Efficient Two-stage Approach for Long Document Summarization

Stanford CS224N Custom

Benson Zu
Mathematical & Computational Engineering
Stanford University
zuyeyang@stanford.edu

Jialuo Yuan
Civil & Environmental Engineering
Stanford University
yjl705@stanford.edu

Fengming Tang
Mathematical & Computational Engineering
Stanford University
tfmin127@stanford.edu

Abstract

Long document summarization is essential in NLP, but state-of-the-art models like BART and BERT face limitations summarizing long documents. Current methods like Longformer Encoder Decoder (LED) can handle longer document summarization but suffers from slower processing times due to their complex attention mechanisms. To address this challenge, we propose a two-stage approach that combines sentence extraction algorithms with BART for generating abstractive summaries. Our approach leverages the efficiency of extraction algorithms to identify key sentences from the input document. BART then generates more coherent and informative abstractive summaries from these extracted sentences. Our experimental results show that our approach is four times more time-efficient than the LED baseline while processing the same amount of data, with approximately the same performance in terms of Rouge 1,2, and L F-measure. With the hope of further improving the generated summaries, we also use Generative Adversarial Network to train the model. Our proposed approach has important implications for NLP applications that require summarization of long text, such as legal documents or scientific papers.

1 Key Information

- TA mentor: Elaine Yi Ling Sui
- No sharing project, or other External collaborators and mentor.

2 Introduction

Long document summarization is an increasingly important task in the field of natural language processing (NLP) as the amount of information available in the form of lengthy documents, articles, and reports continues to grow [1]. Automatic summarization helps users access relevant information quickly and efficiently, saving time and effort. However, summarizing long documents (>1024 input tokens) poses significant challenges for state-of-the-art pre-trained models like BART(Bidirectional Auto-Regressive Transformers) [2] and BERT(Bidirectional Encoder Representations from Transformers) [3] due to their token limitations (typically around 1024 tokens).

To overcome these limitations, we propose an efficient two-stage approach for long document summarization that leverages both unsupervised and supervised techniques. The first stage extracts

key sentences from the original text and reduced the input to an extracted summary of <1100 tokens, which is then passed into BART, a state-of-the-art full-attention transformer model, to generate abstractive summaries. For the extraction stage, we study both supervised where we use longformer to generate extractive summary and unsupervised algorithms including LexRank [4] and LSA(Latent semantic analysis) [5].

Our experimental results demonstrate that the proposed two-stage approach is more time-efficient than the LED baseline. While the unsupervised extraction algorithms has a small performance trade-off in terms of F-measure (higher precision but lower recall), the supervised extraction algorithm achieves about the same performance in terms of F-measure compared with the baseline approach. This finding is in line with previous work suggesting that hybrid extractive-abstractive methods can achieve a good balance between efficiency and quality [6, 7].

3 Related Work

Current methods such as LED (Longformer Encoder Decoder) [8] have been developed to handle long document summarization, but they often suffer from slower processing times due to their more complex attention mechanisms, such as the sliding window self-attention in Longformer [8]. Our two-stage approach aims to offer a more time-efficient alternative while maintaining competitive summarization quality.

Our idea is inspired by the success of hybrid extractive-abstractive methods in the summarization domain [9, 10]. One of the most significant advantages of unsupervised extraction algorithms is that they do not require labeled data for training. This eliminates the time-consuming and expensive process of manual annotation, making the approach more convenient and cost-effective, and this makes them highly scalable, allowing for quick adaptation to new domains or topics.

We aim to capitalize on the efficiency of extraction algorithm in extracting key sentences and the powerful abstraction capabilities of BART to create coherent and informative summaries. This combination allows us to address the limitations of existing models and provide a practical solution for summarizing long documents.

