Mini Project III

Indian Institute of Information Technology
Allahabad
Semester VII, B.Tech (IT)

Project Report

For

News Summarization using Deep Learning Approaches

Done by

Harsh Sharma - IIT2019097,

Tamoghno Bhattacharya - IIT2019103,

Gurmeet Singh - IIT2019121,

Anurag Bandyopadhyay - IIT2019126,

Aamej Shreyansh - IIT2019153.

Instructor: Dr. Nabajyoti Mazumdar.

CONTENTS

I) ABSTRACT

Abstract - Automatic Text Summarization (ATS) is growing in relevance due to the exponential growth of databases around the world in the form of news articles, legal documents, research papers, etc. It is a very rich domain for research as it has multiple cross-domain applications such as in natural language understanding, and information retrieval. Although pre-trained language models based on transformers have emerged as the state of the art for the problem, their performance falls off on longer documents. In this paper, we discuss the various widely accepted approaches used and present our own extractive summarization algorithm, by using a sentence ranking algorithm to pick the sentences from the text containing maximum information about the document, then feeding them into a pre-trained BERT language model as encoder and a transformer decoder, which is fine tuned on the text summarization dataset. In this way, we aim to alleviate the shortcomings of transformer-based approaches and generate concise, meaningful and exhaustive summaries for our text documents.

II) INTRODUCTION

In the age of the internet, massive volumes of textual data have been amassed and are still growing at a rapid pace every day in the form of cloud resources like websites, blogs, news, user messages, and social networking sites. Numerous articles, books, novels, blogs, documents, scientific papers, chats, biomedical materials, and other archives also include substantial textual content. Information overload is hence becoming a graver problem. Users must browse a variety of lengthy texts almost daily and spend a lot of time weeding out unnecessary material, which significantly lowers their productivity. Because of this, it is now urgent and fundamentally important to provide a solution for how to swiftly locate the information required from the text resources, then summarize and compress it.

One technique for extracting the key ideas from a paper or group of linked documents and condensing them into a concise version while maintaining their main points is Text Summarizing. It decreases the amount of time needed to read an entire document and solves the space issue caused by the requirement to store a lot of data. Traditional methods for text summarization are those that date back to the early 2000s. Traditional text summarizing techniques demand a deeper document-level understanding to identify the crucial keywords. Manual summarization of text includes reading the entire text before summarizing, which is very expensive and vulnerable to error in large amounts of data. Automatic text summarizing (ATS) offers a practical solution to this issue.

The objective of an Automatic Text Summarization system is to generate a summary of the document(s) that are fed into the system, by keeping the main ideas intact [1] and to keep repetition to a minimum [2]. Automatic text summarization helps users as well as other systems such as information retrieval systems as it eliminates the need to store entire documents. Search engine optimization is another domain in which summarization can be applied, since it reduces the complexity of the queries and is more efficient in terms of space. ATS systems can be divided into single-document summarization systems and multi-document summarization systems. In the former, a single document is used to create the summary, but in the latter, a collection of documents is used to create the summary. The extractive, abstractive, or hybrid text summarization methodologies are used while designing ATS systems. The extractive approach chooses the key phrases and employs them to produce the summary. The abstractive approach chooses phrases that are different from the original input for constructing the summary. In the hybrid technique, extractive and abstractive methods are combined.

The problem with an extractive approach is that since we only pick entities from the original document, there is no paraphrasing and the model is not very flexible. Also, there is a major problem with redundancy and cohesion since the selected sentences need to be linked together properly. An abstractive approach is more flexible since it generates words of its own like humans do, however such models deal with errors with semantics and grammar and it is difficult to generate high quality abstractive summary through them [3]. Hybrid approaches try to combine the advantages of both methods however since they only employ abstractive methods on an extracted version of the original text, there is a chance of missing out on crucial information which should be summarized.

