Abstract

COVID-19 research dataset has a pivotal role in the current pandemic as clinical researchers are dependent on this increasingly growing corpus to refer and draw insights. Summarization of lengthy research documents such as COVID-19 is still a challenge in NLP domain. The existing strategies involving Transformers have limitation with respect to the number of input tokens that can be processed as these architectures have quadratic dependency on the sequence length due to their full attention mechanism. The aim of this research work is to explore and build a method to generate context rich abstract summaries of research articles from CORD-19 data set by leveraging the potential of recently introduced sparse attention transformer i.e., BIGBIRD. BIGBIRD has the potential to handle input tokens up to 4096 which is almost 8 times higher than that of the existing transformers. The intent of the study is to explore the improvisation on CORD-19 data summarization task with BIGBIRD. In the process, the research aims to compare performance of BIGBIRD standalone architecture with that of BIGBIRD when ensembled with an extractive summarization layer. The intent here is to build a summarization strategy to improve the abstractive summary and, in the process, explore the possibility of leveraging an already existing extractive and abstractive summarization approach on BIGBIRD model for further improvisation. The intent here is to design an approach to innovate and achieve good performance metrics on COVID dataset which will eventually benefit the research community in the coronavirus crisis.

Table of Contents

Introduction

Automatic summarization (Ibrahim Altmami and El Bachir Menai, 2020) of lengthy documents using Machine Learning e.g., Legal data sets, Science / Medical Journals etc., has gained popularity since few years as the concept of chunking and filtering only the important information without losing the context from exhaustive factual documents not only saves time but also contributes to an improved understandability for target audience.

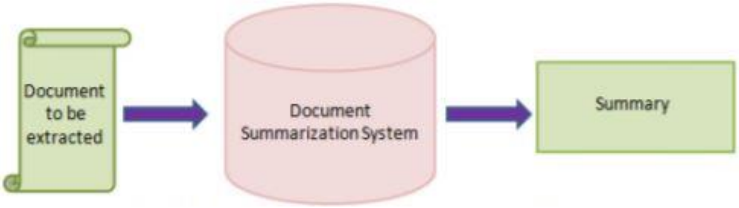


Figure 1.1 Concept of Summarization

Extractive Summarization (Moratanch and Chitrakala, 2017) and Abstractive Summarization (Moratanch and Chitrakala, 2017) are two existing approaches that has been implemented using Machine Learning and has found to be doing as extraction and abstraction concepts are subjective to domain at times, this always creates a scope of improvisation. Extractive summary retains the most important i.e., those sentences or phrases from the original text that are conveying the context of the original texts. In Abstract summarization new sentences are generated from the original text and at the same time preserving the context. Text Summarization is an important feature in medical domain where there is a need for summarization long scientific journals, articles, and related texts for research purpose.

Leading Research Groups in collaboration with White House have prepared CORD-19 i.e., COVID-19 Open Research Dataset (Lu Wang et al., 2020) and it has been made public for the research communities across the world. This data set contains over 4000,000 researched scholarly pandemic articles including over 150000 full length articles on coronaviruses including COVID-19 and SARS-CoV-2. This content can be leveraged by the Research communities across the world to synthesize new insights to battle the ongoing pandemic and help in insulating from the pandemics that may come up in the future. By leveraging Natural Language Processing and other AI techniques on this data, a lot of useful work can be done one of the ways to achieve this is by performing efficient text summarization. An efficient text summarization here will not only save time for the researchers in going through the exhaustive content but also leverage them to get hold of the context and help them draw new insights of each article which otherwise can be lost in exhaustive reading of the original content.

2. Background and related research

2.1 Introduction

Text summarization attempts to condense long texts without losing the context and at the same time preserving the important information. Initial works in areas of text summarization focused on extractive techniques aiming to retain the most important sentences from the documents.

Abstractive approach on the other hand involves summarizing the whole context in a more condensed structure drastically reducing the length of the content. Obviously abstractive summarization is a challenging task as generating creating abstractive demands command on the domain and as well as the natural language which is a tedious task for the machine.

Scientific document summarization is a special case of summarization as characteristics of scientific papers – length, writing styles, scientific terms and discourse structure demands an exclusive model consideration to maintain the context and at the same time retaining the accuracy of the topic. Researchers have engineered different approaches to address the challenges in Scientific document Summarization.

The metamorphosis of Long Short-Term Memory networks to attention mechanism combined with sequence-to-sequence framework was a pivotal in improvising language modelling tasks. Introduction of transformer architecture coupled with novel self-attention mechanism was a significant leap in language modeling task.

2.2 Extractive Summary Approaches

In a recent work extractive summarization as Text Matching (Zhong et al., 2020) is proposed. This is a novel summary framework which scores and extracts sentences one by one to form a summary, a strategy to formulate extractive summarization in form of semantic text matching problem. In this Siamese network structure and basic BERT(Devlin et al., 2019) have been combined to form Siamese-BERT architecture to compute the similarity between the source document and the candidate summary. In one of the models, Extractive summarization for lengthy structured content is attained by leveraging both local and global context from the entire document (Xiao and Carenini, 2020). This approach is inspired by natural topic-oriented structure of long documents which are created using human intelligence, where the binary conclusion of whether the sentence should be part of the summary is dependent on the sentence itself, the entire document and the current topic. The representation of document is cascading of the last 'n' hidden states of the forward and backward RNNs, while the representation of topic segment is done by leveraging LSTM-Minus method. REFRESH (Narayan et al., 2018) is a trained extractive summarization model for a globally optimized ROUGE metric and uses reinforcement learning. NEUSUM (Zhou et al., 2018) is an extractive summarization system that has the capability of scoring and selecting sentences

2.3 Graph Based Summarization

There have been Graph Based Text summarization models that have advantages and shortcomings. Word based graph methodology (Le and Le, 2013) was good at maintaining syntactic constraints but produced grammatically incorrect sentences and didn't consider meaning of word or phrases which led to loss of context in the generated summary.

Improvisation in Graph Based Methodology was seen in semantic graph reduction model (Moawad and Aref, 2012) which initially creates rich semantic graph followed by semantic graph reduction that includes the domain ontology class instances which helps to capture the meaning of sentences and even paragraphs which finally yields a better result in text generation step with less data loss.