4 Approach

4.1 Baseline Model

We load the pretrained LED model "led-base-16384" from Hugging Face [11] as a baseline for text summarization tasks due to its ability to handle long documents. LED extends the Transformer architecture by incorporating Longformer's self-attention mechanism. It uses a sliding window attention mechanism called the "local attention"[8] that allows it to process longer inputs efficiently by focusing on a fixed-size window of neighboring tokens. The traditional Encoder or Decoder pre-trained model like BART[2] uses standard self-attention that scales quadratically with input length. In theory, Longformer's attention mechanism should be more efficient for processing long input sequences, but it could still be slower due to other factors depending on the datasets.

4.2 Extractive-Abstractive two stage system

Figure 1 shows the architecture of our two stage summarization system. In the first stage, unsupervised extraction algorithms such as LexRank and LSA and supervised algorithm using Longformer encoder are used to dynamically select the top k featured sentences, limiting the output to 1100 tokens per sample. In the second stage, the extracted text is used to fine-tune bart-large-cnn, with the ground-truth abstract summary serving as the target output. Finally, the trained bart-large-cnn is evaluated on the test set by generating predictions and comparing the results to the ground truth summary using the ROUGE metrics.

4.2.1 LexRank Algorithm

LexRank is an unsupervised graph-based algorithm for extractive text summarization. It models sentence importance as a graph, computes sentence similarity, and applies the PageRank algorithm to identify the most important sentences for the summary. The key steps are as follows:

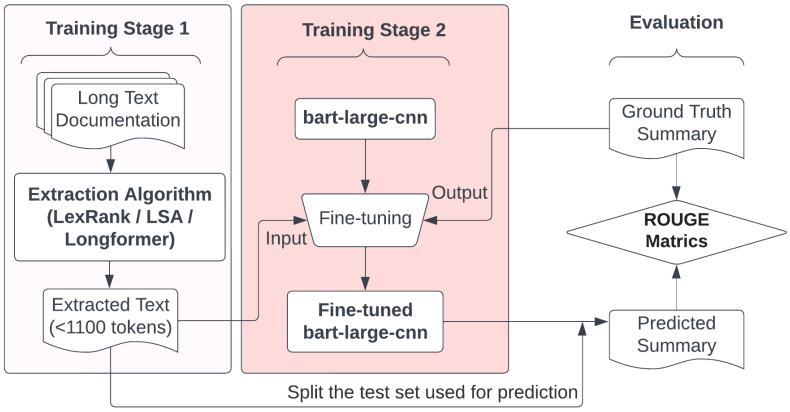

Figure 1: two stage model training architecture and evaluation process

1. **Sentence similarity**: Calculate the similarity between sentences using the cosine similarity or another similarity metric, and select the top-k sentences with the highest PageRank scores to create the summary. The similarity is given by:

$$[sim(s_i, s_j) = \frac{\vec{s_i} \cdot \vec{s_j}}{||\vec{s_i}|| \cdot ||\vec{s_j}||}]$$

2. **Sentence selection based on PageRank calculation**: Compute the importance (PageRank) of each sentence using the following iterative formula:

$$[\vec{PR}(s_i) = (1-d) + d \sum_{s_j \in Adj(s_i)} \frac{\vec{PR}(s_j)}{L(s_j)}]$$

where: $\vec{PR}(s_i)$ is the PageRank of sentence. d is the damping factor (usually set to 0.85). $Adj(s_i)$ represents the adjacent nodes (sentences) to $L(s_j)$ is the sum of the edge weights connected to sentence

4.2.2 LSA Algorithm

LSA can be used for extractive text summarization by projecting documents and sentences into a lower-dimensional semantic space, then selecting the most representative sentences for the summary. The key steps are as follows:

1. **SVD and dimensionality reduction**: Perform SVD on matrix A and retain the top-k singular values to obtain reduced matrices $\tilde{U}$, $\tilde{S}$, and $\tilde{V^T}$:

$$A = U \cdot S \cdot V^T \approx \tilde{U} \cdot \tilde{S} \cdot \tilde{V^T}$$

2. **Project sentences into semantic space**: Multiply the term-sentence matrix B by the reduced term matrix $\tilde{U}$:

$$SemanticSentences = \tilde{U}^T \cdot B$$

3. **Sentence selection**: Select the most representative sentences for the summary, either by choosing those closest to the centroid of the semantic space or by employing a clustering algorithm to group similar sentences and select a representative sentence from each cluster.

