since 1995

Data Science Natural Language Processing NLP Papers Summary

Day 174: NLP Papers Summary – PEGASUS: Pre-Training With Extracted Gap-Sentences For Abstractive Summarization

By Ryan 22nd June 2020 No Comments

Objective and Contribution

Proposed PEGASUS, a large Transformer-based model pre-trained on a new self-supervised objective that's related to abstractive text summarisation. This new self-supervised objective involves masking important sentences from the input document and are generated together as one output sequence from the remaining sentences. We applied our PEGASUS to 12 different summarisation datasets and achieved SOTA performance in all datasets. PEGASUS lows us to perform low-resource summarisation, allowing our fine-tuned model to surpass previous

SOTA results on 6 datasets with only 1000 data samples. Finally, our generated summarie so achieve human performance on multiple datasets.

The contributions of the paper are as follows:

1. Proposed a new pre-training objectives known as Gap Sentences Generation (GSG)
2. Achieved SOTA results on 12 different summarisation datasets
3. Shown good cross domain summarisation and surpassing previous SOTA results with as little as 1000 data samples
4. Achieved human-level summarisation performance on XSum, CNN/DM, and Reddit TIFU dataset

PEGASUS

It’s different than previous language models such as UniLM, T5, and BART in the sense that it masks multiple whole sentences rather than short text spans. We also chose to mask important sentences rather than randomly as done in previous work. Our pre-training corpus is Colossal Cleaned Common Crawl (C4) and HugeNews, both huge dataset of articles.

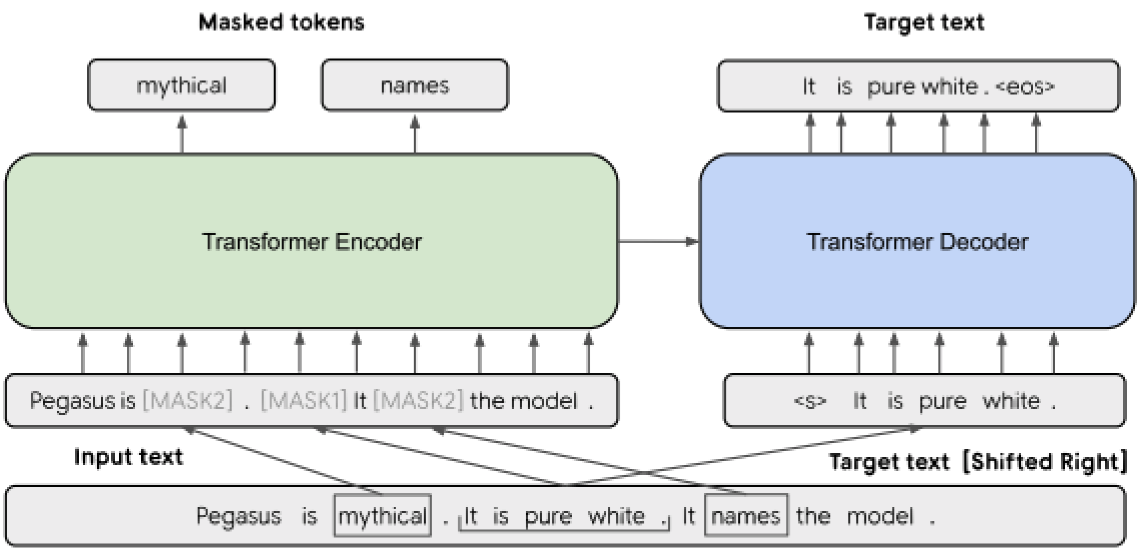
Masked tokens
mythical
names
Target text
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] It [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

Figure 1: The base architecture of PEGASUS is a standard Transformer encoder-decoder. Both GSG and MLM are applied simultaneously to this example as pre-training objectives. Originally there are three sentences. One sentence is masked with [MASK1] and used as target generation text (GSG). The other two sentences remain in the input, but some tokens are randomly masked by [MASK2] (MLM).

GAP-SENTENCE GENERATION (GSG)

We believe that using a pre-training objective that's more inline with the downstream task would lead to better and faster fine-tuning performance. Our new pre-training objective GSG select and mask whole sentences from documents. Those masked sentences would then be concatenated to form pseudo summaries. We also computed the gap sentences ratio which computes the number of selected gap sentences over the total number of sentences in the document. We experimented with three strategies to select gap sentences (without replacement):

1. Random
2. Lead
3. Principal

Principal strategy involves selecting top-m scored sentences based on their impor Our proxy for importance is measured by the ROUGE-1 score between the sentence and the rest of the document. There are four variants to the principal strategy:

1. Sentences are scored independently
2. Sentences are scored sequentially, maximising ROUGE-1 score between selected sentences and remaining sentences
3. Computing ROUGE-1 score by considering n-grams as a set (a. unique n-gram) instead of double counting identical n-grams (b. original method)

We apply Masked Language Model (MLM) to train the transformer encoder, either with or without GSG.

