Text Summarization using PEGASUS Transformer

Abstract:

Pre-training Transformers with self-supervised objectives on large text corpora has shown great success when fine-tuned on downstream NLP tasks including text summarization. However, pre-training objectives tailored for abstractive text summarization have not been explored. This work proposes pre-training large Transformer-based encoder-decoder models on massive text corpora with a new self-supervised objective. Important sentences from an input text are removed/masked in PEGASUS, and the remaining sentences are formed as one output sequence from the remaining sentences, similar to an extractive summary.

Architecture:

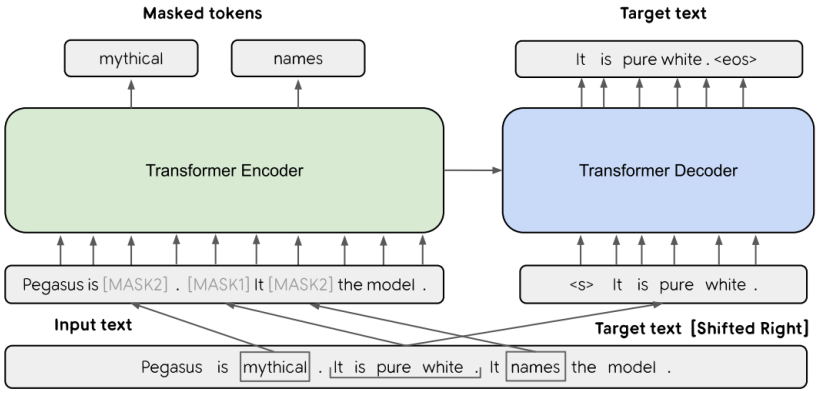


Figure: The base architecture of PEGASUS is a standard Transformer encoder-decoder

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).

Introduction:

Pre-training with **E**xtracted **G**ap-sentences for **A**bstractive **SU**mmarization **S**equence-to-sequence models. Important sentences from an input text are removed/masked in PEGASUS, and the remaining sentences are formed as one output sequence from the remaining sentences, similar to an extractive summary. Text summarization aims at generating accurate and concise summaries from input document(s). In contrast to extractive summarization which merely copies informative fragments from the input, abstractive summarization may generate novel words. A good abstractive summary covers principal information in the input and is linguistically fluent.

Contemporaneously, the adoption of Transformer models pre-trained using self-supervised objectives on large text corpora has improved performance on many NLP tasks. Recent work leveraging such pre-training for Transformer based sequence-to-sequence models has extended the success of text generation, including abstractive summarization. We find that masking whole sentences from a document and generating these gap sentences from the rest of the document works well as a pre-training objective for downstream summarization tasks. In particular, choosing putatively important sentences outperforms lead or randomly selected ones. We hypothesize this objective is suitable for abstractive summarization as it closely resembles the downstream task, encouraging whole-document understanding and summary-like generation. We call this self-supervised objective Gap Sentences Generation (GSG). Using GSG to pre-train a Transformer encoder-decoder on large corpora of documents (Web and news articles) results in our method, Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence models, or PEGASUS.

Pre-training Corpus:

- C4, or the Colossal and cleaned version of Common Crawl, introduced in Raffel et al. (2019); consists of text from 350M web pages (750GB).
- Huge News, a dataset of 1.5B articles (3.8TB) collected from news and news-like websites from 2013 to 2020. A whitelist of domains ranging from high-quality news publishers to lower-quality sites like high-school newspapers and blogs was curated and used to see a web crawler. Heuristics were used to identify news-like articles, and only the main article text was extracted as plain text.

Downstream Abstractive Summarization Datasets:

In addition to exploring the huge news datasets for pre-training, PEGASUS is going to explore a lot of different downstream abstractive summarization datasets.

List of Downstream Abstractive Summarization Datasets:

1. Xsum
2. CNN/DailyMail
3. NEWSROOM
4. Gigaword
5. orXiv, PubMed
6. BIGPATENT
7. WikiHow
8. Reddit TIFU

Pre-training Objectives:

We propose a new pre-training objective, GSG, in this work, but for comparison, we also evaluate BERT's masked language model objective, in isolation and in conjunction with GSG.