4.2.3 Longformer Extractive Model

Besides the unsupervised algorithms, we can also perform the extractive summary task using a supervised model. Specifically, we use the allenai/longformer-base-4096 model from Hugging Face[12], which is a BERT-like transformer model, to obtain an extracted summary that is around 1000 tokens for each sample. This is essentially an binary classification task where the model predicts whether to include a sentence in the original text to the summary or not. To do this, we first convert the abstractive datset into an extractive dataset. This is done by creating a completely extractive summary that maximizes the ROUGE score between itself and the ground-truth abstractive summary. We use the code `convert_to_extractive.py` from TransformerSum[13] to convert our pubmed abstractive dataset into an extractive version.

The training and validation sets for the extractive dataset have the sources and labels keys saved as json. The source value is a list of lists where each list contains a series of tokens. The labels value is a list of 0s (not in summary) and 1s (sentence should be in summary) that is the same length as the source value (the number of sentences). Each value in this list corresponds to a sentence in source. The testing set is special because it needs the source, labels, and target keys. The target key represents the target summary as a single string with a <q> between each sentence.

Once we have the extractive dataset, we train the longformer model for 2 epoch (around 50000 steps) and then generate extracted summaries for all samples in our training set.

4.2.4 BART Algorithm

We load the pretrained BART (Bidirectional and Auto-Regressive Transformers)[2] model "bart-cnn-large" from Hugging Face[11]. BART is a pre-trained model based on the Transformer architecture and employs a denoising autoencoder framework for pretraining. It will be fine-tuned for text summarization: input is the source text, and output is the target summary. During fine-tuning, the model learns to generate summaries conditioned on the input text. Given a new input text, the BART encoder generates contextualized representations, which the decoder then uses to generate a summary in an autoregressive fashion:

$$P(y_1, y_2, \ldots, y_T | x_1, x_2, \ldots, x_S) = \prod_{t=1}^{T} P(y_t | y_{<t}, x_1, x_2, \ldots, x_S)$$

where x_i and y_j represent input tokens and output tokens, respectively, S is the input length, and T is the output length.

4.3 LED-GAN

To improve the prediction result, we implement Generative Adversarial Network (GAN) for LED. The generator is LED and padding block to extend the length of generated result to 512. Reference abstract is also padded into length 512. The discriminator takes the tensor with length 512 as inputs and produces the probability of how likely the input is real. The discriminator has 2 linear layers, 1 leakyRelu layer and 1 Sigmoid layer. The GAN trains the generator and discriminator at the same time, with cross entropy as the loss function.

$$J = -(p \log(q) + (1 - p) \log(1 - q)) \tag{1}$$

In which p represents the actual classification and q represents the classification by discriminator. When the generator produces a better result, the loss of discriminator increases. When the discriminator has a better ability to discriminate the real and fake inputs, the loss of generator increases.

5 Experiments

5.1 Data

The Pubmed dataset for text summarization is a collection of scientific articles and their corresponding summaries from PubMed OpenAccess repositories[14]. The original dataset has a train/validation/test size of 120k/6k/6k samples, each with an average 3200 words article text and 220 summarization

