THE UNIVERSITY OF TEXAS AT DALLAS

TEXT SUMMARIZATION

PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

MOTIVATION

Time-saving: Text summarization condenses text, enabling readers to quickly understand main ideas without reading the entire document.

Information overload: Text summarization filters important information from large data.

Accessibility: Text summarization offers an alternative way for those with reading difficulties or visual impairments to access information.

Computation: These transformer based summarization models are huge. Can they be made smaller?

Impracticality: Can performance of these models be replicated with limited resources?

Text Summarization - PEGASUS

PEGASUS pre-trains a neural network model on a large textual corpus to produce high-quality abstractive summaries for diverse inputs.

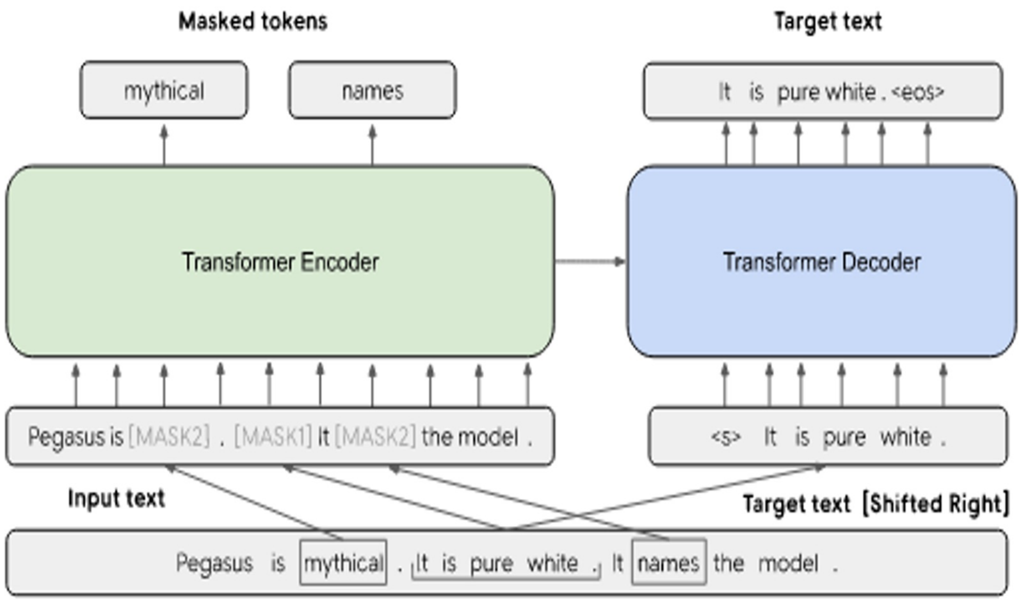
It utilizes extract gap sentences to generate high-quality abstractive summaries for a wide range of input texts.

Involves masking out sentences in the source text and predicting the missing sentence using the pre-trained model. The missing sentence is then used to generate a summary of the source text.

It involves training a large neural network model on a massive corpus of text data, enabling it to learn to summarize text by predicting missing sentences.

PEGASUS has shown significant improvements in abstractive summarization tasks, making it a promising approach for natural language processing applications.

Architecture of PEGASUS


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 .

PEGASUS' fundamental architecture is a Transformer encoder-decoder.

This example uses GSG(Gap Sentences Generation) and MLM(Masked Language Model) as pre-training objectives simultaneously.

One sentence is masked[MASK1] and is used as a text generating text.

The other two sentences are included in the input. Some tokens are randomly masked[MASK2] by MLM

Pegasus$_{\text{Base}}$ has 223M parameters. ~1.09 GB on disk.

Pegasus$_{\text{large}}$ has 568M parameters ~2.2 GB on disk.

DATASET

The CNN/DailyMail Dataset is an English-language corpus that comprises slightly over 300,000 distinctive news articles, written by journalists at CNN and the Daily Mail.

The dataset CNN/Daily Mail (See et al. (2017),Hermann et al., 2015) is available on Kaggle.

