Make Lead Bias in Your Favor: A Simple and Effective Method for News Summarization

Chenguang Zhu[1] , **Ziyi Yang**[2] , **Robert Gmyr**[1] , **Michael Zeng**[1] and **Xuedong Huang**[1]
[1]Microsoft Speech and Dialogue Research Group
[2]Stanford University
{chezhu,rogmyr,nzeng,xdh}@microsoft.com, zy99@stanford.edu

Abstract

Lead bias is a common phenomenon in news summarization, where early parts of an article often contain the most salient information. While many algorithms exploit this fact in summary generation, it has a detrimental effect on teaching the model to discriminate and extract important information. We propose that the lead bias can be leveraged in a simple and effective way in our favor to pretrain abstractive news summarization models on large-scale unlabeled corpus: predicting the leading sentences using the rest of an article. Via careful data cleaning and filtering, our transformer-based pretrained model without any finetuning achieves remarkable results over various news summarization tasks. With further finetuning, our model outperforms many competitive baseline models. Human evaluations further show the effectiveness of our method.

1 Introduction

The goal of text summarization is to condense a piece of text into a shorter version that contains the salient information. Due to the prevalence of news articles and the need to provide succinct summaries for readers, a majority of existing datasets for summarization come from the news domain [Hermann *et al.*, 2015; Sandhaus, 2008; Narayan *et al.*, 2018]. However, according to journalistic conventions, the most important information in a news report usually appears near the beginning of the article [Kedzie *et al.*, 2018]. While it facilitates faster and easier understanding of the news for readers, this lead bias causes undesirable consequences for summarization models. The output of these models is inevitably affected by the positional information of sentences. Furthermore, the simple baseline of using the top few sentences as summary can achieve a stronger performance than many sophisticated models [See *et al.*, 2017]. It can take a lot of effort for models to overcome the lead bias [Kedzie *et al.*, 2018].

Additionally, most existing summarization models are fully supervised and require time and labor-intensive annotations to feed their insatiable appetite for labeled data. For example, the New York Times Annotated Corpus [Sandhaus, 2008] contains 1.8 million news articles, with 650,000 summaries written by library scientists. Therefore, some recent work [Gusev, 2019] explores the effect of domain transfer to utilize datasets other than the target one. But this method may be affected by the domain drift problem and still suffers from the lack of labelled data.

The recent promising trend of pretraining models [Devlin *et al.*, 2018; Radford *et al.*, 2018] proves that a large quantity of data can be used to boost NLP models' performance. Therefore, we put forward a novel method to leverage the lead bias of news articles in our favor to conduct large-scale pretraining of summarization models. The idea is to leverage the top few sentences of a news article as the target summary and use the rest as the content. The goal of our pretrained model is to generate an abstractive summary given the content. Coupled with careful data filtering and cleaning, the lead bias can provide a delegate summary of sufficiently good quality, and it immediately renders the large quantity of unlabeled news articles corpus available for training news summarization models.

We employ this pretraining idea on a three-year collection of online news articles. We conduct thorough data cleaning and filtering. For example, to maintain a quality assurance bar for using leading sentences as the summary, we compute the ratio of overlapping non-stopping words between the top 3 sentences and the rest of the article. As a higher ratio implies a closer semantic connection, we only keep articles for which this ratio is higher than a threshold.

We end up with 21.4M articles based on which we pretrain a transformer-based encoder-decoder summarization model. We conduct thorough evaluation of our models on five benchmark news summarization datasets. Our pretrained model achieves a remarkable performance on various target datasets without *any* finetuning. This shows the effectiveness of leveraging the lead bias to pretrain on large-scale news data. We further finetune the model on target datasets and achieve better results than a number of strong baseline models. For example, the pretrained model without finetuning obtains state-of-the-art results among unsupervised models on CNN/DailyMail. The finetuned model obtains 3.2% higher ROUGE-1, 1.6% higher ROUGE-2 and 2.1% higher ROUGE-L scores than the best baseline model on XSum dataset [Narayan *et al.*, 2018]. Human evaluation results also show that our models outperform existing baselines like pointer-generator network.

