Advanced Natural Language Processing

[CS7.501]

ASSIGNMENT 3 - REPORT

Rahothvarman P

2020114008

Tuesday, 15 November, 2022

2 Analysis :

2.1

Since only 30K sentences were used from the CNN/Daily Mail dataset, the model was only able to produce rouge scores averaging around 0.01 (between 0 and 1) for the sentences in test datasets. Also only the indices of the input sentences were given as input embeddings to the encoder while training as mentioned in the paper. The low scores could be attributed to the lack of pre-trained embeddings too.

2.2

The negative log likelihood loss function wasn't a proper metric to calculate the optimal hyperparameters. Hence, the summaries were manually checked for various hyperparameters. The hyperparameters are reported below :

1. No. of Epochs :

Loss scores were plotted for each epoch and variation was noted.

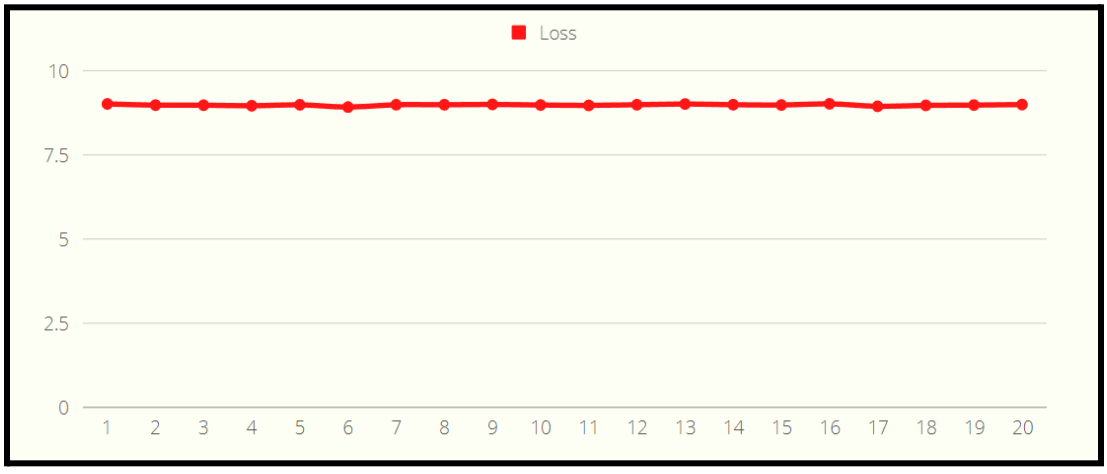

FIG 1

With the increase of epochs, the summaries became better, but due to computational restraints, no. of epochs were kept at **20**.

2. Article and Summary lengths :

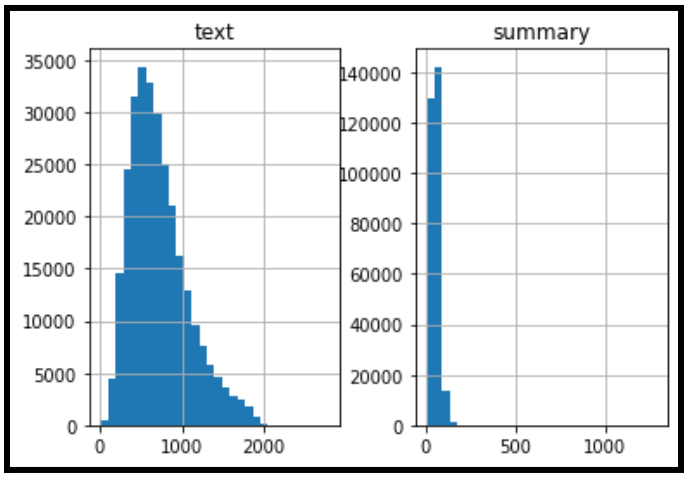

FIG 2

Choosing maximum article and summary lengths was an important parameter to the model. Since, 30K sentences were required to train the model, the article-summary pairs were chosen in such a way that, **max-article length was 512** and **max-summary length was 256**.