The data fields includes id - Id is a string that contains the SHA1 hash of the story's retrieval URL in hexadecimal format, article - A string of news article and highlights - A string that contains highlights written by article author.

DATA	SIZE
Train	287,113
Validation	13,368
Test	11,490

Structure Of Data

Id: 001097a19e2c96de11276b3cce11566ccfed0030

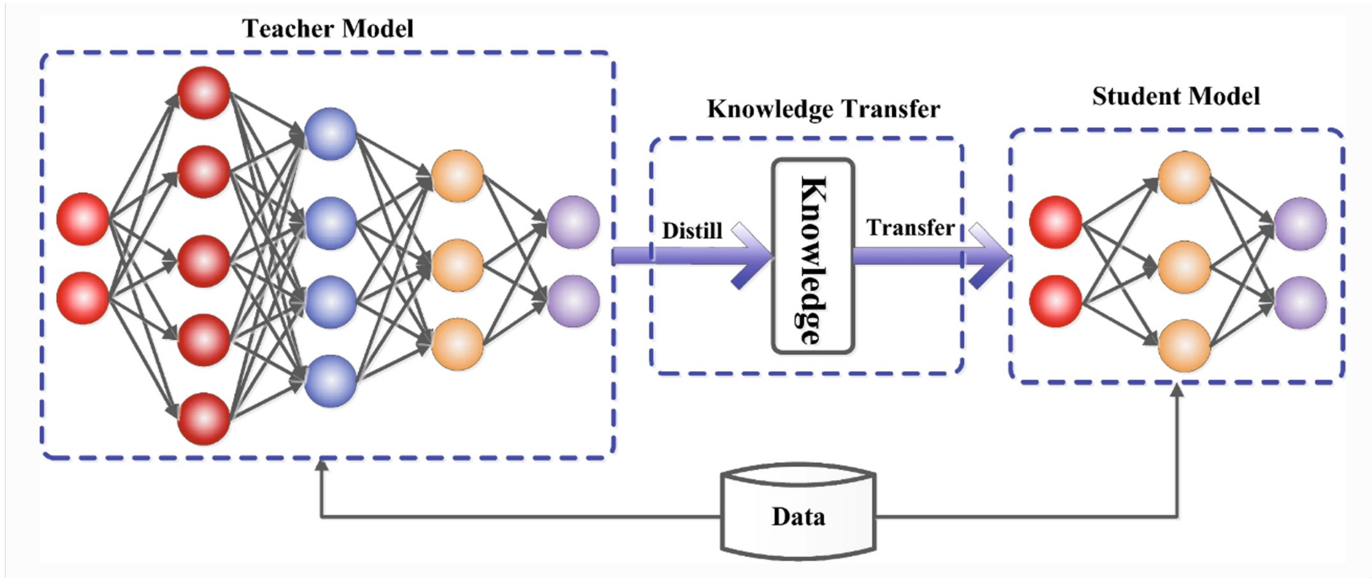
Text: For most people, it has become a travel essential. Taking your smartphone or tablet away on holiday keep you in touch with what's going on back home, as well as offering a chance to monitor 'work emails.' But a 'digital detox' revolution is taking place - a chance to embrace the holiday free from modern technology and reminders of home life. The Red Mountain Resort, in Utah, US, is an adventure spa next to Snow Canyon State Park and offers a real 'disconnected' break . Digital Detox Holidays offer the chance to leave your smartphone at home and enjoy all the luxury; pictured is Lake Placid Lodge, in the Adirondacks, US . The temptation to scour work emails on holiday has led to more and more people looking for a digital detox . In an age where itâ€™s becoming increasingly difficult to unplug, a third of Brits say they regret spending too much time on their mobile device while theyâ€™re on holiday. Half of all Brits polled admit to checking work e-mails while away and four in 10 say having access to social media is 'very important' to them when theyâ€™re abroad. One website showcasing the spots around the world free of Wi-Fi and phone reception, www.digitaldetoxholidays.com have reported a five-fold increase in customers in six months, report The Independent. Their website slogan reads: 'Since you became increasingly addicted to your devices, we have been selecting hotels that are offering detox holidays to help you de-stress.' This spot in Essex, the 'Lifehouse Spa, has a strict tech-free policy in their grounds to enable you to be at peace with the world . Recognized as 'one of the worldâ€™s nine amazing yoga retreat destinations,' Via Yoga in Mexico is the escape youâ€™ve been waiting for . The Teton Lodge at Jackson Hole, US is the perfect accommodation for the people who like winter sports and visiting nature parks - you won't even miss your smartphone . From remote beach huts, to garden lodges and mountain lodges, the company aim to find the perfect holiday where the smartphone is reduced to useless. Locations are marketed in the US, the Caribbean, and even a 'Lifehouse Spa' in Thorpe-le-Soken, Essex. Kimpton Monaco residence in Chicago, US Offers a 'black-out' option, with guests surrendering all devices upon check-in . A unique luxury ranch nestled in British Columbiaâ€™s picturesque Cariboo region, the Echo Valley Ranch & Spa, Canada offers ultimate serenity . Alison Couper, of Hotels.com, said: â€˜Going away on holiday should be a time to take stock and unwind, whether you're lying on a beach in the Seychelles or snowboarding down a mountain in Canada. â€˜While smartphones have their plus points while on leave from work, using them to check the weather or view maps, it seems travellers would benefit from switching off their e-mails to disconnect, restoring a little more of the all-important work/life balance.â€™