The rest of paper is organized as follows. We introduce related work in news summarization and pretraining in Section 2. We describe the details of pretraining using lead bias in Section 3. We introduce the transformer-based summarization model in Section 4. We show the experimental results in Section 5 and conclude the paper in Section 6.

2 Related work

2.1 Document Summarization

End-to-end abstractive text summarization has been intensively studied in recent literature. To generate summary tokens, most architectures take the encoder-decoder approach [Sutskever *et al.*, 2014]. [Rush *et al.*, 2015a] first introduces an attention-based seq2seq model to the abstractive sentence summarization task. However, its output summary degenerates as document length increases, and out-of-vocabulary (OOV) words cannot be efficiently handled. To tackle these challenges, [See *et al.*, 2017] proposes a pointer-generator network that can both produce words from the vocabulary via a generator and copy words from the source article via a pointer. [Paulus *et al.*, 2017] utilizes reinforcement learning to improve the result. [Gehrmann *et al.*, 2018] uses a content selector to over-determine phrases in source documents that helps constrain the model to likely phrases. [You *et al.*, 2019] adds Gaussian focal bias and a salience-selection network to the transformer encoder-decoder structure [Vaswani *et al.*, 2017] for abstractive summarization. [Grenander *et al.*, 2019] randomly reshuffles the sentences in news articles to reduce the effect of lead bias in extractive summarization.

2.2 Pretraining

In recent years, pretraining language models have proved to be quite helpful in NLP tasks. The state-of-the-art pretrained models include ELMo [Peters *et al.*, 2018], GPT [Radford *et al.*, 2018], BERT [Devlin *et al.*, 2018] and UniLM [Dong *et al.*, 2019]. Built upon large-scale corpora, these pretrained models learn effective representations for various semantic structures and linguistic relationships. As a result, pretrained models have been widely used with considerable success in applications such as question answering [Zhu *et al.*, 2018], sentiment analysis [Peters *et al.*, 2018] and passage reranking [Nogueira and Cho, 2019]. Furthermore, UniLM [Dong *et al.*, 2019] leverages its sequence-to-sequence capability for abstractive summarization; the BERT model has been employed as an encoder in BERTSUM [Liu and Lapata, 2019] for extractive/abstractive summarization.

Compared to our work, UniLM [Dong *et al.*, 2019] is a general language model framework and does not take advantage of the special semantic structure of news articles. Similarly, BERTSUM [Liu and Lapata, 2019] directly copies the pretrained BERT structure into its encoder and finetunes on labelled data instead of pretraining with the large quantity of unlabeled news corpus available. Recently, PEGASUS [Zhang *et al.*, 2019] leverages a similar idea of summarization pretraining, but they require finetuning with data from target domains, whereas our model has a remarkable performance without any finetuning.

3 Pretraining with Leading Sentences

News articles usually follow the convention of placing the most important information early in the content, forming an inverted pyramid structure. This lead bias has been discovered in a number of studies [Kedzie *et al.*, 2018; Grenander *et al.*, 2019]. One of the consequences is that the lead baseline, which simply takes the top few sentences as the summary, can achieve a rather strong performance in news summarization. For instance, in the CNN/Daily Mail dataset [Hermann *et al.*, 2015], using the top three sentences as summaries can get a higher ROUGE score than many deep learning based models. This positional bias brings lots of difficulty for models to extract salient information from the article and generate high-quality summaries. For instance, [Grenander *et al.*, 2019] discovers that most models' performances drop significantly when a random sentence is inserted in the leading position, or when the sentences in a news article are shuffled.

On the other hand, news summarization, just like many other supervised learning tasks, suffers from the scarcity of labelled training data. Abstractive summarization is especially data-hungry since the efficacy of models depends on high-quality handcrafted summaries.