The architecture of an ATS system is;

1. Pre-processing:- Using a variety of linguistic approaches, including sentence segmentation, word tokenization, stop-word removal, part-of-speech tagging, etc., we create a structured representation of the original text [4].
2. Processing:- We apply techniques to the pre-processed document(s) in order to transform them into the summary utilizing one of the text summarizing methods.
3. Post-processing:- Before generating the final summary, various issues with the created summary sentences are resolved, including anaphoras.

III) LITERATURE SURVEY

Automatic text summary started to gain popularity in the 1950s. A method for extracting meaningful sentences from text was developed by Luhn et al. [5] utilizing factors like word and phrase frequency. They suggested eliminating frequently used terms and weighting phrases in a text based on their frequency. A paradigm based on key phrases was presented by Edmunson et al. [6] in addition to the customary frequency-dependent weights. As was already established, extractive summarizing techniques produce summaries by choosing a subset of the sentences from the original text. Only the most significant sentences from the input are included in these summaries. Basically an extractive summarization process consists of three major steps: First an intermediate representation of the input document is constructed, then the different segments in the document are scored, then a summary based on the scores of the different segments is constructed. Topic representation and indicator representation are the two different categories of intermediate representation approaches.

The goal of topic representation is to frame the text in terms of the cited topics. Latent semantic analysis, topic word approaches, frequency driven approaches, and Bayesian topic models are a few of the methodologies [7]. An important score is assigned to each sentence and then important sentences are selected based on the scores. The sentence selection procedure may be greedy or may be formulated as an optimization problem.

Indicator representation aims to map the sentences on the basis of a set of features obtained through feature selection, and pick sentences which are indicative of high content. The techniques include graph methods, probabilistic machine learning methods such as Naive Bayes classifier, and semi-supervised learning.

Henning et al. [8] proposed a method of sentence extraction that mapped sentences to concepts in an ontology. They improved the semantic representation of sentences by taking the ontology's aspects into consideration, which helped improve the semantic representation of the sentences and led to earlier selection. Using the UMLS medical ontology to generate a summary of medical material, Chen et al. [9] suggested a user query-based text summarizer. By employing the YAGO ontology, Baralis et al. [10] suggested a YAGO-based summarizer that extracted the key ideas from the articles.

Abstractive text summarization produces summaries by paraphrasing various sections of the text. The idea is to recognize the main concepts and generate summaries like a human does. It includes Graph based, Tree based, Rule based, Template based, Ontology based, Semantic based, and Deep Learning based methods.

In Graph based approaches, Ganesan, Zhai, and Han (2010) [12] proposed the "Opiniosis" abstractive summarizer. In Tree-based approaches, sentences that are comparable and share information are accumulated using tree-based algorithms to create the abstractive summary [13]. In order for rule-based techniques to function, questions must be created and summaries must be produced as the responses [13]. These employ information extraction rules and content selection heuristics. In methods based on templates. Depending on the genre of the input material, an abstractive summary can be created by using the data in the text to fill the slots in the appropriate preset templates [14]. Correct information from the input text can be recovered using ontology-based algorithms to produce an abstractive summary [15]. The final abstractive summary is created by the natural language generation system using verb and noun phrases after the input document(s) have been represented using semantic-based approaches [13].

The advent of Big Data and Data Mining processes have led to generation of very large datasets. Deep Learning algorithms make use of this huge volume of data for language generation tasks and have led to a revolution in Natural Language Processing (NLP). Sequence to Sequence models were introduced in 2014 [16] that were found to be significantly better in terms of machine translation. These models also make abstractive summarization feasible [17]. The words are converted into a vector representation using embeddings and then are fed to an encoder model such as an LSTM which converts it into a fixed length representation. The representation is then fed into a decoder model which generates the summary given the representation and the words that are already generated. However, memorizing long sentences is not feasible for an encoder network. Vaswani et al. 2017 [18] proposed an attention mechanism for machine translation where, for generating the translation of a word the decoder only pays attention to parts of the original sentence and is more representative of the ways humans translate sentences. Although the attention mechanisms of self-attention and multi-head attention were very successful, they suffered from severe computational complexity. Specifically the computation time of an attention network is quadratic with respect to the length of the input sequence.