Highlights: "Half of Brits admit to checking work e-mails while on holiday, while a third regret spending so much time on them. Rural getaways are becoming more popular in 'digital detox' revolution, many with no signal and no Wi-Fi . Offers a chance to leave smartphones and tablets firmly switched off and enjoy the sights and scenery ."

Methodology Knowledge Distillation

Knowledge distillation is a method for compressing a larger, more complex teacher model into a smaller student model by training the student model to reproduce the output probabilities (logits) of the teacher model.

This results in a more efficient and compact student model that can still achieve high performance, and is widely used in various fields including natural language processing and computer vision.



Methodology Shrink and Fine-Tune

The Shrink and Fine-Tune (SFT) is a basic method for model compression.

The SFT involves shrinking a pre-trained teacher model to a smaller size and re-fine-tuning the new student model.

To create a student model, an arbitrary number of full decoder layers from the teacher are copied.

The decoder is the most useful part to compress, while distilling the encoder does not significantly influence the inference time.

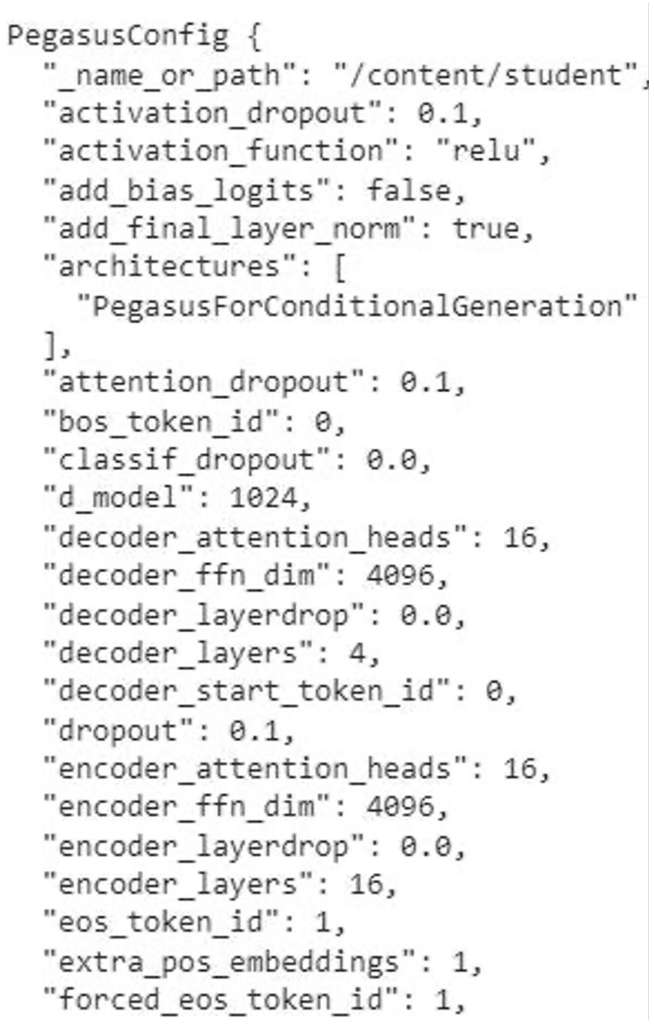
The student model is trained on the same dataset the teacher was fine-tuned on, which consists of pairs of source documents and target summaries.