Gap Sentences Generation (GSG):

We hypothesize that using a pre-training objective that more closely resembles the downstream task leads to better and faster fine-tuning performance. Given our intended use for abstractive summarization, our proposed pre-training objective involves generating summary-like text from an input document. In order to leverage massive text corpora for pretraining, we design a sequence-to-sequence self-supervised objective in the absence of abstractive summaries. A naive option would be to pre-train as an extractive summarizer. However, such a procedure would only train a model to copy sentences, thus not suitable for abstractive summarization.

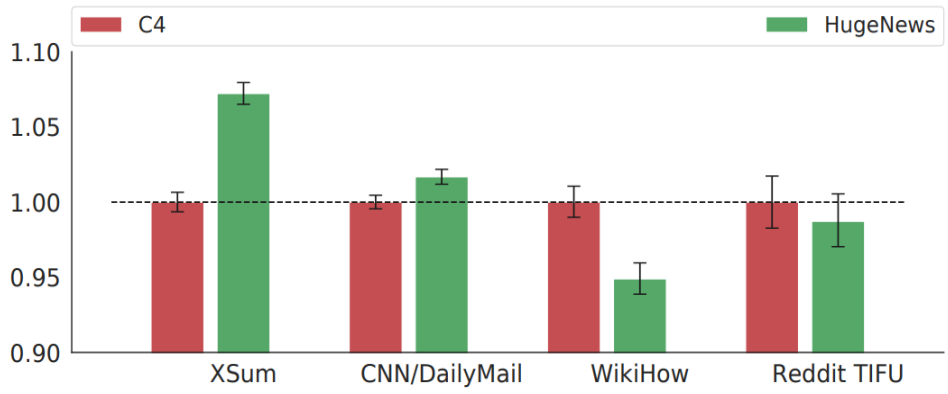
Inspired by recent success in masking words and contiguous spans, we select and mask whole sentences from documents, and concatenate the gap sentences into a pseudo-summary. The corresponding position of each selected gap sentence is replaced by a mask token [MASK1] to inform the model. Gap sentences ratio, or GSR, refers to the number of selected gap sentences to the total number of sentences in the document, which is similar to the mask rate in other works.

To even more closely approximate a summary, we select sentences that appear to be important/principal to the document. The resulting objective has both the empirically demonstrated benefits of masking and anticipates the form of the downstream task.

Masked Language Model (MLM):

Following BERT, we select 15% of tokens in the input text, and the selected tokens are 80% of the time replaced by a mask token [MASK2], or 10% of the time replaced by a random token, or 10% of time unchanged. We apply MLM to train the Transformer encoder as the sole pre-training objective or along with GSG.

PRE-TRAINING CORPUS:



Effect of pre-training corpus.

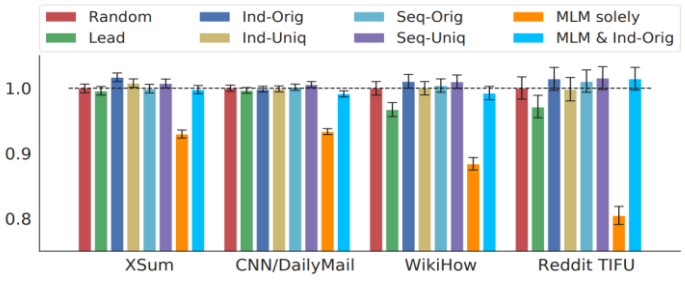
The graph shows that pre-training on HugeNews was more effective than C4 on the two news downstream datasets, while the non-news informal datasets (WikiHow and Reddit TIFU) prefer the pre-training on C4. This suggests that pre-training models transfer more effectively to downstream tasks when their domains are aligned better.