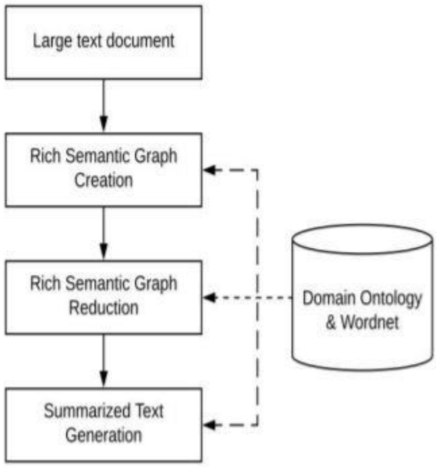

Large text document
Rich Semantic Graph Creation
Rich Semantic Graph Reduction
Summarized Text Generation
Domain Ontology & Wordnet

Figure 1.2 Workflow of semantic graph 1 model (Moawad and Aref, 2012)

In graph-based hybrid strategy to abstractive summarization applying Markov's Clustering (Sahoo et al., 2018), it was seen that it took into account sentence connections leading to sentence clustering followed by sentence positioning using sentence ranks and eventually sentence compression to produce effective summarization.

2.4 Reinforcement Learning for Text Summarization:

- Use of Reinforcement Learning has been explored for NLP tasks and based on this a text summarizer task combining Neural Networks and Reinforcement Learning (K et al., 2019) is proposed where Reinforcement Learning Algorithm is used to introduce feedback into the text summarization workflow which uses Encoder-Decoder module of Neural Networks. The comparison of the results with and without the feedback workflow added shows that use of RL facilitates the system to make corrections via feedback and produce a more relevant summary with a better Rouge Score.

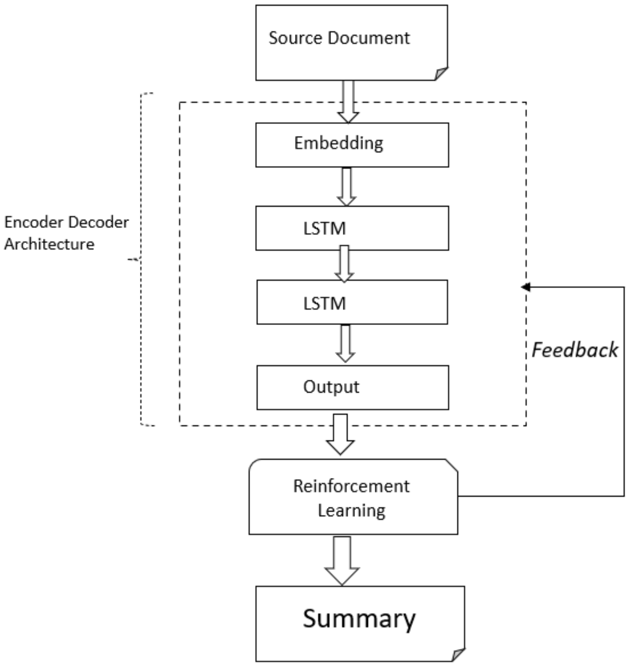


Figure 1.3 Reinforcement Learning for Text Summarization

- A hierarchical reinforcement learning model (Koupaee, 2018) approach that divides the task into a set of sub-tasks and optimization of individual sub tasks are done to optimize the abstractive summarization of texts. This approach experimented on WikiHow and CNN/Daily Mail data sets have given higher ROUGE scores.
- In one of the recent work in abstractive summarization the concept of discriminative adversarial search (Scialom et al., 2020) is proposed which uses Beam search which is de-facto algorithm used to decode generated sequences of text. Beam search has led to performance improvements of State of Art models Q&A Generation, Text Summarization and Neural Machine Translation

2.5 Transformers and Text Summarization

- The metamorphosis of Long Short-Term Memory networks to attention mechanism combined with sequence-to-sequence framework was a pivotal in improvising language modelling tasks. Introduction of transformer architecture coupled with novel self-attention mechanism was a significant leap in language modeling task. In Transformers architecture is Multi-head Attention Model i.e., self-attention is computed multiple times independently and in parallel and the outputs are concatenated followed by linear transformation.

- The key differentiator in Transformers is the application of a self-attention mechanism, which computes and evaluates the similarity scores for all pairs in an input sequence in parallel for individual tokens of the input sequence, completely bypassing the sequential dependency present in recurrent neural networks and thus outperforming previous sequential models.

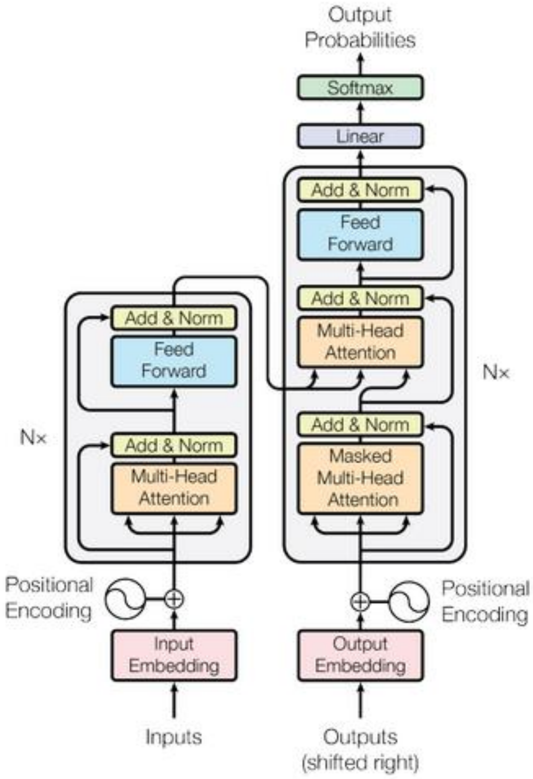


Figure 1.4 The Transformer – Model Architecture (Vaswani et al., 2002)

BERT (Devlin et al., 2019) implemented masked language modelling, which facilitated pre-training to learn interactions between left and right context words and enabling pre-trained deep bi-directional representations. The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based on its context. However, since

predictions are not made auto-regressively, effectiveness in generation tasks like abstract summarization effectiveness of BERT is reduced. BERTSUM (Liu, 2019) is a variant of BERT designed for Extractive Summarization which is achieved by modification of input sequence and embeddings of BERT. BERTSUM with Transformer is found to have achieved a very good performance on ROUGE metrics

T5 (Raffel et al., 2020) generalized the text-to-text framework to a variety of NLP tasks and showed the advantage of scaling up model size (to 11 billion parameters) and pre-training corpus, introducing C4, a massive text corpus derived from Common Crawl, which we also use in some of our models. T5 was pre-trained with randomly corrupted text spans of varying mask ratios and sizes of spans.
On a CNN/ Daily Mail Data set this has demonstrated a ROUGE-2-F score of 21.55.
Unidirectional language model such as GPT (Peters et al., 2018) can be used for text generation as tokens are predicted auto regressively but due its limitation of conditioning of words on leftward context, it is incapable of learning bi-directional interactions.