Implementation

Our basic method, SFT, involves shrinking the teacher model to student size and re-fine-tuning the student model.

Our decoder model consist of 0,5,10,15 decoder layers from the 16 decoder layer teacher, and copied the full encoder 16 layers.

After initialization, the student model continues to fine-tune on the summarization dataset, with the objective of minimizing LData.



```
PegasusConfig {
  "_name_or_path": "/content/student",
  "activation_dropout": 0.1,
  "activation_function": "relu",
  "add_bias_logits": false,
  "add_final_layer_norm": true,
  "architectures": [
    "PegasusForConditionalGeneration"
  ],
  "attention_dropout": 0.1,
  "bos_token_id": 0,
  "classif_dropout": 0.0,
  "d_model": 1024,
  "decoder_attention_heads": 16,
  "decoder_ffn_dim": 4096,
  "decoder_layerdrop": 0.0,
  "decoder_layers": 4,
  "decoder_start_token_id": 0,
  "dropout": 0.1,
  "encoder_attention_heads": 16,
  "encoder_ffn_dim": 4096,
  "encoder_layerdrop": 0.0,
  "encoder_layers": 16,
  "eos_token_id": 1,
  "extra_pos_embeddings": 1,
  "forced_eos_token_id": 1,
```

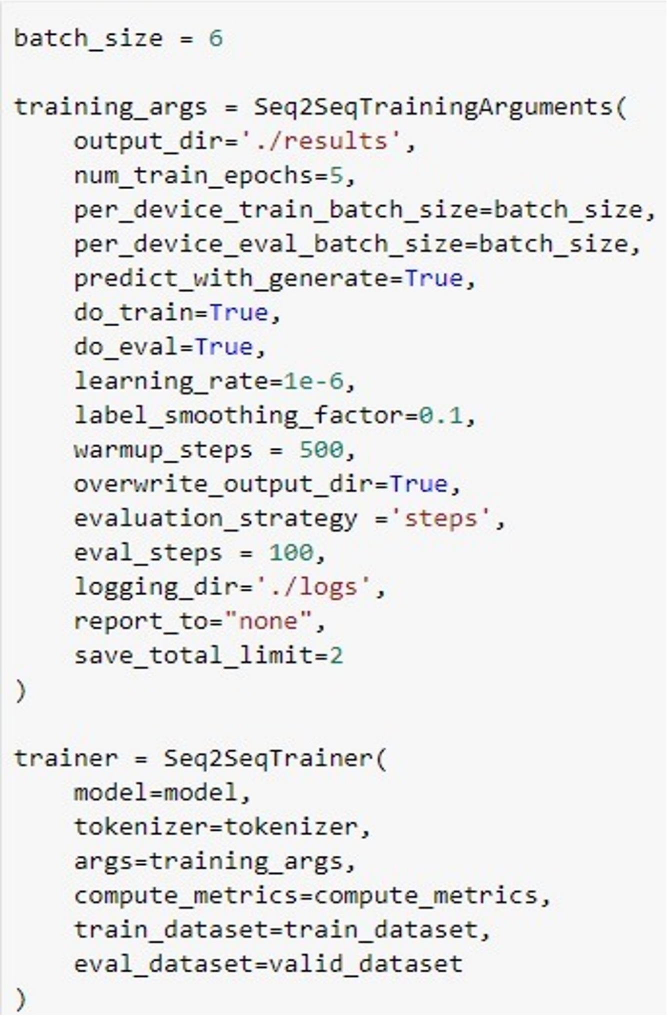
Implementation