We propose that the lead bias in news articles can be leveraged in our favor to train an abstractive summarization model without human labels. Given a news article, we treat the top three sentences, denoted by Lead-3, as the target summary, and use the rest of the article as news content. The goal of the summarization model is to produce Lead-3 using the following content, as illustrated in Figure 1.

The benefit of this approach is that the model can leverage the large number of unlabeled news articles for pretraining. In the experiment, we find that the pretrained model alone can have a strong performance on various news summarization datasets, without any further training. We also finetune the pretrained model on downstream datasets with labelled summaries. The model can quickly adapt to the target domain and further increase its performance.

It is worth noting that this idea of utilizing structural bias for large-scale summarization pretraining is not limited to specific types of models, and it can be applied to other types of text as well: academic papers with abstracts, novels with editor's notes, books with tables of contents.

However, one should carefully examine and clean the source data to take advantage of lead bias, as the top three sentences may not always form a good summary. We provide more details in the experiments about the data filtering and cleaning mechanism we apply.

4 Model

In this section, we introduce our abstractive summarization model, which has a transformer-based encoder-decoder structure. We first formulate the supervised summarization problem and then present the network architecture.

4.1 Problem formulation

We formalize the problem of supervised abstractive summarization as follows. The input consists of a pairs of articles and summaries: $\{(X_1, Y_1), (X_2, Y_2), ..., (X_a, Y_a)\}$. Each article and summary are tokenized: $X_i = (x_1, ..., x_{L_i})$ and $Y_i = (y_1, ..., y_{N_i})$. In abstractive summarization, the summary tokens need not be from the article. For simplicity, we will drop the data index subscript. The goal of the system is to generate summary $Y = (y_1, ..., y_m)$ given the transcript $X = \{x_1, ..., x_n\}$.

4.2 Network Structure

We utilize a transformer-based encoder-decoder structure that maximizes the conditional probability of the summary: $P(Y|X, \theta)$, where θ represents the parameters.

Encoder

The encoder maps each token into a fixed-length vector using a trainable dictionary $\mathcal{D}$ randomly initialized using a normal distribution with zero mean and a standard deviation of 0.02. Each transformer block conducts multi-head self-attention. And we use sinusoidal positional embedding in order to process arbitrarily long input. In the end, the output of the encoder is a set of contextualized vectors:

$$\text{Encoder-Transformer}(\{x_1, ..., x_n\}) = \{u_1^E, ..., u_n^E\}$$

Decoder

The decoder is a transformer that generates the summary tokens one at a time, based on the input and previously generated summary tokens. Each token is projected onto a vector using the same dictionary $\mathcal{D}$ as the encoder.

The decoder transformer block includes an additional cross-attention layer to fuse in information from the encoder. The output of the decoder transformer is denoted as:

$$\text{Decoder-Transformer}(\{w_1, ..., w_{k-1}\}) = \{u_1^D, ..., u_{k-1}^D\} \tag{1}$$

To predict the next token w_k, we reuse the weights of dictionary $\mathcal{D}$ as the final linear layer to decode u_{k-1}^D into a probability distribution over the vocabulary: $P(w_k|w_{<k}, u_{1:m}^E) = \text{softmax}(\mathcal{D}u_{k-1}^D)$.

Training. During training, we seek to minimize the cross-entropy loss:

$$L(\theta) = -\frac{1}{m}\sum_{k=1}^{m} \log P(y_k|y_{<k}, X) \tag{2}$$

We use teacher-forcing in decoder training, i.e. the decoder takes ground-truth summary tokens as input. The model has 10 layers of 8-headed transformer blocks in both its encoder and decoder, with 154.4M parameters.

Inference. During inference, we employ beam search to select the best candidate. The search starts with the special token $\langle$BEGIN$\rangle$. We ignore any candidate word which results in duplicate trigrams. We select the summary with the highest average log-likelihood per token.