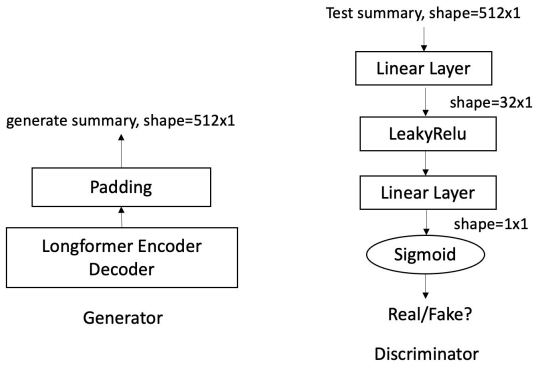

Figure 2: LED-GAN Architecture

Parameters	led-base-16384	bart-large-cnn
Learning rate	auto	auto
Batch size	2	2
Gradient accumulation steps	4	4
Optimizer	AdamW	AdamW
Training epochs	3	3
Document length	Up to 17k tokens	Up to 1024 tokens
Extraction algorithm	-	LexRank / LSA /Longformer

Table 1: Model Configurations

text. Since it takes too long to train and evaluate on such big data set, we only experiment on a subset PubMed-SMALL consisting 10k/3k/3k samples for train/validation/test. We choose this dataset to develop models that can effectively summarize scientific articles, which can help researchers quickly identify relevant information and improve their efficiency in reading and synthesizing new research findings.

5.2 Evaluation method

We use ROUGE (Recall-Oriented Understudy for Gisting Evaluation) to evaluate our summarization model, which includes precision, recall, and f-measurement from metrics ROUGE-1, ROUGE-2, and ROUGE-L [15]. ROUGE-1 measures the overlap of unigrams (single words) between the generated summary and the reference summary. ROUGE-2 extends this to the overlap of bigrams (pairs of adjacent words). ROUGE-L (Longest Common Subsequence) measures the longest common subsequence between the generated summary and the reference summary. Rouge measure both precision and recall to ensure a large overlap between the generated summary and the reference summary while keeping the generated text length as short as possible.

5.3 Experimental details

5.3.1 Two-stage model

Table 1 summarizes the experimental details for the LED baseline model and our proposed two-stage approach. Both models use a pre-trained base model, with the LED baseline utilizing the LED-base and our two-stage approach using the BART-base model combined with the sentence extraction algorithm. The learning rate is auto adjusted for both models, and the AdamW optimizer is used with batch size of 2 (We can only run a batch size 2 on AWS), while maintaining a gradient accumulation of 4 steps. This allows for efficient training of the two-stage approach while keeping GPU memory usage within acceptable limits.

Model	Summary Length
Ground Truth	220
LED Baseline	158
Two-Stage (LSA)	160
Two-Stage (LexRank)	159
Two-Stage (Longformer)	180
LED-GAN	372

Table 2: Average summary lengths for the ground truth and different summarization models.

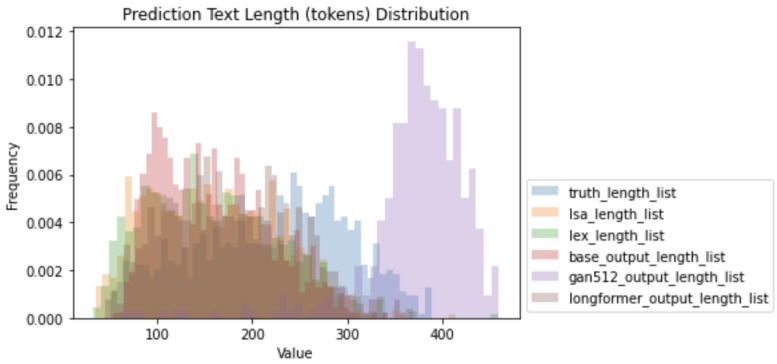

Figure 3: Distribution of the generated text summary length from tokens for different models

For our two-stage model, we set the extraction limit up to 1100 tokens, meaning that the we dynamically select top k sentences (total tokens number close but < 1100) ranked by the extraction methods in order to meet the 1024 token limit of the BART model.

5.3.2 LED-GAN

We also conduct an additional experiment by training LED with GAN. For this LED-GAN model, we load the LED "led-base-16384" as our generative model, and train the GAN defined in section 4 on our small dataset PubMed-SMALL. However, due to limited time and resources, we only have chance to run it for 1 epoch. The training takes about 40 hours and evaluation takes about 24 hours.