3. Learning rates :

Loss scores were plotted for different learning rates and variation was noted. A learning rate of 0.001 was chosen.

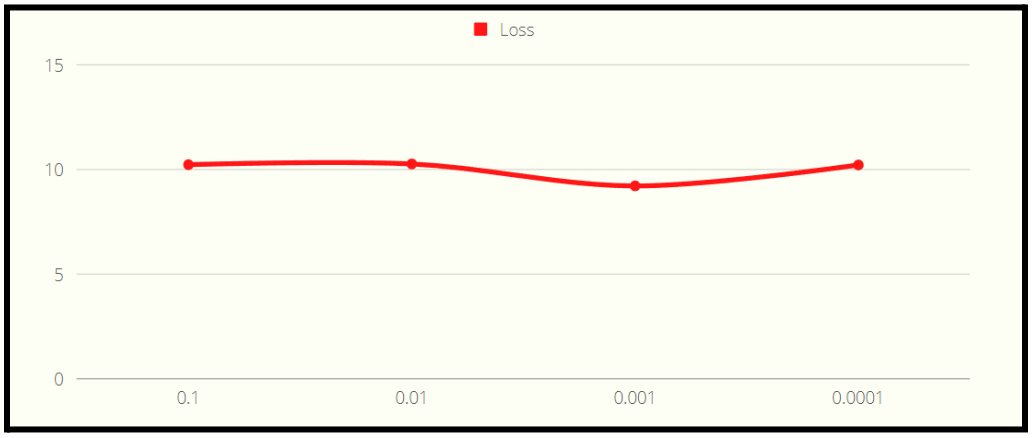

FIG 3

2.3 (a) OOV words refer to Out-Of-Vocabulary words. Pointer-Generator is made up of two words, pointer and generator. The generator refers to the decoder distribution. While the pointer points the model to the input distribution. Since, the decoder can only generate words from its vocabulary, sometimes the target word may not exist in its vocabulary hence, it generates an unknown token “UNK”. A pointer-generator concept is used to overcome this problem. It maintains a probability **Pgen**, which represents how much of the **decoder's output distribution** to consider while generating the output. On the other hand, (1 - Pgen) represents the amount of **input distribution** (in this case the **attention vector** of that time step) to consider while generating an output. When the model prefers the input distribution over the decoder’s output distribution, it essentially **copies from the input**. This can very well handle out of vocabulary words.

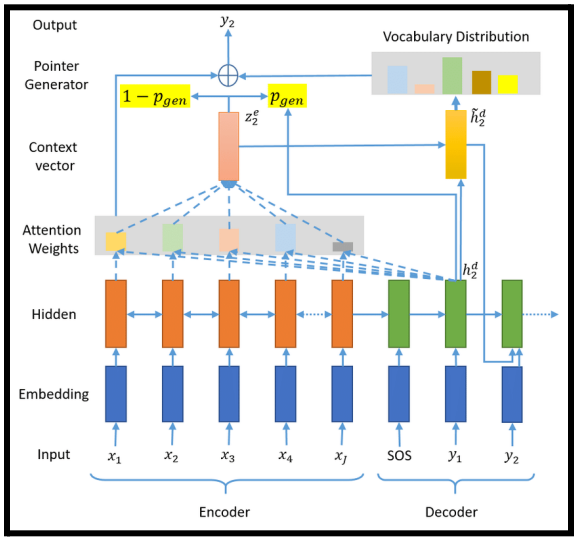

FIG 4

The attention model maintains an extended vocabulary which consists of the input vocabulary and OOVs of the input sentence. It uses a multilayer perceptron to calculate attention from the encoder outputs and the current decoder hidden state. It returns

the probability distribution of the input sequence by applying softmax to the neural network output.