EFFECT OF PRE-TRAINING OBJECTIVES:

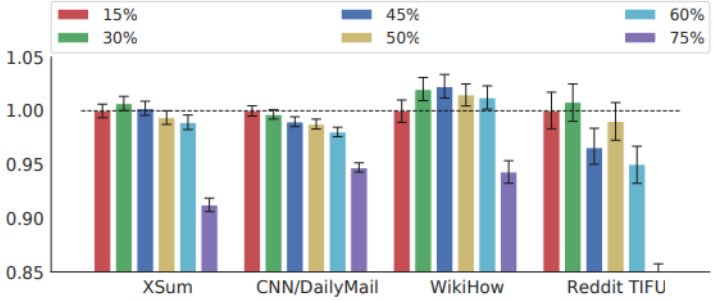
We compared six variants of GSG (Lead, Random, Ind-Orig, Ind-Umi, Seq-Orig, Seq-Uniq) while choosing 30% sentences as gap sentences. As shown in Figure 4a, Ind-Orig achieved the best performance followed by Seq-Uniq. Ind-Orig and Seq-Uniq were consistently better (or similar) than Random and Lead across the four downstream datasets. The lead had a decent performance on the two news datasets but was significantly worse on the two non-news datasets, which agrees with findings of lead bias in news datasets (See et al., 2017; Zhong et al., 2019). The results suggest choosing principal sentences works best for downstream summarization tasks, and we chose Ind-Orig for the PEGASUS.

A significant hyper-parameter in GSG is the gap-sentences ratio (GSR). A low GSR makes the pre-training less challenging and computationally efficient. On the other hand, choosing gap sentences at a high GSR loses the contextual information necessary to guide the generation. We compared GSRs from 15% to 75%. For a fair comparison, the original documents were truncated to have up to 400 words. The maximum input length in the encoder and the maximum target length in the decoder were set as 512 tokens.

Graph (b) shows that different downstream datasets had slightly different optima. The best performance always had GSR lower than 50%. The model with 15% gap sentences achieved the highest ROUGE scores on CNN/DailyMail, while XSum/Reddit TIFU and WikiHow did better with 30% and 45% respectively. When scaling up to PEGASUS we chose an effective GSR of 30%.



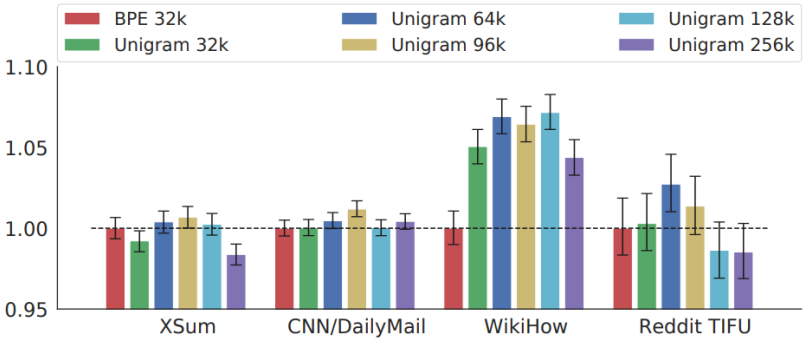
(a) Effect of pre-training objectives (30% GSR).



(b) Effect of gap sentences ratio with GSG (Ind-Orig).

EFFECT OF VOCABULARY:

We compared two tokenization methods: The byte-pair encoding algorithm (BPE) the and Sentence Piece Unigram algorithm (Unigram) proposed in Kudo (2018). We evaluated Unigram with different vocabulary sizes ranging from 32k to 256k. In these experiments, models were pre-trained for 500k steps on the C4 corpus with the Ind-Orig objective and 15% GSR. As shown in Figure 5, BPE and Unigram were comparable on news datasets while Unigram outperformed BPE on non-news datasets, especially WikiHow. On XSum and CNN/DailyMail, Unigram 96k achieved the highest ROUGE scores. On WikiHow and Reddit TIFU, the best configurations were Unigram 128k and 64k respectively. Therefore, we used the overall best vocabulary option Uni gram 96k in PEGASUS.