5 Experiments

5.1 Datasets

We evaluate our model on five benchmark summarization datasets: the New York Times Annotated Corpus (NYT) [Sandhaus, 2008], XSum [Narayan *et al.*, 2018] and the CNN/DailyMail dataset [Hermann *et al.*, 2015]. These datasets contain 104K, 227K, 312K news articles and human-edited summaries respectively, covering different topics and various summarization styles. For NYT dataset, we use the same train/val/test split and filtering methods following [Durrett *et al.*, 2016].

5.2 Implementation Details

We use SentencePiece [Kudo and Richardson, 2018] for tokenization, which segments any sentence into subwords. We train the SentencePiece model on pretrained data to generate a vocabulary of size 32K and of dimension 720. The vocabulary stays fixed during pretraining and finetuning.

Pretraining. We collect three years of online news articles from June 2016 to June 2019. We filter out articles overlapping with the evaluation data on media domain and time range. We then conduct several data cleaning strategies.

First, many news articles begin with reporter names, media agencies, dates or other contents irrelevant to the content, e.g. "New York (CNN) –", "Jones Smith, May 10th, 2018:". We therefore apply simple regular expressions to remove these prefixes.

Second, to ensure that the summary is concise and the article contains enough salient information, we only keep articles with 10-150 words in the top three sentences and 150-1200 words in the rest, and that contain at least 6 sentences in total. In this way, we filter out i) articles with excessively long content to reduce memory consumption; ii) very short leading sentences with little information which are unlikely to be a good summary. To encourage the model to generate abstrative summaries, we also remove articles where any of the top

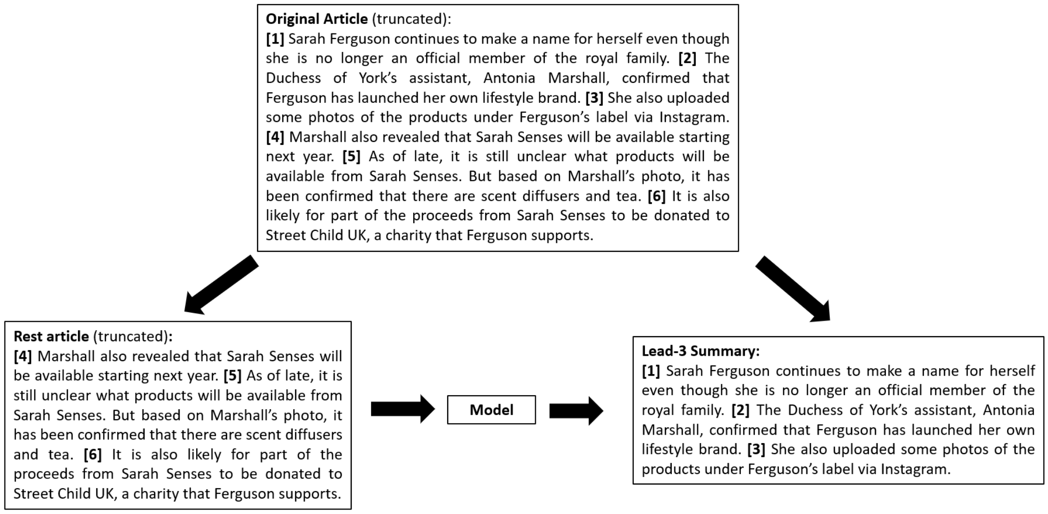

Figure 1: Using Lead-3 summary as target in pretraining.

three sentences is exactly repeated in the rest of the article.

Third, we try to remove articles whose top three sentences may not form a relevant summary. For this purpose, we utilize a simple metric: overlapping words. We compute the portion of non-stopping words in the top three sentences that are also in the rest of an article. A higher portion implies that the summary is representative and has a higher chance of being inferred by the model using the rest of the article. To verify, we compute the overlapping ratio of non-stopping words between human-edited summary and the article in CNN/DailyMail dataset, which has a median value of 0.87. Therefore, in pretraining, we keep articles with an overlapping word ratio higher than 0.65.

