more dense in both plots.

**Word clouds:** Plotting word clouds on both normal text and original text to see the words which occurred frequently in source dataset. In word cloud ,words which occurred more frequently having more weight and same shown in below diagram.

**Word Cloud on Normal Text****Word Cloud on English Text**

**6.Preprocessing**

Preprocessing of the data is one of the most important steps to clean the dataset before training the ML models. It involved in removing special characters (‘[$)\?”’.°!;\’€%:,(/]” .) and de-contracted the strings present in dataset.

**Gist Link:** https://gist.github.com/Madhusudhan77/74fa1024844935b7fe5990a30105b8c9

**7. Tokenization and Padding**

Once the dataset has been preprocessed, i am going to transform this dataset into model understandable format. I am going to add <start> and <end> tag for every source sentence, this tags help my encoder and decoder model to identify the starting and ending of the sentence while encoding and decoding the sentence.

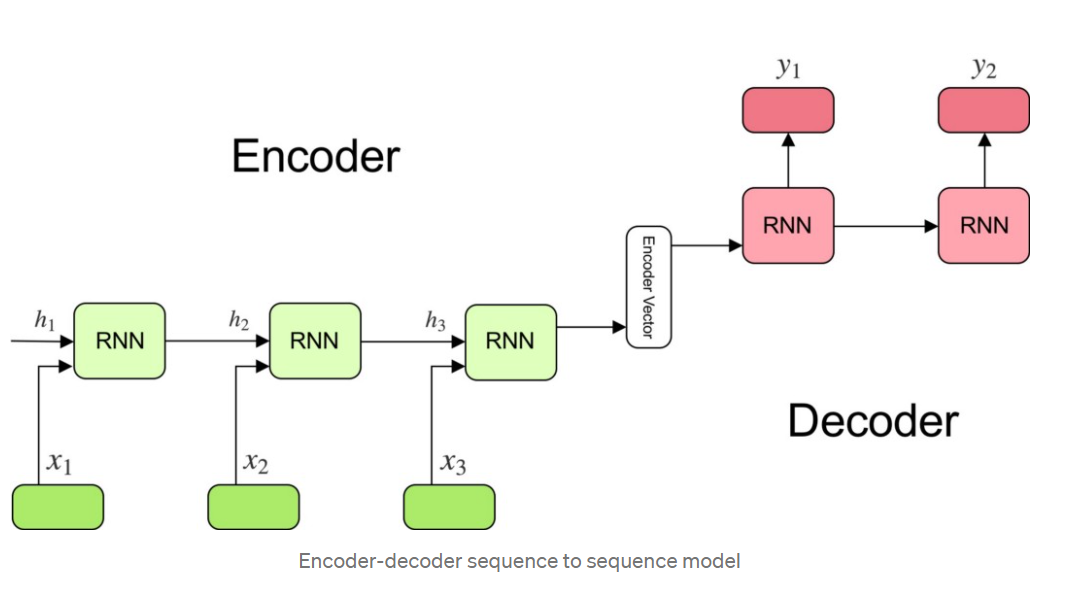
The tokenization involved in converting sentences to token and form the vocabulary. Each word present in sentence convert into tokens.