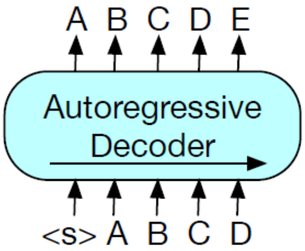


Figure 1.5 A schematic representation of GPT (Clark et al., 2020)

BART which is a denoising sequence-to-sequence pre-training for Natural Language Generation, Translation, and Comprehension by (Mike Lewis et al., 2019), performs pretraining of sequence-to -sequence models by denoising autoencoder. Training of BART is done by corrupting text with an arbitrary noising function and learning a model is made to reconstruct the original text. BART is one of the best performing transformers as it generalizes BERT, GPT and many other most pre-training schemes.

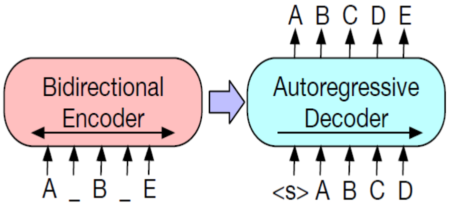


Figure 1.6 A schematic representation of BART (Lewis et al., 2019)

MASS (Song et al., 2019) proposed a model that involved masked sequence-to sequence generation to construe a sentence fragment from a given remaining part of the sentence that was randomly selected. UniLM (Dong et al., 2019) proposed a model that jointly trained on three types of modeling tasks: bidirectional (word-level mask followed by sentence prediction), sequence-to-sequence (word-level mask) prediction and

unidirectional (both left to- right and right-to-left). Similar hybrid language model XLNet (Yang et al., 2019), is a generalized autoregressive pretraining method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. In a recent work (Zhang et al., 2019) proposes pre-training with gap-Sentences for abstractive summarization. This model is popularly is known as PEGASUS model, in which the key sentences are removed or masked from an input source and are generated collectively as single output sequence from the remaining sentences, this approach is similar to an extractive summary. In this model instead of continuous text spans it masks multiple whole sentences and chooses sentences based on importance as output. Its architecture is a standard Transformer encoder in which Both Gap Sentence Generation and Masked Language Model are applied simultaneously to achieve the effective abstractive summarization.

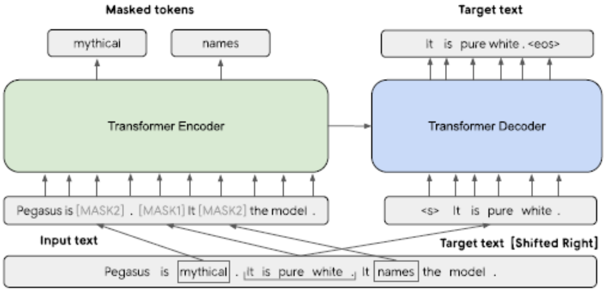
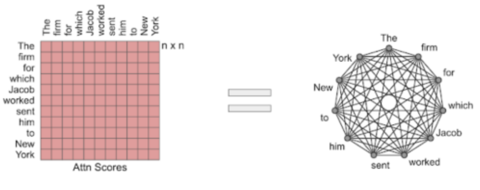

Masked tokens
mythical
names
Target text
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] It [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

Figure 1.7 Pegasus Architecture (Zhang et al., 2019)

2.6 Longformers

Limitations of Transformers:

Transformers models and their variants have found to revolutionize the way NLP problems such as summarization is perceived due to multi head self-attention mechanism, however they have a major drawback with specific to length of the source document. Transformer based Models fail to process documents where the length is long and this is majorly due to self-attention that scales quadratically with length of the sequence of the source input. The self-attention component in Transformer has time and memory complexity as O(n2) (where n is length of sequence input). Due to quadratic relationship of computational and memory requirements with input sequence length and existing hardware/ resource constraints transformers have limitation of input sequence length of 512 tokens, which limits its applicability for tasks that require longer contexts like Summarization.



Full attention can be viewed as a complete graph.

Figure 1.8 Full attention viewed as complete graph (ai.googleblog, n.d.)

<u>The Solution:</u>

Extended Transformer Construction (ETC) (Alberti et al., 2020) introduced a novel strategy for sparse attention, in which puts a limit on the computed pair of similarity scores based on structural information thereby reducing the quadratic dependency on input length to linear dependency and giving a superior performance for larger contexts.

BIGBIRD (Zaheer et al., 2020) is a sparse attention mechanism model is developed to address the quadratic complexity challenges of Transformers induced due to full self-attention mechanism. BIGBIRD utilizes MLM pretraining for base-sized models and for large sized models it is making use of summarization specific pretraining from Pegasus. The interesting feature here is that the sparse mechanism is implemented only at encoder size and this is primarily because the length of the sequence as output is quite small as compared to length of length of input sequence. The sparse mechanism at encoder level also helps to cover the input sequence entirely as the salient features in lengthy documents could be evenly distributed across the entire document.

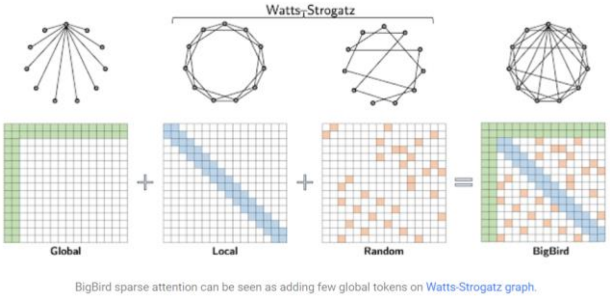

Watts-Strogatz
Global
Local
Random
BigBird
BigBird sparse attention can be seen as adding few global tokens on Watts-Strogatz graph.

Figure 1.9 BIGBIRD Sparse Attention seen on graph (ai.googleblog, n.d.)

Longformers have been introduced to address the limitations of Transformers and at the same time retaining the benefits of attention mechanism. Longformers are pre-trained counterparts of Transformers with attention mechanism scaling linearly with length of the sequence of the source input which makes processing of longer documents feasible. In Longformers attention mechanism combines a local windowed attention with a task motivated global attention which is a drop-in replacement for standard self-attention mechanism available in transformers.