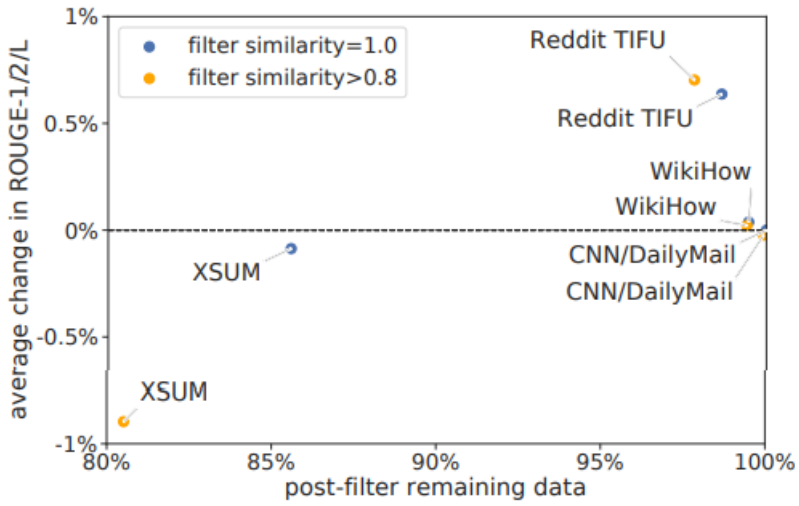


Effect of vocabulary with PEGASUS

Findings:

In real-world practice, it is often difficult to collect a large number of supervised examples to train or fine-tune a summarization model. To simulate the low-resource summarization setting, we picked the first 10k (k = 1, 2, 3, 4) training examples from each dataset to fine-tune PEGASUSLARGE (HugeNews). We fine-tuned the models up to 2000 steps with batch size 256, a learning rate of 0.0005, and picked the checkpoint with the best validation performance.

The pre-training corpora are a large collection of documents from the Internet and potentially overlap with the downstream test sets. In this section, we measured the extent of overlap between the pre-training corpus and downstream datasets. To measure the overlap, we calculated similarities between all pairs of downstream test set targets and pre-training documents. We use the ROUGE-2 recall as a similarity measure (common 2-grams / test set targets 2-grams). It is not necessarily an exact match even if the similarity score is 1.0. We filtered all test set examples that have similarities to any pre-training example above a threshold and recalculated the ROUGE scores on the remaining test set. In Figure 7, we conducted this study on the pre-training corpus C4 and test set of XSum, CNN/Dailymail, Reddit TIFU, and WikiHow, with a similarity threshold of 1.0 and 0.8. Results show that only XSum has a significant amount of overlap 15% to 20%, and filtering those examples does not change ROUGE scores by more than 1%. We also manually examined those overlapped examples with similarity of 1.0 and found that the models produce very different summaries compared to the human-written ones, suggesting that there was no clear memorization



Percentage of overlap between C4 and downstream test sets.

Additional Improvements:

1. The model was pre-trained on the mixture of C4 and HugeNews weighted by their number of examples.
2. The model dynamically chose gap sentences ratio uniformly between 15%-45%.
3. Importance sentences were stochastically sampled with 20% uniform noise on their scores.
4. The model was pre-trained for 1.5M steps instead of 500k steps, as we observed slower convergence of pre-training perplexity.
5. The SentencePiece tokenizer was updated to encode the newline character.

Conclusion:

In this work, we proposed PEGASUS, a sequence-to-sequence model with gap-sentences generation as a pre-training objective tailored for abstractive text summarization. We studied several gap-sentence selection methods and identified principle sentence selection as the optimal strategy. We demonstrated the effects of the pre-training corpora, gap-sentences ratios, and vocabulary sizes and scaled up the best configuration to achieve state-of-the-art results on all 12 diverse downstream datasets considered. We also showed that our model was able to adapt to unseen summarization datasets very quickly, achieving strong results in as little as 1000 examples. We finally showed our model summaries achieved human performance on multiple datasets using human evaluation.