6 Result

We compared performance of our proposed two-stage approach using different extraction methods (LSA, LexRank, Longformer) and the LED-GAN variant with the LED baseline model. The evaluation is based on ROUGE scores (F1, recall, precision) and summary length.

6.1 Summary Length

During the stage 1 extraction process, the outputs extracted by both the LexRank and LSA algorithms have an average of 990 tokens. Outputs extracted by longformer has an average of 925 tokens. After the stage 2 abstraction generation process, the average lengths of the summaries generated by different models are shown in Table 2 and Figure 3. The average summary lengths for our two-stage approach are closer to the ground truth compared to the LED baseline model. However, the LED-GAN variant produces longer summaries, with an average length of 372.

6.2 ROUGE Scores

Table 3 shows the ROUGE scores for the different models. The bold number indicates the highest score for each column and the underlined number shows the second best. The LED baseline model achieves the highest F1 scores for all three Rouge metric. The performance of the Two-Stage model using longformer is very close to the baseline. It has the second highest F1 in all rouge scores.

Model	ROUGE-1			ROUGE-2			ROUGE-L		
	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
LED Baseline	**0.3945**	0.3909	0.4253	**0.1671**	0.1587	**0.1930**	**0.3624**	0.3588	0.3909
Two-Stage (LexRank)	0.3828	0.3712	**0.4322**	0.1504	0.1425	0.1794	0.3509	0.3403	**0.3962**
Two-Stage (LSA)	0.3715	0.3676	0.4127	0.1401	0.1339	0.1667	0.3413	0.3374	0.3793
Two-Stage (Longformer)	0.3936	0.4037	0.4113	0.1647	**0.1642**	0.1808	0.3601	0.3689	0.3767
LED-GAN	0.2991	**0.4195**	0.2442	0.1018	0.1538	0.0806	0.2746	**0.3847**	0.2243

Table 3: ROUGE scores (F1, recall, and precision) for the LED baseline and different models.

Model	Text Extraction Time (H)	Training Time (H)	Prediction Time (H)
LED Baseline	0	12	24
Two-Stage (LexRank)	10	2	1
Two-Stage (LSA)	8	2	1
Two-Stage (Longformer)	13	2	1
GAN512	0	40	24

Table 4: Times for Extracting 10k data, training BART/LED on 10k data, and generating summaries on 3k test data using BART/LED

It's also very competitive in recall and precision. The two-stage model using LexRank has its own strength. It performs overally better than that of LSA, and has the highest ROUGE-1 and ROUGE-L precision scores. The LED-GAN model tends to have a higher recall score in general, possibly due to its summary length, which is relatively long.

6.3 Training Time

Table 4 shows the extraction time to generate 10k extractive summaries, the training time of BART/LED to train the 10k data, and their time to generate 3k abstractive summaries. Note that the training and prediction time using BART model is significantly less than using the LED model. This can be attributed to the use of the extraction algorithm in the first stage, which reduces the complexity and speeds up the generation of abstractive summaries. We believe the result is very meaningful in the case where we need to generate abstractive summaries for a large amount of samples.

7 Analysis

7.1 Two stage model Performance Analysis

Our results indicate that this approach generates summaries with token lengths more closely aligned to ground truth compared to the LED baseline model. The supervised extraction method using Longformer model has very similar performance compared with the LED baseline. The LexRank algorithm demonstrates superior performance in terms of ROUGE scores compared to LSA. The LexRank algorithm, using PageRank for sentence importance, generates summaries that effectively capture relevant content from input text, particularly in scientific papers with straightforward language. However, despite its advantages over LSA, the two-stage LexRank model does not significantly outperform the LED baseline in terms of ROUGE scores. However, considering training time, the two-stage fine-tuned BART model significantly reduces the training process duration from 12 hours to 2 hours and accelerates evaluation time by a factor of four. This evaluation efficiency can be attributed to:

1. Model Complexity: The fine-tuned LED model is based on Longformer architecture and is more complex than fine-tuned BART. LED's sliding window self-attention mechanism contributes to increased processing time, particularly for longer input sequences.
2. Input Length Reduction: The two-stage approach utilizes LexRank in the first stage to extract key sentences and reduce input length to fewer than 1024 tokens. This significant reduction allows BART to process text more efficiently within its standard token limit.