**Gits Link:** <https://gist.github.com/Madhusudhan77/d8c923291fe79074ade858d4227b1b85>

**8.Deep Learning Models**

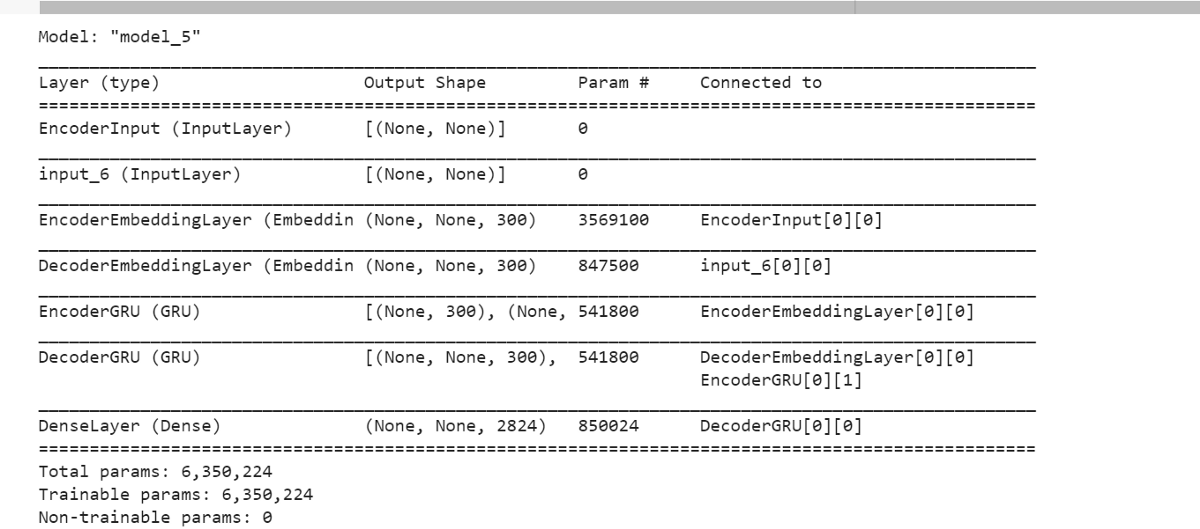
**Encoder and Decoder Architecture:**

Machine translation is a major problem domain for sequence transduction models, whose input and output are both variable-length sequences. To handle this type of inputs and outputs, we can design an architecture with two major components. The first component is an *encoder*: it takes a variable-length sequence as the input and transforms it into a state with a fixed shape. The second component is a *decoder*: it maps the encoded state of a fixed shape to a variable-length sequence. This is called an *encoder-decoder* architecture



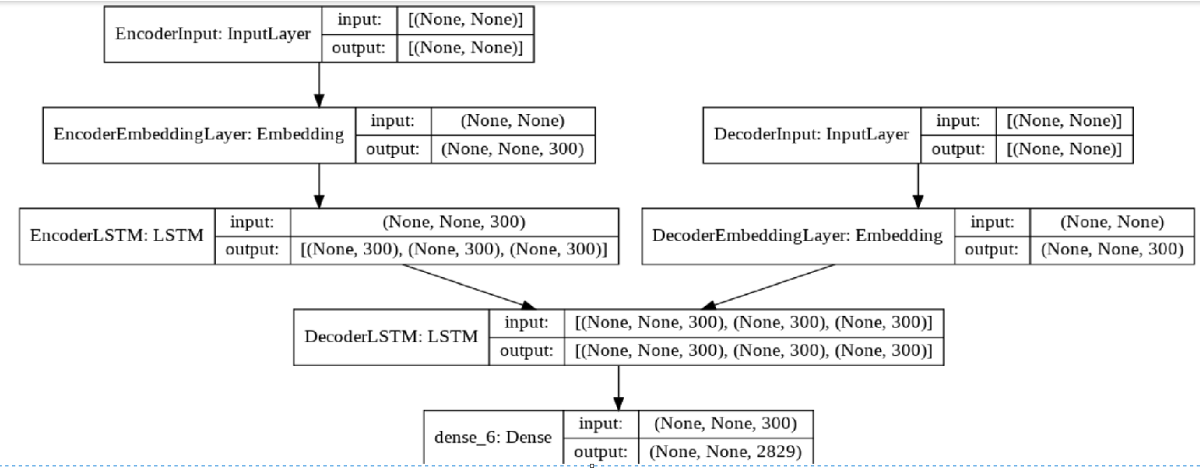
1. ***Simple Encoder and Decoder with GRU***
2. ***Simple Encoder and Decoder neural network with LSTM***
3. ***Encoder and Decoder with Bahdanau’s Attention***

***1.Simple Encoder and Decoder with GRU :***Tried simple encoder and Decoder with GRU. I used fast text embedding weights and 300 GRU units and trained this neural network with 200 epochs. The accuracy of the model is slowly increasing but this neural network not giving the results as expected. The accuracy of the model after trained with 200 epochs is 0.0651 and loss is nan.