Longformer-Encoder-Decoder (LED) (Beltagy et al., 2020) is variant of Longformer variant which supports generative sequence to sequence of lengthy documents. LED is based of BART's architecture with same number of layers with the difference that the position embedding is extended from 1K to 16K tokens to perform for longer input size. LED has been evaluated on long document summarization tasks such as scientific literature datasets and is found to outperform with a good Rouge Score.

REFORMER (Nikita Kitaev et al.,2020) proposes model to improve efficiency of Transformers by using reversible residual layers instead of standard layers and using attention with locality-sensitive hashing reducing the complexity to O(nlogn) where n is the length of input sequence.

2.7 Enhancing Abstraction using Extraction

In a recent work on neural document summarization using Transformer language models (Subramanian et al., 2019) author proposes combining extraction and abstraction strategies to come up with a more effective abstract summarization. The author has used encoder decoder architecture for Extractive summary where a sentence encoder is implemented using bi-directional LSTM and the decoder is implemented as autoregressive LSTM taking the sentence-level LSTM hidden state of the previously extracted sentence as input and predicting the next extracted sentence. For abstraction they have trained a Transfer Language Models with -220M parameters with 20 layers, 768 dimensional embeddings, 3072-dimensional position-wise MLPs and 12 attention heads. This model is trained on 4 components: Introduction, extracted sentences, abstract and rest of the paper. For creating abstracts of a long document, the trained model uses "Introduction" of the document as proxy to contain enough information for abstract along with the extracted content from extraction model. For a smaller document the introduction is the entire document. The combination has given a better rouge score. In another similar work titled “ Combination of abstractive and extractive approaches for summarization of long scientific texts” (Vladislav and Denis, 2020), the author proposes combining extractive and abstractive summary strategies with the usage of pre-trained Transformer Models. Extractive Model is trained as a classification to generating abstractive summary. They have experimented on 3 different architectures BERT, RoBERTa (Liu et al., 2019), and ELECTRA(Clark et al., 2020) for extractive Summary. Word piece tokenizer is used where BERT (Devlin et al., 2019) and ELECTRA(Clark et al., 2020) is experimented. In the process they have added special tokens like [math], [graph], [table], [equation] extracted using regex. Rouge Score is calculated using extracted summary and Ground Truth Abstract. As an outcome of the experiment BERT exhibited the highest performance Metrics as compared to other models. Extractive Summarization output was further fed into abstractive summarization model. The Abstract Summarization various pre-trained autoregressive language models were experimented out of which GPT-2 and BART (Lewis et al., 2019) were found to perform better as compared to other models. Combining BERT with BART and conditioning was done on input combination of introduction and conclusion (from the original document) along with the extractive summary (derived from the extraction model). The experiment resulted in the best ROUGE (Lin and Rey, 2001) score.

In an attempt to combine extractive and abstractive summary another improvisation that has been experimented is Keywords-Guided Abstractive Sentence Summarization (Li et al., 2020) where he proposes extracting overlapping words between the input and the reference as the ground-truth keywords followed by Multi-task learning i.e., generating summary using the input sentence and the ground-truth keywords. Keywords are generated using the trained keywords extractor for the input sentence in the training set and then fine-tuning the sentence summarizer using the original sentence and the predicted keywords. During testing, first keywords are generated using the trained keywords extractor for the input sentence and then the summary is produced using the input sentence and the predicted keywords. Similar Keyword based extraction followed by abstraction is proposed in work related to COVID-19 Medical Research dataset's abstract text summarization (Tan et al., 2020). He proposes a model where initially source text is scanned to extract keywords using token classification tools such as part of speech tagging packages of NLTK, or part of speech tagging of fine-tuned BERT token classifier. The extracted keywords are categorized into nouns, verbs and noun and verbs. Subsequently the keywords are paired with the gold summary abstract and model is processed using GPT-2.

2.8 Evaluation Metrics

Various performance metrics have been defined to measure the various NLP tasks. BLEU Score (BiLingual Evaluation Understudy) (Papineni et al., 2001) for language translation, SQuAD (Stanford Question Ansering Dataset) (Rajpurkar et al., 2015) for prediction of answers, GLUE (general Language Understanding Evaluation) (Wang et al., 2018) for collection of tasks and ROUGE (Lin and Rey, 2001) (Recall-Oriented Understudy for Gisting Evaluation) for text summarization.

ROUGE (Chin-Yew Lin and et al. 2020) compares machine generated summaries with reference summaries (human composed summaries) to determine the quality of summaries generated. It takes into account word pairs and n-gram sequences between the two summaries for comparison.

2.9 Discussion (key takeaways)

The study has covered the metamorphosis of Text Summarization in Machine learning encompassing techniques like graph-based approaches, leveraging Reinforcement Learning, Neural Networks, Transformers models and has finally put the spotlight on the gaps that are yet to be filled for long document summarization. Sparse-attention based models like BIGBIRD, LED (longtransformer) and Reformers have a lot of potential yet to be exploited in the field of text summarization for larger context domains. The evolution and metamorphosis of standard Transformer models has been discussed and along with the limitations of the same for larger documents has been explored in the study.

Literature review highlights the below focus areas that need more experimentation and analysis especially in the context of Text summarization of Long Documents such as scientific journals - (1) Exploring Sparse-attention based Transformer variants for long document summarization (2) Combining Extractive and Abstractive Summarization techniques to optimise the Abstractive Summarization output (3) Leveraging sparse attention based Transformers in combination strategy of extractive and abstractive strategy and evaluating the performance on basis of ROUGE Score.

2.10 Summary

This study reviewed a considerable number of literatures from conference proceedings, survey papers, thesis, journal articles and blogs to understand the evolution of Summarization in Machine Learning capacity. It has explored the latest strategies experimented and implemented to enhance extractive and abstractive summarization both in isolation as well as in combination. In the process of the review, we explored the various standard Transformer Models for text summarization and we also studied the limitations of these models for summarization of larger context datasets such as Scientific journals which are important as the salient features of the subject are spread across the entire document. One of the interesting finds in the entire review was introduction of Sparse-attention based Transformer variants such as BIGBIRD and Longtransformer designed to take longer input sequences.

In the last section, the study has discussed sparse-attention based transformer variants which have been introduced recently have a lot potential for longer documents and this can be leveraged for a good ensemble technique to come up with an effective summarization model and this is the motivation for the current study.

3. Research Questions (If any)

The following research questions are suggested for each of the research objective as highlighted as follows.

The target was to answer the following four questions.

- Can BIGBIRD be leveraged for automatic summarization of CORD-19 dataset?