These filters rule out around 95% of the raw data and we end up with 21.4M news articles, 12,000 of which are randomly sampled for validation.

We pretrain the model for 10 epochs and evaluate its performance on the validation set at the end of each epoch. The model with the highest ROUGE-L score is selected.

During pretraining, we use a dropout rate of 0.3 for all inputs to transformer layers. The batch size is 1,920. We use RAdam [Liu *et al.*, 2019] as the optimizer, with a learning rate of 10^{-4}. Also, due to the different numerical scales of the positional embedding and initialized sentence piece embeddings, we divide the positional embedding by 100 before feeding it into the transformer. The beam width is set to 5 during inference.

Finetuning. During finetuning, we keep the optimizer, learning rate and dropout rate unchanged as in pretraining. The batch size is 32 for all datasets. We pick the model with the highest ROUGE-L score on the validation set and report its performance on the test set.

Our strategy of Pretraining with unlabeled Lead-3 summaries is called **PL**. We denote the pretrained model with finetuning on target datasets as **PL-FT**. The model with only pretraining and no finetuning is denoted as **PL-NoFT**, which is the same model for all datasets.

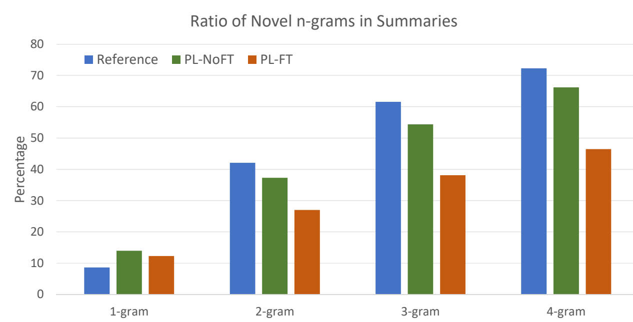

Figure 2: Ratio of novel n-grams in summaries from reference, PL-NoFT and PL-FT models in NYT test set.

5.3 Baseline

To compare with our model, we select a number of strong summarization models as baseline systems. LEAD-X uses the top X sentences as a summary [Liu and Lapata, 2019]. The value of X is 3 for NYT and CNN/DailyMail[1] and 1 for XSum to accommodate the nature of summary length. PTGEN [See *et al.*, 2017] is the pointer-generator network. DRM [Paulus *et al.*, 2017] leverages deep reinforcement learning for summarization. TCONVS2S [Narayan *et al.*, 2018] is based on convolutional neural networks. BOTTOMUP [Gehrmann *et al.*, 2018] uses a bottom-up approach to generate summarization. ABS [Rush *et al.*, 2015b] uses neural attention for summary generation. DRGD [Li *et al.*, 2017] is based on a deep

[1]The ROUGE scores here on CNN/Daily Mail are higher than those reported in the original paper, because we extract 3 sentences in Daily Mail rather than 4.

Model	R1	R2	RL
Lead-3	39.58	20.11	35.78
PTGen	42.47	25.61	—
PTGen + Cov	43.71	26.40	—
DRM	42.94	26.02	—
PL-NoFT	35.32	17.80	31.88
PL-FT	**44.18***	**27.49***	**40.65****

Table 1: ROUGE recall scores on **NYT** test set.

Model	R1	R2	RL
Lead-1	16.30	1.60	11.95
PTGen	29.70	9.21	23.24
PTGen+Cov	28.10	8.02	21.72
TConvS2S	31.89	11.54	25.75
PL-NoFT	24.12	5.59	19.20
PL-FT	**35.06****	**13.12****	**27.86****

Table 2: ROUGE F1 results on **XSum** test set.