BART, designed for shorter sequences and featuring a simpler attention mechanism, is faster when processing shorter inputs.

Regarding the lower recall values from the two-stage unsupervised algorithm, one possible explanation is that the abstraction stage does not capture the full context of the articles, resulting in potentially important context being excluded due to the 1024 input limit. A Pearson correlation test was performed to analyze the correlation between various ROUGE score measurements, revealing significant correlations between the performance of the unsupervised extraction stage and the final stage output. Regardless, there is still advantage of using an unsupervised algorithm as it does not require manually generating an extractive dataset and training an extractive summarization task, making it more accessible and easier to apply in practice.

As evident in Table 3, the supervised extraction algorithm has greatly improve the extraction stage's performance. Thus, we conclude that our two-stage method can achieve similar performance on long document summarization with much lower computational complexity and much less generation time.

7.2 GAN Performance Analysis

Additionally, we explored an alternative strategy to potentially outperform the baseline model without focusing on efficiency. The LED-GAN variant of our two-stage model generates longer summaries, resulting in improved ROUGE recall scores at the expense of precision. While the longer length contributes to lower precision and higher recall, the additional content does not significantly enhance the ROUGE scores. The limited performance compared to the LED baseline may be attributed to the computational complexity of our custom LED-GAN model and the extensive training time required for long-document datasets (40 hours per epoch). Due to the project's constraints, we only trained the model for one epoch and set the generated summary length to 512 tokens, causing LED to produce longer summarizations.

In future work, we plan to further train the LED-GAN model. Despite the current results, we anticipate that GAN can be employed to train the LED model once the standard trainer has reached its optimal performance. To prevent the LED from generating excessively long summaries, we can implement a reinforcement learning strategy similar to that used in SeqGAN, as proposed in previous research[16].

8 Conclusion

In conclusion, this paper presents an innovative two-stage approach for long document summarization that utilizes practical unsupervised extraction algorithms, a more powerful supervised extraction model and the efficient summary generation capabilities of BART [2]. Our method addresses the token limitations of current pre-trained models and offers a time-efficient alternative to existing techniques, such as LED [8]. Adding an unsupervised extraction stage on BART significantly reduces the computation complexity with only a slightly trade-off on performance. Moreover, our supervised extraction model, which employs Longformer, improves extraction performance and yields results that are competitive with LED, but with substantially less training and generation time complexity. Additionally, the LED-GAN variant also presents some potential to further improve the generated summaries as evident by its high recall scores. By demonstrating the effectiveness of our approach, we hope to inspire further research in long document summarization and the development of new techniques that can tackle the challenges of this important task.

9 Contribution

We equally contributed to this project, and we extend our gratitude to Professor Chris Manning, John Hewitt, and our the wonderful TA Elaine Sui for hosting this amazing course and assisting our project.

References

[1] Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2-3):103–233, 2011.

[2] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2020.

[3] 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.

[4] Gunes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[5] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[6] Yen-Chun Chen, Mohit Bansal, and Wei Wang. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, 2018.

[7] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3721–3731, 2019.

[8] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[9] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, 2018.

[10] Junhui Xu and Greg Durrett. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2963–2973, 2019.

[11] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing, 2019.

[12] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

[13] Hayden Housen. Transformersum. `https://github.com/HHousen/TransformerSum`, 2021.

[14] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018.

[15] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[16] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.