- Can BIGBIRD address the gaps in the existing summarization strategies of long scientific documents such as COVID-19?
- Can adding extractive summarization layer before the BIGBIRD’s abstractive summarization model enhance the performance?
- Is the performance of BIGBIRD satisfactory for short, medium and long sized documents?
- Given an input size that levels the playing field for BART and BIGBIRD, is the performance of BIGBIRD better?

4. Aim and Objectives

The main aim of this research is to propose a generate improved abstract summaries of COVID-19 data set (Scientific document) by leveraging BIGBIRD. In the wake of recent pandemic, the need for robust research from large volume of data within a short time span has become inevitable. Extractive and Abstractive summarization approaches through recent advances in Open AI and NLP, can leverage processing and retrieval of comprehensive information from healthcare domain within a short time frame.

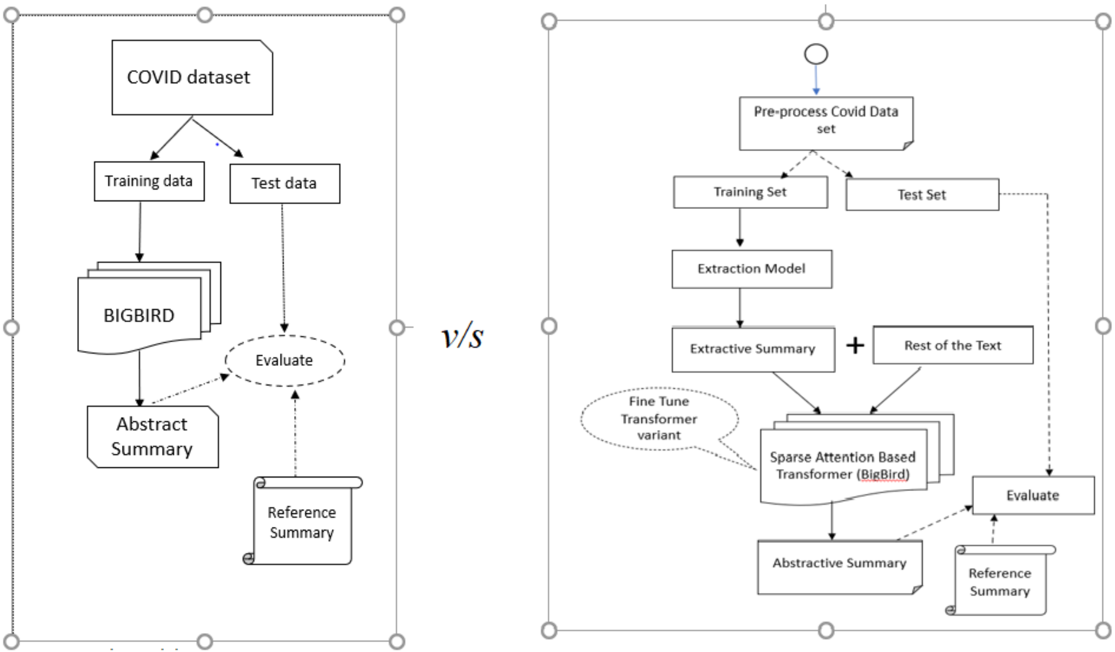
This research is intended to help the medical research community to keep up with the rapidly growing coronavirus literature and draw insights in a short turnaround time to fight the pandemic. The research objectives are formulated based on the aim of this study which are as follows:

- As BIGBIRD has capacity to process 4096 input tokens, train and evaluate the performance of BIGBIRD on CORD-19 summarization task
- Evaluate the effect of adding of extractive summarization layer before the final abstraction task in BIGBIRD
- Compare between BIGBIRD (standalone) and BIGBIRD with extraction ensemble to find the best approach for CORD-19 summarization task.
- Analyze the performance of BIGBIRD on three ranges of input document size i.e., short, medium and long
- Compare BIGBIRD with BART on input size that levels the playing field for comparison for COVID dataset.

5. Research Methodology

The study would like to explore how the latest sparse attention-based transformer variant such as BIGBIRD can be used to leverage an effective text summarization of Covid dataset containing long scientific document corpus. The study will further explore the feasibility of improvisation with an ensemble model leveraging both extractive and abstractive techniques with the above-mentioned transformer variant. The study will include evaluation of performance through ROUGE metrics.

Subsequent subsections will discuss all steps that were addressed in order to achieve the goal. The flow diagram in Figure 3.2 depicts planned sequence of activities in the modelling and evaluation phase. The flow diagram in Figure 3.1 depicts the overview of the two separate summarization pipelines proposed to be built for comparison.



BIGBIRD standalone

BIGBIRD ensemble with extraction Layer

Figure 0.1 High Level Research Approach

The overall research work can be categorized in four broad set of activities

- Data Analysis & Pre-processing
- Training BIGBIRD on CORD-19 dataset
- Training BIGBIRD ensemble with extraction layer on CORD-19 dataset
- Performing evaluation on two summarization pipelines and draw comparisons and propose the best model.

The above specified research activities have been elaborated below.

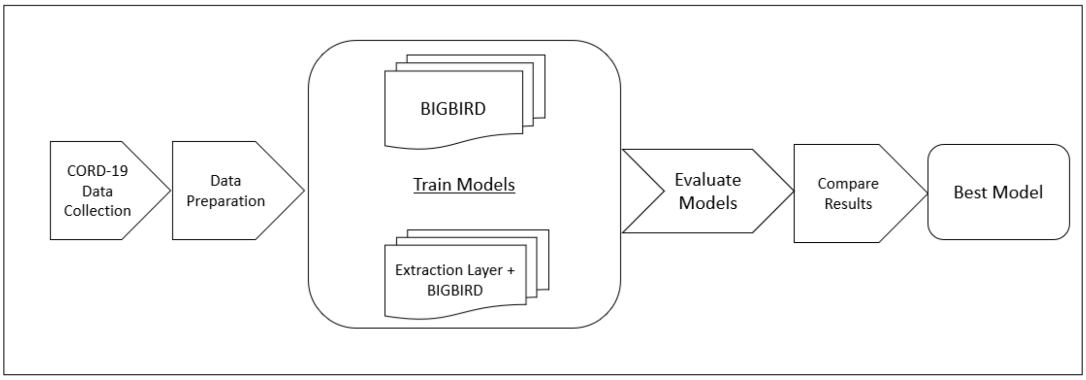


Figure 0.2 Research Framework

5.2 Data Analysis & Pre-processing

This section will provide an overview or probable steps to be performed to analyze, transform and process the existing data and prepare the training data

5.2.1 Dataset Description

Leading Research Groups in collaboration with White House have prepared CORD-19 i.e., COVID-19 Open Research Dataset and it has been made public for the research communities across the world. This data set contains over 4000,000 researched scholarly pandemic articles including over 150000 full length articles on coronaviruses including COVID-19 and SARS-CoV-2. This content can be leveraged by the Research communities across the world to synthesize new insights to battle the ongoing pandemic and help in insulating from the pandemics that may come up in the future. By leveraging Natural Language Processing and other AI techniques on this data, a lot of useful work can be done one of the ways to achieve this is by performing efficient Text Summarization.