2.3 (b) Repetition problem refers to generating the same word again and again by the decoder of Seq2Seq models. This is due to the fact of using **auto-regressive** decoders in Seq2Seq models. In auto regressive decoders, the hidden states tend to be closer or similar to each other across consecutive time steps. Since, the attention mechanism generates context vectors based on the decoder hidden state and the input sequence, the context vector tends to be the same. Hence, the same word is generated again and again. However, this problem is not that prominent in the decoders of the transformers since their decoder architecture is different to that of the Se2Seq models.

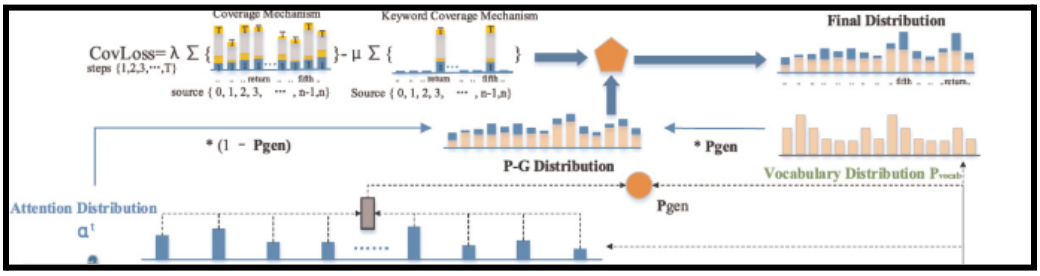

FIG 5

If there was a way to inform the model that it has paid enough attention to a particular word in the input sequence, this problem of repetition can be solved. This is exactly what the coverage mechanism does. **Coverage mechanism** maintains a **cumulative attention vector** or in simple terms, the attention vector produced at each time step of the decoder is summed up. Hence, the cumulative attention vector at each step would be the summation of attention vectors at previous time steps. This brings in some kind of **regularization** and forces the decoder model to uniform attention over all the words of the input sequence.

1.4 Bonus : (1) Coverage and (2) Model Analysis :

Baseline Seq2Seq :

Summary: world no . novak djokovic beaten in the second round of the paris masters . the serb who had a first round bye lost to veteran fabrice santoro . top seeds roger federer and rafael nadal win opening matches at bercy .

Generated Summary: the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the

Rouge L - f : 1.0

Seq2Seq + Pointer-Generator :

Summary : empty anti tank weapon turns up in front of new jersey home . device handed over to army ordnance disposal unit . weapon not capable of being reloaded experts say .

Generated Summary : lenient fragility up satin foiled nano civilization division suvarnabhumi mccurry happened observance yvonne lottery now defunct yousif oppmann migration dengue sidner organizations satin idlib changing suvarnabhumi quantico yeardley amended wozniacki departed

Rouge L - f : 0.034482758620689655

Seq2Seq + Pointer-Generator + Coverage :

Summary: if you build it the tourists will come to your museum . museums for hobos medical oddities and trash . kentucky museum is where dummies go to die

Generated Summary: kashi sniper fights welsh engaged your costly myrtle manila goalkeeper yang internationally easy cowardly deliberate sadly listed obligations midst consumption frontier sgarbi trophies prisoners certainly ann amazing resulting consumption

Rouge L - f : 0.03571428571428571

From the above examples, it can be clearly seen that ROUGE Metric is not an appropriate metric for summarization.

Inference :

Baseline Seq2Seq :

- Repetition of the same word is evident from the outputs
- Doesn’t make any sense both syntactically and semantically

Seq2Seq + Pointer Generator :

- Repetition has been decreased although few words / phrases do repeat
- Although the words generated themselves don’t have any relation between

Seq2Seq + Pointer Generator + Coverage :

- No repetition can be observed.
- The words generated themselves do have some relation between
- Phrase level syntax and semantics can be clearly observed although the entire sentence still doesn’t make sense

Link to the Pointer-Generator Network