Model	R1	R2	RL
Lead-3	40.5	17.7	36.7
Unsupervised			
SEQ^3	17.85	3.94	19.53
GPT-2	29.34	8.27	26.58
PL-NoFT	**38.95****	**16.27****	**35.11****
Supervised			
PTGen	36.44	15.66	33.42
PTGen+Cov	39.53	17.28	36.38
DRM	39.87	15.82	36.90
BottomUp	**41.22**	**18.68**	**38.34**
PL-FT	40.41	17.81	37.19

Table 3: ROUGE F1 results on **CNN/DailyMail** test set.

recurrent generative decoder.

To compare with our pretrain-only model, we include several unsupervised abstractive baselines: SEQ^3 [Baziotis *et al.*, 2019] employs the reconstruction loss and topic loss for summarization. BottleSum [West *et al.*, 2019] leverages unsupervised extractive and self-supervised abstractive methods. GPT-2 [Radford *et al.*, 2018] is a large-scaled pretrained language model which can be directly used to generate summaries[2].

5.4 Metrics

We employ the standard ROUGE-1, ROUGE-2 and ROUGE-L metrics [Lin, 2004] to evaluate all summarization models. These three metrics respectively evaluate the accuracy on unigrams, bigrams and longest common subsequence. ROUGE metrics have been shown to highly correlate with the human judgment [Lin, 2004]. Following [Durrett *et al.*, 2016; West *et al.*, 2019], we use F-measure ROUGE on XSUM and CNN/DailyMail, and use limited-length recall-measure ROUGE on NYT. In NYT, the prediction is truncated to the length of the ground-truth summaries.

5.5 Results

The results are displayed in Table 1, Table 2 and Table 3. As shown, on both NYT and XSum dataset, PL-FT outperforms all baseline models by a large margin. For instance, PL-FT obtains 3.2% higher ROUGE-1, 1.6% higher ROUGE-2 and 2.1% higher ROUGE-L scores than the best baseline model on XSum dataset. We conduct statistical test and found that the results are all significant with p-value smaller than 0.05 (marked by *) or 0.01 (marked by **), compared with previous best scores. On CNN/DailyMail dataset, PL-FT outperforms all baseline models except BottomUp [Gehrmann *et al.*, 2018].

PL-NoFT, the pretrained model without any finetuning, also gets remarkable results. On XSum dataset, PL-NoFT is almost 8% higher than Lead-1 in ROUGE-1 and ROUGE-L. On CNN/DailyMail dataset, PL-NoFT significantly outperforms unsupervised models SEQ^3 and GPT-2, and even surpasses the supervised pointer-generator network. It's worth noting that PL-NoFT is the same model for all experiments, which proves that our pretrain strategy is effective across different news corpus.

5.6 Abstractiveness Analysis

We measure the abstractiveness of our model via the ratio of novel n-grams in summaries, i.e. the percentage of n-grams in the summary that are not present in the article. Figure 2 shows this ratio in summaries from reference and generated by PL-NoFT and PL-FT in NYT dataset. Both PL-NoFT and PL-FT yield more novel 1-grams in summary than the reference. And PL-NoFT has similar novelty ratio with the reference in other n-gram categories. Also, we observe that the novelty ratio drops after finetuning. We attribute this to the strong lead bias in the NYT dataset which affects models trained on it.

5.7 Human Evaluation

We conduct human evaluation of the generated summaries from our models and the pointer generator network with coverage. We randomly sample 100 articles from the CNN/DailyMail test set and ask 3 human labelers from Amazon Mechanical Turk to assess the quality of summaries with a score from 1 to 5 (5 means perfect quality. The labelers need to judge whether the summary can express the salient information from the article in a concise form of fluent language. The evaluation guidelines are given in Table 5. To reduce bias, we randomly shuffle summaries from different sources for each article.

As shown in Table 5, both of our models PL-NoFT and PL-FT outperform the pointer generator network (PT-

[2]We follow GPT-2's approach to add *TL;DR:* after the article for summary generation. And we use the GPT-2 small model available.

Score	Criteria
5	Summary contains all key points.
4	Summary misses one key point.
3	Summary misses two key points.
2	Summary misses all key points.
1	Summary is hardly related to the news or the language is not natural and fluent.