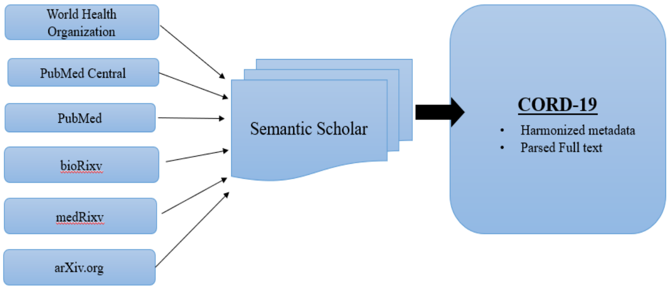


Figure 0.3 Sources of CORD-19 Research data

The article text holds full text of a paper, section name is a list of papers sections, sections contain full text of paper divided into sections. That means, we could identify which text is from the introduction section, which is from the abstract section or conclusion.

Table 0.1 Data statistics of CORD-19 dataset

Data Set	Total Scientific Articles	Average no. of words per articles	Reference Summaries (Average no. of words)
CORD-19	490904	4000	200

Table 0.2 Research Article Size Distribution

	50%	75%	max
Article (wordcount)	3440	5165	254446
Abstract(wordcount)	157	234	5091

The data statistics shows that 75% of the total articles in the document have length up to 5000 words and abstract length as 234 words. Hence it is clear that COVID dataset is dominated by lengthy articles.

5.2.2 Data Pre-processing

- The data from PDFs and other paper documents are parsed to provide structured text which is preserved in JSON schema and converted into readable csv file format.
- The important columns that have been extracted are paper_id, title, authors, affiliations, abstract, text and bibliography.
- As part of data cleaning too short papers and too long papers, papers without abstract were excluded.
- For the scope of the training, only those records with article length up to 5000 words were included as part of training strategy. This was due to the fact that articles with word length 5000 words constituted 75% of the total corpus and BIGBIRD's capacity is 4096 input tokens. So, the strategy is catering to majority of the records in the corpus and is aligned with the capacity of BIGBIRD.

- Removal of irrelevant characters was done as part of data cleaning.
- For the purpose of training only articles and abstract attributes were retained.
- Sections which are subtopics in the articles were extracted and concatenated as single string. This value was added as a derived attribute “sections” in the training data.
- The data was split into training, validation and test sets.

5.3 Training BIGBIRD Model

BIGBIRD has greater capacity to process input tokens as compared to other transformers. As the data corpus used in the training data has articles length up to 5000 words and BIGBIRD’s capacity is 4096 input tokens, BIGBIRD was trained on the final training data directly. The model was trained on two attributes i.e., articles and sections.

5.3.1 Training BIGBIRD Ensemble with Extraction Layer

The ensemble planned as part of this strategy consists of an Extraction Layer followed by BIGBIRD’s abstractive summarization layer. As part of this ensemble the output extraction layer becomes one of the inputs to BIGBIRD. The final training data on which the BIGBIRD is trained consists of:

Introduction + Extractive Summarization + Results

5.3.1 Building the Extractive Summarization Layer

As the length of the input articles in the training samples is up to 5000 words with average number of records falling in the range of 3000 words, an extractive strategy that can cover the entire document is needed.

The approach proposed here is a combination of BERT and K-medoid clustering (Miller, 2019). A pre-trained BERT model is used for sentence embedding. Each sentence in the article is transformed into 768 high dimensional representation. K-medoid clustering analysis is done on the transformed high dimensional representations. The entire flow results in cluster centers which represents the semantic centers of the analyzed text. These semantic centers when collated together can constitute the extractive summary of the articles. The extractive summary is thus constructed using the cluster centers. The BERT model used in the architecture is DistilBERT (Sanh et al., 2019) from Huggingface Transformer. The extraction %

targeted in the architecture is 40-50% of the entire document. The final layer in the ensemble is the abstractive summarization layer using BIGBIRD. The data preparation for this model includes extraction of Introduction and Results sections from the articles. For purpose of training only those records from training data are considered which have Introduction and Results sections. The motive behind selecting Introduction and Results section is based on the general assumption and observation that Introduction and Results sections contain the main context of research articles. Extracted summary from the extraction layer is taken and combined with Introduction and Results sections. The final training dataset consists of "Introduction + Extracted summary + Results"

5.4 Model Evaluation

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin and Rey, 2001) is one of the widely used performance metric used for evaluating performance of Text Summarization tasks in NLP. ROUGE compares machine generated summaries with reference summaries (ground truth) to determine the quality of summaries generated. It takes into account word pairs and n-gram sequences between the two summaries for comparison. The various ROUGE measures available are ROUGE-N, ROUGE-L, ROUGE-W and ROUGE -S. ROUGE score is used to evaluate the performance of each model.

5.5 Comparison between the trained models

The two summarization pipelines built are evaluated on ROUGE scores and the quality of summary generated as that of the ground truth summary. Comparisons are drawn and the best model is recommended.

