Paper Review

Title: PEGASUS: PRE-TRAINING WITH EXTRACTED GAP-SENTENCES FOR ABSTRACTIVE SUMMARIZATION

by
Anurag Saraswat (M20CS066)
Parsa Revanth (M20CS058)

Date: May 23, 2021

1 ARCHITECTURE USED IN THE PAPER

The architecture is the sequence to sequence model based on the transformers as encoder and decoder.

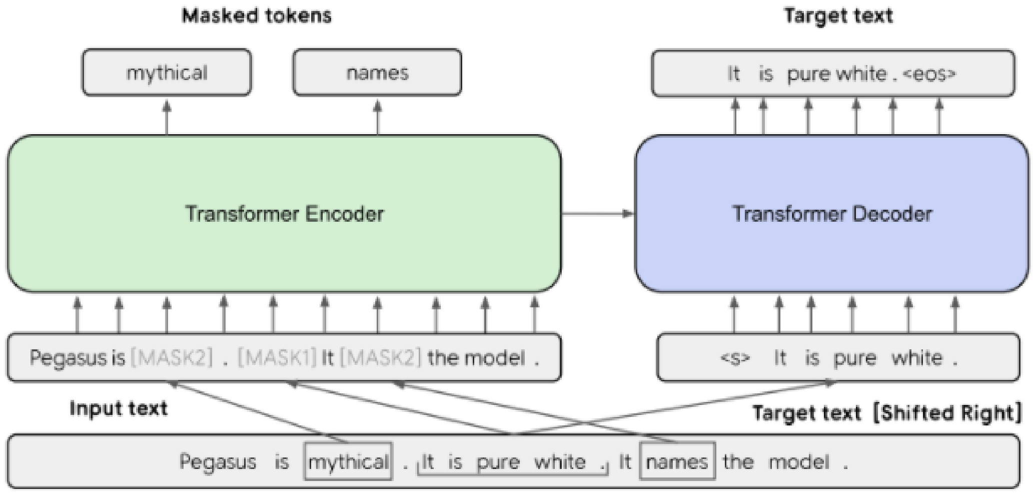
Masked tokens
Target text
mythical
names
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] It [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

Fig. 1. Transformer based encoder-decoder architecture

2 SUMMARY

The authors proposed preparing a large transformer on a massive text corpus with the self supervised objective. The massive corpus are the C4 dataset and Hugenews dataset. The C4 dataset has a text corpus of 350 million webpages and size of 750GB. Hugenews dataset has a text corpus of 1.5 billion news articles and size of 3.8TB. The architecture is the transformer based encoder-decoder architecture. The data set is created to achieve the objective by gap sentence generation (GSG) and masked language model (MLM).

In gap sentence generation, the principal approach is to calculate the Rouge1-F1 measure for a sequence in a document to the rest of sequences in the document. In this way for all the sequences the Rouge1-F1 is calculated and the maximum Rouge1-F1 is taken as the important sentence and is masked in the document and given as the target sentence to the decoder. From the remaining document the masked language model [1] is created by taking top-m Rouge1-F1 score sequences and masking them and passing the data to the encoder.

The authors trained by creating the datasets accordingly for C4 and Hugenews datasets and trained both the encoder and decoder. This is the pre-training of gap sentence generation and later performed the abstractive summarization task on the 12 datasets which achieved the state of the art results. They calculated the Rouge1-F1 measure, Rouge2-F1 measure and RougeL-F1 measure for the evaluation of the 12 datasets. [2].

3 EXPERIMENTS

The 12 datasets are XSum, CNN/DailyMail, NEWSROOM, Multi-News, Gigaword, arXiv, PubMed, BIGPATENT, WikiHow, Reddit TIFU, AESLC, BillSum. Each dataset contains millions of documents.

The authors evaluated the pretrained transformer architecture on C4 and separately evaluated the pretrained transformer architecture on Hugenews on four datasets XSum, CNN/DailyMail, WikiHow and Reddit TIFU. The Rouge based metric is the average of the sum of the ratios of Rouge score computed on pretrained architecture to that of non-pretrained architecture. The two news datasets Xsum and CNN/DailyMail yielded better Rouge based metric on Hugenews pretrained architecture and the non-news datasets WikiHow and Reddit TIFU yielded better Rouge based metric on C4 pretrained architecture.

The number of masked sentences in the document is taken as 30% which is validated based on the Gap sentence ratio (GSR) ratio computed. The lower the GSR the less challenging and computationally efficient. They computed the Rouge scores by taking GSR's as 15%, 30%, 45%, 50%, 60%, and 75%. The authors prove that 30% is the best by calculating the Rouge scores for each model.

The authors used sentence piece unigram algorithm (Unigram) for tokenization and the vocabulary length of 96K. The size of the vocabulary and the tokenization technique are also evaluated for vocabulary lengths of 32K, 64K, 96K, 128K and 256K based on the Rouge scores. The vocabulary length of 96K got the best Rouge score. Byte pair encoding algorithm was also used to pick the tokenization algorithm but Unigram produced a better Rouge score.

The authors compared the Rouge1-F1, Rouge2-F1 and RougeL-F1 scores on the 12 datasets with respect to the same metrics calculated using the different architectures. The metrics achieved are the state of the art. Because the model is trained on data which is so huge the downstream tasks on this pretrained model will perform better.

4 CRITICAL DISCUSSION

The authors did not mention the amount of time taken to train the model on C4 and Hugenews and also not mention the details of the Graphics processing unit (GPU) used. When we give the summary which is just 4-5 sentences created is extractive rather than abstractive because the training objective is mainly extractive. Summarizing bigger text hides this fact, but for smaller text this problem is clear. The objective of abstractive summarization is achieved in the document with many sequences.

5 CONCLUSIONS

The paper's content is precise and the authors have the following contribution. The model designed is tailor made for the abstractive summarization task. The authors used a sequence to sequence model with the gap sentence generation as the pre-training objective and performed the abstractive summarization task on the 12 benchmark datasets. Also achieved the near state of art results by training just the 1000 samples of the new dataset. Overall paper presented a model which makes much more sense for generating summary than previous models and opens up a new dimension of exploration.

REFERENCES

[1] S. Rothe, S. Narayan, and A. Severyn. Leveraging pre-trained checkpoints for sequence generation tasks, 2020.

[2] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.