Number of training epochs - 5

Learning rate - 1e-6

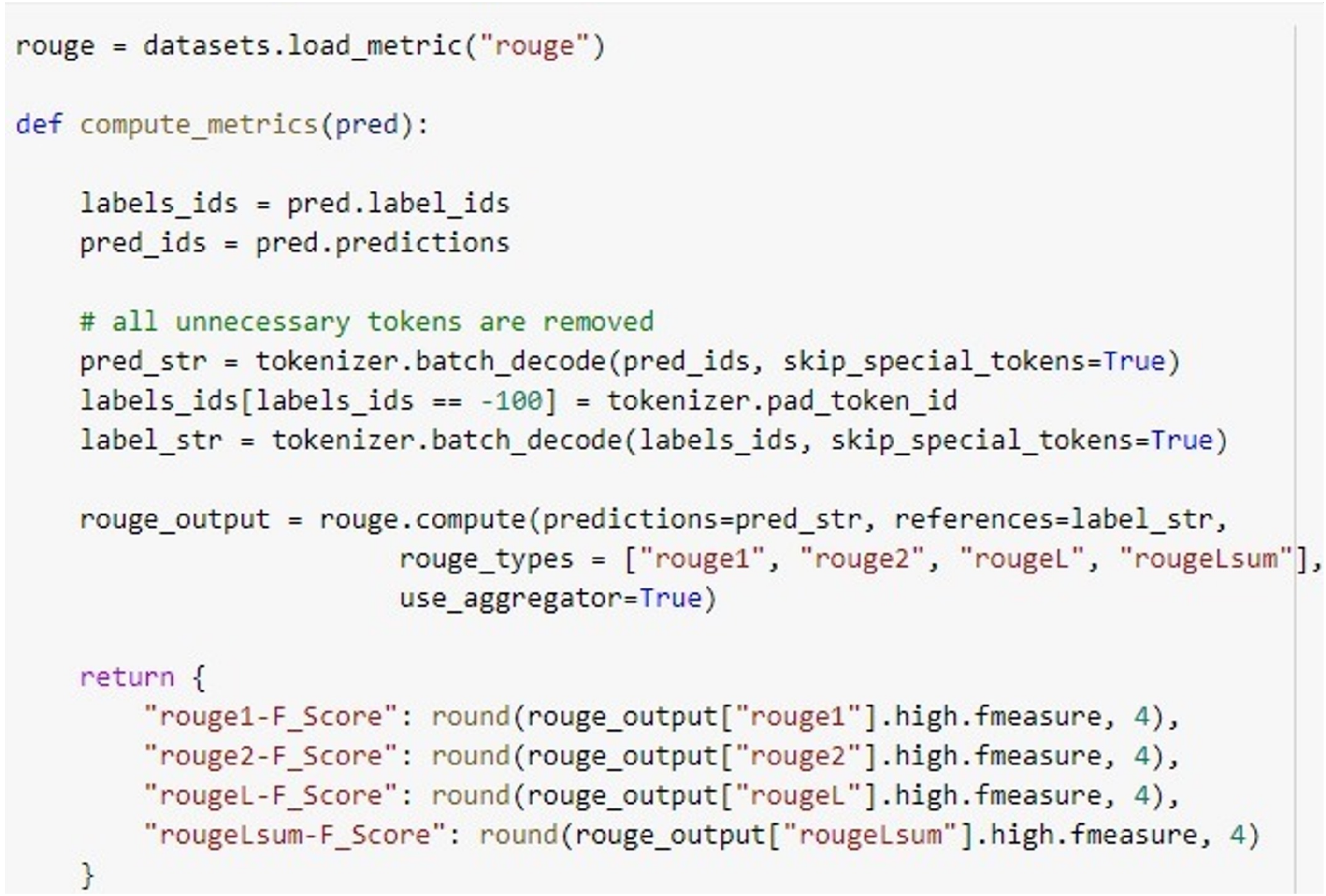
Batch size - 6

Eval Steps - 100



```
batch_size = 6

training_args = Seq2SeqTrainingArguments(
    output_dir='./results',
    num_train_epochs=5,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    predict_with_generate=True,
    do_train=True,
    do_eval=True,
    learning_rate=1e-6,
    label_smoothing_factor=0.1,
    warmup_steps = 500,
    overwrite_output_dir=True,
    evaluation_strategy ='steps',
    eval_steps = 100,
    logging_dir='./logs',
    report_to="none",
    save_total_limit=2
)

trainer = Seq2SeqTrainer(
    model=model,
    tokenizer=tokenizer,
    args=training_args,
    compute_metrics=compute_metrics,
    train_dataset=train_dataset,
    eval_dataset=valid_dataset
)
```

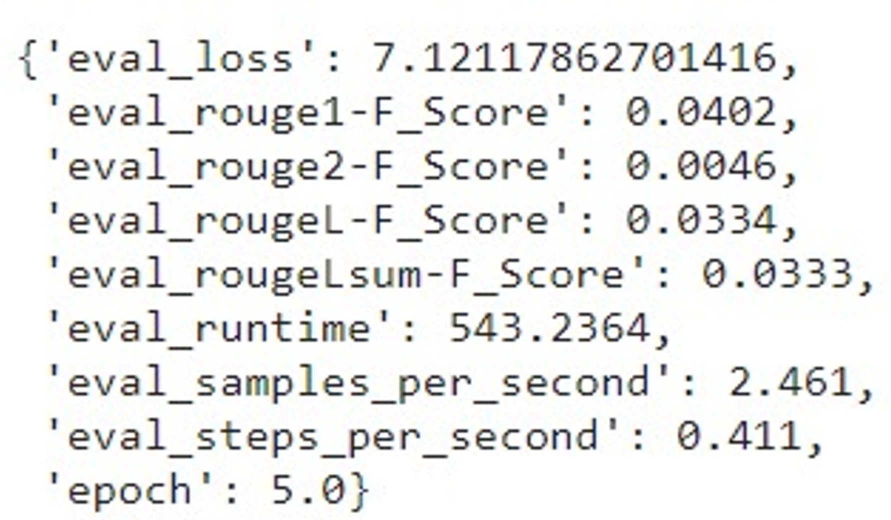
Implementation



```
rouge = datasets.load_metric("rouge")

def compute_metrics(pred):

    labels_ids = pred.label_ids
    pred_ids = pred.predictions

    # all unnecessary tokens are removed
    pred_str = tokenizer.batch_decode(pred_ids, skip_special_tokens=True)