6. Expected Outcomes

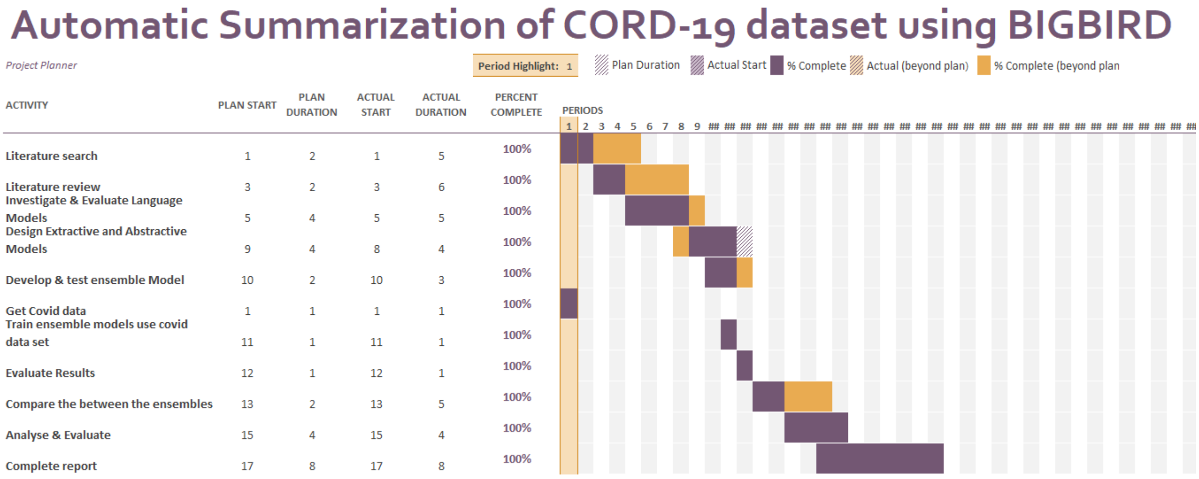
- The research is expected give clarity over the research question i.e., whether BIGBIRD can address the existing gaps in summarization task of long scientific research articles such as COVID-19 dataset.
- The research is expected to clarify if performance of BIGBIRD can be improved by combining extractive and abstractive summarization approach.
- From the research work the expected outcome is to identify the best performing summarization ensemble using BIGBIRD on CORD-19 dataset.

- The research is expected evaluate the performance of BIGBIRD when compared to other popular Transformers such as BART on summarization task of COVID research articles.

7. Requirements / resources:

- Cloud lab with GPUs

8. Research Plan



Automatic Summarization of CORD-19 dataset using BIGBIRD

Project Planner

Period Highlight: 1

▨ Plan Duration ▨ Actual Start ■ % Complete ▨ Actual (beyond plan) ■ % Complete (beyond plan

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Literature search	1	2	1	5	100%
Literature review	3	2	3	6	100%
Investigate & Evaluate Language Models	5	4	5	5	100%
Design Extractive and Abstractive Models	9	4	8	4	100%
Develop & test ensemble Model	10	2	10	3	100%
Get Covid data	1	1	1	1	100%
Train ensemble models use covid data set	11	1	11	1	100%
Evaluate Results	12	1	12	1	100%
Compare the between the ensembles	13	2	13	5	100%
Analyse & Evaluate	15	4	15	4	100%
Complete report	17	8	17	8	100%

REFERENCES

Alberti, C., Cvicek, V., Ainslie, J., Onta, S., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q. and Yang, L., (2020) ETC: Encoding Long and Structured Inputs in Transformers ˜.

Anon (n.d.) ai.googleblog.

Beltagy, I., Peters, M.E. and Cohan, A., (2020) *Longformer: The Long-Document Transformer*. *arXiv*.

Clark, K., Luong, M.T., Le, Q. V. and Manning, C.D., (2020) *Electra: Pre-training text encoders as discriminators rather than generators*. *arXiv*.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. Association for Computational Linguistics (ACL), pp.4171–4186.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.W., (2019) Unified Language Model Pre-training for Natural Language Understanding and Generation. *arXiv*, NeurIPS.

Ibrahim Altmami, N. and El Bachir Menai, M., (2020) *Automatic summarization of scientific articles: A survey*. *Journal of King Saud University - Computer and Information Sciences*, Available at: https://doi.org/10.1016/j.jksuci.2020.04.020.

K, B., P C, R. and Murali, R., (2019) Automatic Text Summarizing System Using Reinforcement Learning Technique. *SSRN Electronic Journal*.

Koupaee, M., (2018) Abstractive Text Summarization Using Hierarchical Reinforcement Learning. pp.1920–1949.

Le, H.T. and Le, T.M., (2013) An approach to abstractive text summarization. *2013 International Conference on Soft Computing and Pattern Recognition, SoCPaR 2013*, pp.371–376.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L., (2019) BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv*.

Li, H., Zhu, J., Zhang, J., Zong, C. and He, X., (2020) Keywords-Guided Abstractive Sentence Summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 3405, pp.8196–8203.

Lin, C. and Rey, M., (2001) ROUGE : A Package for Automatic Evaluation of Summaries.

Liu, Y., (2019) *Fine-tune BERT for Extractive Summarization*. *arXiv*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., (2019) RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*, 1.

Lu Wang, L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., Kinney, R., Liu, Z., Merrill, W., Mooney, P., Murdick, D., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., Wade, A.D., Wang, K., Wilhelm, C., Xie, B., Raymond, D., Weld, D.S., Etzioni, O. and Kohlmeier, S., (2020) CORD-19: The Covid-19 Open Research Dataset. *ArXiv*. [online] Available at: http://www.ncbi.nlm.nih.gov/pubmed/32510522%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC7251955.

Moawad, I.F. and Aref, M., (2012) Semantic graph reduction approach for abstractive Text Summarization. *Proceedings - ICCES 2012: 2012 International Conference on Computer Engineering and Systems*, pp.132–138.

Moratanch, N. and Chitrakala, S., (2017) A survey on extractive text summarization. *International Conference on Computer, Communication, and Signal Processing: Special Focus on IoT, ICCCSP 2017*, January.

Narayan, S., Cohen, S.B. and Lapata, M., (2018) Ranking sentences for extractive summarization with reinforcement learning. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, pp.1747–1759.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J., (2001) BLEU: a method for automatic evaluation of machine translation. *ACL*, pp.311–318.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., (2018) Improving Language Understanding by. *OpenAI*, [online] pp.1–10. Available at: https://gluebenchmark.com/leaderboard%0Ahttps://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Raffel, C., Roberts, A. and Liu, P.J., (2020) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. 21, pp.1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., (2015) SQuAD: 100,000+ Questions for Machine Comprehension of Text. ii.

Sahoo, D., Bhoi, A. and Balabantaray, R.C., (2018) Hybrid Approach to Abstractive Summarization. *Procedia Computer Science*, [online] 132Iccids, pp.1228–1237. Available at: https://doi.org/10.1016/j.procs.2018.05.038.

Scialom, T., Sylvain, P.D., Benjamin, L. and Jacopo, P., (2020) Discriminative Adversarial Search for Abstractive Summarization.

Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.Y., (2019) MASS: Masked sequence to sequence pre-training for language generation. *36th International Conference on Machine Learning, ICML 2019*, 2019-June, pp.10384–10394.

Subramanian, S., Li, R., Pilault, J. and Pal, C., (2019) On extractive and abstractive neural document summarization with transformer language models. *arXiv*.

Tan, B., Kieuvongngam, V. and Niu, Y., (2020) *Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2*. *arXiv*.