Table 4: Scoring criteria for human evaluation of summaries.

Model	Average Score	Standard deviation
PTGen+Cov	3.24	1.17
PL-NoFT	3.47	1.12
PL-FT	**4.09****	0.88

Table 5: Average and standard deviations of human evaluation scores for summaries on CNN/DailyMail test set. Scores range from 1 to 5 with 5 being perfect. Each summary is judged by 3 human evaluators. PL-FT's result is statistically significant compared with pointer-generator network with coverage with a p-value less than 10^{-7}.

Gen+Cov), and PL-FT's advantage over PTGen+Cov is statistically significant. This shows the effectiveness of both our pretraining and finetuning strategy. To evaluate the inter-annotator agreement, we compute the kappa statistics among the labels and the score is 0.34.

6 Conclusions

In this paper, we propose a simple and effective pretraining method for news summarization. By employing the leading sentences from a news article as its target summary, we turn the problematic lead bias for news summarization in our favor. Based on this strategy, we conduct pretraining for abstractive summarization in a large-scale news corpus. We conduct thorough empirical tests on five benchmark news summarization datasets, including both automatic and human evaluations. Results show that the same pretrained model without any finetuning can achieve state-of-the-art results among unsupervised methods over various news summarization datasets. And finetuning on target domains can further improve the model's performance. We argue that this pretraining method can be applied in more scenarios where structural bias exists.

References

[Baziotis *et al.*, 2019] Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. Seqˆ 3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. *arXiv preprint arXiv:1904.03651*, 2019.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dong *et al.*, 2019] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*, 2019.

[Durrett *et al.*, 2016] Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. Learning-based single-document summarization with compression and anaphoricity constraints. *arXiv preprint arXiv:1603.08887*, 2016.

[Gehrmann *et al.*, 2018] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*, 2018.

[Grenander *et al.*, 2019] Matt Grenander, Yue Dong, Jackie C.K. Cheung, and Annie Louis. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. *EMNLP*, 2019.

[Gusev, 2019] Ilya Gusev. Importance of copying mechanism for news headline generation. *arXiv preprint arXiv:1904.11475*, 2019.

[Hermann *et al.*, 2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, pages 1693–1701, 2015.

[Kedzie *et al.*, 2018] Chris Kedzie, Kathleen McKeown, and Hal Daume III. Content selection in deep learning models of summarization. *arXiv preprint arXiv:1810.12343*, 2018.

[Kudo and Richardson, 2018] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[Li *et al.*, 2017] Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. Deep recurrent generative decoder for abstractive text summarization. *arXiv preprint arXiv:1708.00625*, 2017.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out,*, 2004.

[Liu and Lapata, 2019] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *EMNLP*, 2019.

[Liu *et al.*, 2019] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[Narayan *et al.*, 2018] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*, 2018.

[Nogueira and Cho, 2019] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[Paulus *et al.*, 2017] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[Peters *et al.*, 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[Rush *et al.*, 2015a] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[Rush *et al.*, 2015b] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[Sandhaus, 2008] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

[See *et al.*, 2017] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems,*, pages 3104–3112, 2014.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. pages 5998–6008, 2017.

[West *et al.*, 2019] Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. Bottlesum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. *arXiv preprint arXiv:1909.07405*, 2019.

[You *et al.*, 2019] Yongjian You, Weijia Jia, Tianyi Liu, and Wenmian Yang. Improving abstractive document summarization with salient information modeling. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2132–2141, 2019.

[Zhang *et al.*, 2019] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization jingqing zhang, yao zhao, mohammad saleh, peter j. liu. *arXiv preprint arXiv:1912.08777*, 2019.

[Zhu *et al.*, 2018] Chenguang Zhu, Michael Zeng, and Xuedong Huang. Sdnet: Contextualized attention-based deep network for conversational question answering. *arXiv preprint arXiv:1812.03593*, 2018.