**Encoder and Decoder with GRU**

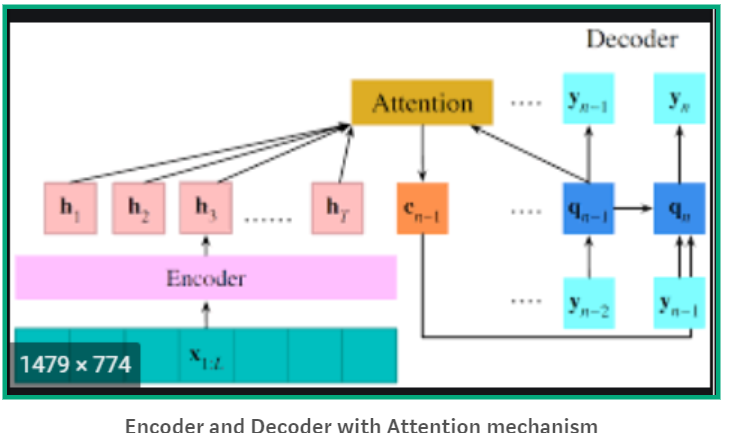
***2.Simple Encoder Decoder neural network with LSTM:***

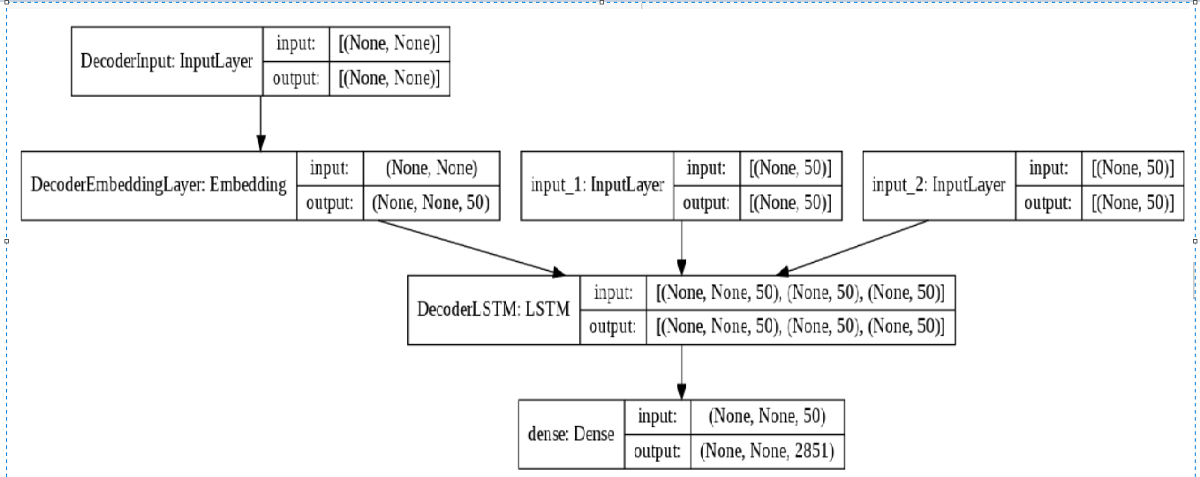
Tried simple encoder and decoder with 300 LSTM units and tanh as activation function. Used soft max as last layer of neural network to get the output probabilities for good interpretation. Trained neural network with 50 epoch's and at the end of the training, got trained accuracy(85%) and loss(0.2602%).

**Encoder and Decoder with LSTM**

***3. Encoder and Decoder with Bahdanau’s Attention***

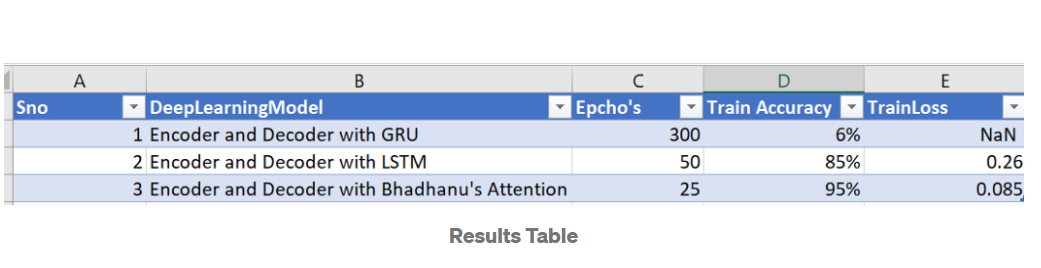
Designed Encoder and Decoder with different attention mechanism's and trained neural network. Encoder and Decoder with Bahdanau’s attention with one step decoder given good train and test accuracy. Used LSTM with 300 units and trained model with 25 epoch's and got 95% train accuracy and cross entropy loss 0.0850.