Vaswani, A., Noam Shazeer, Niki Parmar and Jakob Uszkoreit∗, (2002) The Transformer-Attention Is All You Need. *IEEE Industry Applications Magazine*, 81, pp.8–15.

Vladislav, T. and Denis, S., (2020) *Combination of abstractive and extractive approaches for summarization of long scientific texts*. *arXiv*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S.R., (2018) *GLUE: A multi-task benchmark and analysis platform for natural language understanding*. *arXiv*.

Xiao, W. and Carenini, G., (2020) Extractive summarization of long documents by combining global and local context. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp.3011–3021.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q. V., (2019) XLNet: Generalized autoregressive pretraining for language understanding. *arXiv*, NeurIPS, pp.1–18.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L. and Ahmed, A., (2020) Big Bird: Transformers for Longer Sequences. *arXiv*, NeurIPS.

Zhang, J., Zhao, Y., Saleh, M. and Liu, P.J., (2019) *PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization*. *arXiv*.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X. and Huang, X., (2020) Extractive summarization as text matching. *arXiv*.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M. and Zhao, T., (2018) Neural document summarization by jointly learning to score and select sentences. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1, pp.654–663.

Alberti, C., Cvicek, V., Ainslie, J., Onta, S., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q. and Yang, L., (2020) ETC: Encoding Long and Structured Inputs in Transformers ˜.

Anon (n.d.) ai.googleblog.

Beltagy, I., Peters, M.E. and Cohan, A., (2020) *Longformer: The Long-Document Transformer*. *arXiv*.

Clark, K., Luong, M.T., Le, Q. V. and Manning, C.D., (2020) *Electra: Pre-training text encoders as discriminators rather than generators*. *arXiv*.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. Association for Computational Linguistics (ACL), pp.4171–4186.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.W., (2019) Unified Language Model Pre-training for Natural Language Understanding and Generation. *arXiv*, NeurIPS.

Ibrahim Altmami, N. and El Bachir Menai, M., (2020) *Automatic summarization of scientific articles: A survey*. *Journal of King Saud University - Computer and Information Sciences*, Available at: https://doi.org/10.1016/j.jksuci.2020.04.020.

K, B., P C, R. and Murali, R., (2019) Automatic Text Summarizing System Using Reinforcement Learning Technique. *SSRN Electronic Journal*.

Koupaee, M., (2018) Abstractive Text Summarization Using Hierarchical Reinforcement Learning. pp.1920–1949.

Le, H.T. and Le, T.M., (2013) An approach to abstractive text summarization. *2013 International Conference on Soft Computing and Pattern Recognition, SoCPaR 2013*, pp.371–376.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L., (2019) BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv*.

Li, H., Zhu, J., Zhang, J., Zong, C. and He, X., (2020) Keywords-Guided Abstractive Sentence Summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 3405, pp.8196–8203.

Lin, C. and Rey, M., (2001) ROUGE : A Package for Automatic Evaluation of Summaries.

Liu, Y., (2019) *Fine-tune BERT for Extractive Summarization*. *arXiv*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., (2019) RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*, 1.

Lu Wang, L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., Kinney, R., Liu, Z., Merrill, W., Mooney, P., Murdick, D., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., Wade, A.D., Wang, K., Wilhelm, C., Xie, B., Raymond, D., Weld, D.S., Etzioni, O. and Kohlmeier, S., (2020) CORD-19: The Covid-19 Open Research Dataset. *ArXiv*. [online] Available at: http://www.ncbi.nlm.nih.gov/pubmed/32510522%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC7251955.

Moawad, I.F. and Aref, M., (2012) Semantic graph reduction approach for abstractive Text Summarization. *Proceedings - ICCES 2012: 2012 International Conference on Computer Engineering and Systems*, pp.132–138.

Moratanch, N. and Chitrakala, S., (2017) A survey on extractive text summarization. *International Conference on Computer, Communication, and Signal Processing: Special Focus on IoT, ICCCSP 2017*, January.

Narayan, S., Cohen, S.B. and Lapata, M., (2018) Ranking sentences for extractive summarization with reinforcement learning. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1, pp.1747–1759.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J., (2001) BLEU: a method for automatic evaluation of machine translation. *ACL*, pp.311–318.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., (2018) Improving Language Understanding by. *OpenAI*, [online] pp.1–10. Available at: https://gluebenchmark.com/leaderboard%0Ahttps://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Raffel, C., Roberts, A. and Liu, P.J., (2020) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. 21, pp.1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., (2015) SQuAD: 100,000+ Questions for Machine Comprehension of Text. ii.

Sahoo, D., Bhoi, A. and Balabantaray, R.C., (2018) Hybrid Approach to Abstractive Summarization. *Procedia Computer Science*, [online] 132Iccids, pp.1228–1237. Available at: https://doi.org/10.1016/j.procs.2018.05.038.

Scialom, T., Sylvain, P.D., Benjamin, L. and Jacopo, P., (2020) Discriminative Adversarial Search for Abstractive Summarization.

Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.Y., (2019) MASS: Masked sequence to sequence pre-training for language generation. *36th International Conference on Machine Learning, ICML 2019*, 2019-June, pp.10384–10394.

Subramanian, S., Li, R., Pilault, J. and Pal, C., (2019) On extractive and abstractive neural document summarization with transformer language models. *arXiv*.

Tan, B., Kieuvongngam, V. and Niu, Y., (2020) *Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2*. *arXiv*.

Vaswani, A., Noam Shazeer, Niki Parmar and Jakob Uszkoreit∗, (2002) The Transformer-Attention Is All You Need. *IEEE Industry Applications Magazine*, 81, pp.8–15.

Vladislav, T. and Denis, S., (2020) *Combination of abstractive and extractive approaches for summarization of long scientific texts*. *arXiv*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S.R., (2018) *GLUE: A multi-task benchmark and analysis platform for natural language understanding*. *arXiv*.

Xiao, W. and Carenini, G., (2020) Extractive summarization of long documents by combining global and local context. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp.3011–3021.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q. V., (2019) XLNet: Generalized autoregressive pretraining for language understanding. *arXiv*, NeurIPS, pp.1–18.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L. and Ahmed, A., (2020) Big Bird: Transformers for

Longer Sequences. *arXiv*, NeurIPS.

Zhang, J., Zhao, Y., Saleh, M. and Liu, P.J., (2019) *PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization*. *arXiv*.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X. and Huang, X., (2020) Extractive summarization as text matching. *arXiv*.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M. and Zhao, T., (2018) Neural document summarization by jointly learning to score and select sentences. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1, pp.654–663.