Paper link: <https://arxiv.org/pdf/1910.13461.pdf>

### Overview

One model we chose is BART which is a powerful model that was trained with document corruption (token masking, token deletion, etc.) and fine tuned on a variety of tasks including text summarization. In the paper, BART is evaluated on the CNN/Daily Mail dataset with the ROUGE score. This metric looks at the overlap of n grams between the model and the reference output. Specifically ROUGE-1 (R1) looks at unigrams, ROUGE-2 (R2) looks at bigrams, and ROUGE-L (RL) looks at longest co-occurring n-gram sequences.

### Model Architecture

The BART model consists of a sequence to sequence transformer architecture, and consists of encoder and decoder modules involving self-attention mechanisms and fully connected feed forward networks in each of 6 layers.

### Method of Replication

results = {'rouge1': np.zeros(3), 'rouge2': np.zeros(3), 'rougeL': np.zeros(3)}

for data in tqdm(test\_data):

article, ref = data

inputs = tokenizer(article, return\_tensors="pt", max\_length=1024).to(device)

summary\_ids = model.generate(inputs["input\_ids"], num\_beams=2, min\_length=0, max\_length=100)

res = tokenizer.batch\_decode(summary\_ids, skip\_special\_tokens=True, clean\_up\_tokenization\_spaces=False)[0]

scorer = rouge\_scorer.RougeScorer(['rouge1', 'rougeL', 'rouge2'], use\_stemmer=True)

scores = scorer.score(res, ref)

for key in scores.keys():

results[key] += np.array(scores[key])

# average over test data

for key in results.keys():

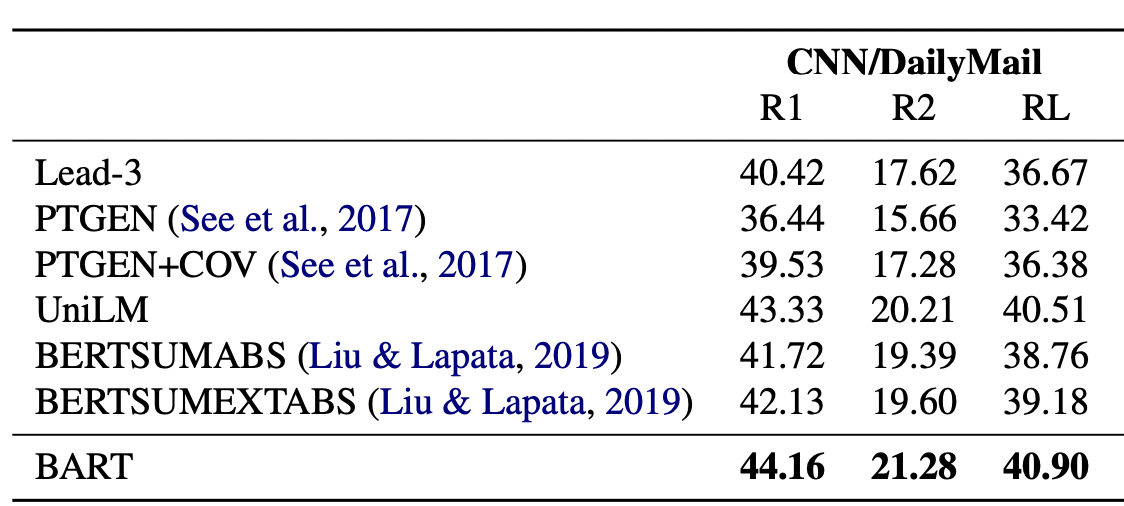
results[key] /= len(test\_data)



We used the pretrained BART model from hugging face (<https://huggingface.co/docs/transformers/model_doc/bart> ) and evaluated it on the CNN/Daily Mail dataset. Above you can see our code for generating a summary given a test article and then calculating the three ROUGE metrics on it.

### Results

The paper reports the following accuracies on the CNN/Daily Mail data:



Our results using the pretrained model on the CNN/DM dataset can be seen below and they seem to be generally close to those reported.

average R1: 0.44161249

average R2: 0.21270519

average RL: 0.35731044