    labels_ids[labels_ids == -100] = tokenizer.pad_token_id
    label_str = tokenizer.batch_decode(labels_ids, skip_special_tokens=True)

    rouge_output = rouge.compute(predictions=pred_str, references=label_str,
                        rouge_types = ["rouge1", "rouge2", "rougeL", "rougeLsum"],
                        use_aggregator=True)

    return {
        "rouge1-F_Score": round(rouge_output["rouge1"].high.fmeasure, 4),
        "rouge2-F_Score": round(rouge_output["rouge2"].high.fmeasure, 4),
        "rougeL-F_Score": round(rouge_output["rougeL"].high.fmeasure, 4),
        "rougeLsum-F_Score": round(rouge_output["rougeLsum"].high.fmeasure, 4)
    }
```

Experiments and Results

Test Results:



```
{'eval_loss': 7.12117862701416,
 'eval_rouge1-F_Score': 0.0402,
 'eval_rouge2-F_Score': 0.0046,
 'eval_rougeL-F_Score': 0.0334,
 'eval_rougeLsum-F_Score': 0.0333,
 'eval_runtime': 543.2364,
 'eval_samples_per_second': 2.461,
 'eval_steps_per_second': 0.411,
 'epoch': 5.0}
```

References

[1] https://link.springer.com/article/10.1007/s11263-021-01453-z

[2] https://arxiv.org/pdf/2010.13002.pdf

[3] https://arxiv.org/pdf/1912.08777.pdf

Thank You