Experiments

DOWNSTREAM DATASETS

We used TensorFlow Summarisation Datasets to process and access 12 different summarisation datasets:

1. XSum
2. CNN/DM
3. NEWSROOM
4. Multi-News
5. Gigaword
6. arXiv
7. PubMed
8. BIGPATENT
9. WikiHow
10. Reddit TIFU
11. AESLC
12. BillSum

ABLATIONS ON PEGASUS-BASE

We use PEGASUS-BASE and normalised ROUGE scores to determine the optimal choices of pre-training corpus, objective, and vocabulary size. The results are displayed in the three figures below. Firstly, we found that pre-training language models with similar domain datasets as the downstream tasks yield better summarisation results. As shown in figure 3, pre-training with HugeNews corpus led to better performance in downstream datasets of XSum and CNN/DM as they are both news datasets whereas both WikiHow and Reddit TIFU benefit more from pre-training language model with C4 dataset.

We evaluated six variants of GSG with 30% GSR. Figure 4a suggests that independently scoring sentences using the original method (ind-orig) achieved the best performance. Seq-uniq achieved the second best performance. Random and lead method underperform consistently against all the variants of principal method. As expected, lead method performed well on the two news datasets but performed badly in the two non-news datasets. We selected principal method, specifically ind-orig variant to train our PEGASUS-LARGE. In addition, we also evaluated whether we show train MLM with or without GSG. Figure 4a shows that MLM doesn't improve the performance of Ind-Orig variant and also doesn't improve fine-tuning performance with long training steps and so we decided to exclude the MLM. An important hyperparameter of GSG is GSR, which determines how many sentences to mask. We compared the performance of our downstream datasets across different level of GSR. Figure 4b shown mix results in terms of the optimal GSR, however, the best performance GSR is always under 50%. We decided to choose the GSR of 30% when scaling to PEGASUS-LARGE.

In terms of vocabulary size and type, we evaluated between byte-pair encoding and SentencePiece Unigram algorithm. Figure 5 below shows that Unigram outperformed BPE in all datasets with 96K vocab size achieving the best ROUGE scores in most datasets.

Overall the ablations study allows us to set the best choices to scale our PEGASUS model. Our PEGASUS-LARGE uses GSG (Ind-Orig) without MLM as pre-training objective, SentencePiece Unigram vocabulary size of 96K, and pre-train on C4 and HugeNews corpus separately. In addition, to encourage our PEGASUS to copy, 20% of selected sentences were left unchanged instead of mask in the original input. This also led us to increasing GSR to 45% in order to achieve the optimal GSR ratio found in our ablation studies.

Results

Table 1 showcase the performance improvements between PEGASUS-BASE and PEGASUS-LARGE on all 12 downstream datasets. PEGASUS-BASE alone was able to outperformed many previous SOTA results and PEGASUS-LARGE was able to achieve SOTA results in all but WikiHow downstream tasks using HugeNews alone. PEGASUS-LARGE trained on C4 was able to achieved SOTA results on WikiHow. We also show that we have a strong improvem in comparison to Transformer-BASE especially in smaller datasets.

ZERO AND LOW-RESOURCE SUMMARISATION

In reality, it's often difficult to fine supervised datasets to fine-tune our language model. Therefore, we decided to fine-tune our PEGASUS-LARGE using 0 (zero shot), 10, 100, 1000, and 10000 samples from each dataset and evaluate the performance. The results are displayed below. PEGASUS-LARGE was able to beat previous SOTA results with only 1000 fine-tuning examples in 6 summarisation datasets. PEGASUS-LARGE also perform better zero-shot learning on CNN/DM dataset when compared to GPT-2.

HUMAN EVALUATION

The table below displayed the results of our human evaluation comparing generated and human-written summaries. Human are asked to rate summaries on 1 – 5, with higher being better. Our results show that both our PEGASUS-LARGE pretrained on C4 and HugeNews respectively were able to generate summaries as good as the reference summaries. In addition, even with low fine-tuning samples, PEGASUS-LARGE was able to perform relatively well when compared to human summaries.

Conclusion and Future Work

Overall, we found that principle sentence selection is the optimal gap-sentence selection methods and we are able to achieve SOTA results in 12 different summarisation datasets and generated summaries close to human performance on multiple datasets.

Source: **https://arxiv.org/pdf/1912.08777.pdf**

Ryan

Data Scientist

Previous Post

Day 173: NLP Discovery - Text-To-Text Transfer Transformer (T5)

Next Post

Day 175: NLP Papers Summary - GPT-3 : Introduction and Context

Data Science Natural Language Processing NLP Papers Summary

Day 365: NLP Papers Summary – A Survey on Knowledge Graph Embedding

Ryan
30th December 2020

Data Science Implementation Natural Language Processing

Day 364: Ryan's PhD Journey – OpenKE-PyTorch Library Analysis + code snippets for 11 KE models

Ryan
29th December 2020

Data Science Ryan's PhD Journey

Day 363: Ryan’s PhD Journey – Literature Review – Knowledge Acquisition – 1st Passes XIV

Ryan
28th December 2020