**Encoder and Decoder with Attention model Diagram**

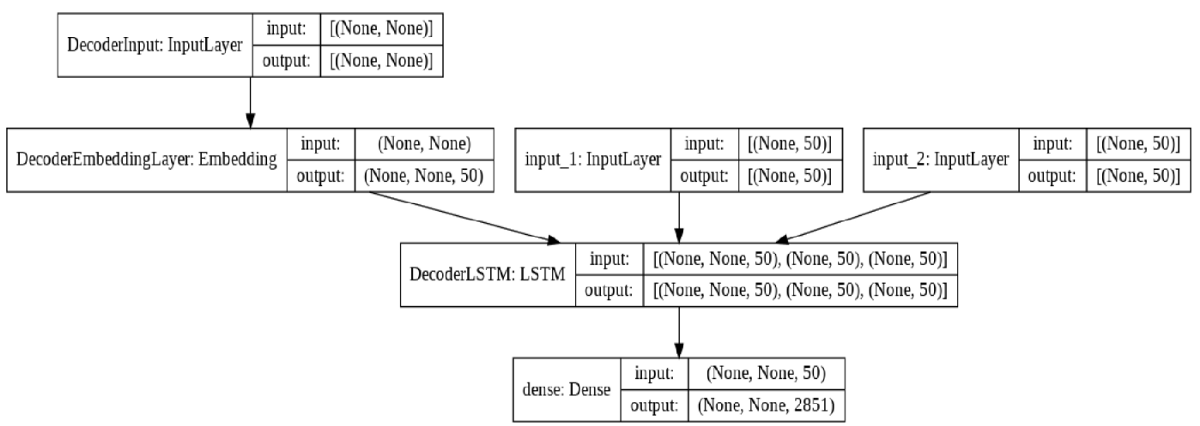
**Results Table:**



**9.Beam Search Decoder and Inference Setup**

**Inference Setup**

Setup inference model to predicate the sentences. Here i am sending multiple input sentences and inference model able to predict output sentences.

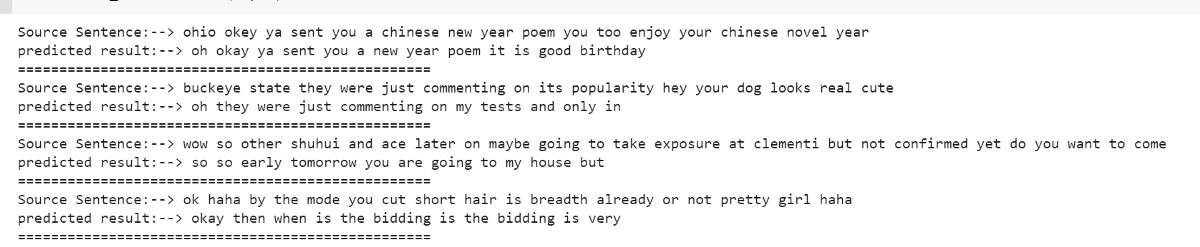
**Inference Model**

**Beam Search**

NMT is using a simple left-to-right beam-search decoder to generate new translations that approximately maximize the trained conditional probability. The current beam search strategy generates the target sentence word by word from left-to- right while keeping a fixed amount of active candidates at each time step. Beam search decoder increased the performance of Encoder and Decoder model and helped in making good language translation.

**Gits Link:** https://gist.github.com/Madhusudhan77/3cd7926b67fe76188ba823f48aece6f6

**Top 5 predicted Sentences:**

Predicted Sentences

**11.Final Pipeline:**

Youtube link:https://youtu.be/idUqeucjNM8

**Future Works**

1. As source dataset size is very less(2K), can use more augmentation technics and NLP mechanism's to generate the artificial data .
2. There might be a possibility to use different RNN’s to increase the performance .Ex : bidirectional LSTM’s.

**Reference Links:**

1. <https://www.appliedaicourse.com/>
2. <https://arxiv.org/abs/1702.01806>
3. <https://machinelearningmastery.com/beam-search-decoder-natural-language-processing/>

**Git code:**

<https://github.com/Madhusudhan77/Sentence-Correction-Using-RNN-s>

**LinkedIn:**

[www.linkedin.com/in/reddivari-madhusudhan-42a002195](http://www.linkedin.com/in/reddivari-madhusudhan-42a002195)