In the networks that were explored till now, fixed embeddings were used for each word. However, the meaning of a word depends on the context as well as the position in the sentence. This, combined with the parallelism of the attention mechanism using a CNN style of processing led to the Transformer architecture which gave a significant performance gain for various natural language tasks.

Pre-trained language models have lately been a significant tool for obtaining impressive advances in a wide range of natural language tasks [19,20,21,22,23]. A new language representation model called Bidirectional Encoder Representation from Transformers (BERT) [21] is trained on a corpus of 3,300M words using masked language modeling and a "next sentence prediction" task. Since these language models can be used for downstream tasks such as text summarization, many variants of transformer networks with BERT as encoder have been invented and represent the state-of-the-art for the text summarization problem.

One of the most cutting-edge abstractive methods for text summarizing is the T5 Transformer [24], which employs a transformer language model and views summary as a text-to-text learning objective. An objective for summarization that is self-supervised is used in PEGASUS (Pre-training with extracted gap-sentences for abstractive summarization) [25]. In the pre-training stage sentences are removed from documents and the model tries to recover them.

BERTSUM [26] is an extension of BERT which inserts multiple [CLS] symbols to learn sentence representations and uses interval segmentation embeddings to distinguish the sentences. BART [27] uses a denoising autoencoder for pretraining sequence-to-sequence models. The training procedure involves corrupting the text with an arbitrary noisy function and learning a model to reconstruct the original text. Some other models include a transformer with W-drop which uses a word dropout perturbation to perform training [28]. Also BART can be trained with R-Drop to regularize the dropout by placing additional constraints on the model parameters [29].

Among hybrid methods, we have extractive to abstractive methods such as “EA-LTS” [30] which uses a graph model for sentence selection and an RNN-based encoder-decoder with pointer and attention mechanism. We also have extractive to shallow abstractive methods like “SumItUp” [31]. It basically uses statistical features such as sentence length, position, TF-IDF etc and semantic features such as text emotion for extraction, followed by a language generator which is a combination of WordNet, Lesk and Part-of-Speech tagger. Hybrid summarization approaches are a promising research direction [32].

IV) DATASET

We will be using the CNN/DailyMail news summarization dataset for training and evaluation of our proposed methodology. CNN/Daily Mail is a single-document dataset for summarization of news documents. The documents contain stories from CNN and Daily Mail news websites, and human generated abstractive summaries are provided for each document. The corpus has 286,817 training, 13,368 validation and 11,487 testing documents. The documents in the training set have on average 766 words, containing an average of 29.74 sentences per document while the summaries have on average 53 words, containing an average of 3.72 sentences per document.

V) PROPOSED METHODOLOGY

The problem with transformer and BERT-based models is that it is inefficient in language generation tasks involving long documents. Specifically, the average case complexity of transformers is given as $O(d*n^2)$ where d is the input embedding dimension, and n is the sequence length. To solve this problem, BERT fixes the maximum input size to 512 tokens, which are generally taken from the start of the sequence, which might be inadequate for long inputs. Another method is to take 512 tokens at a time from the input, and aggregate the outputs accordingly. This method is computationally very expensive and does not take into account attention from each of the groups of tokens.

To deal with this, we propose a two-stage extractive approach in which the extractive model picks the top sentences from the document according to a novel topic representation method using sentence ranking. The selected sentences are then fed into a transformer model for summarization. The transformer model in our case consists of a pre-trained BERT as encoder, with a standard transformer decoder. The entire model is then evaluated as per the dataset. Our hypothesis is that the resulting summaries are more representative of the actual content of the articles, with the transformer model being more efficient in terms of computational resources.

We start with tokenizing the text, performing stop word removal, and stemming to convert words to their base forms. For single-document summarization, we calculate the Term Frequency-Inverse Sentence Frequency (TF-ISF) [33] score for each word in each sentence. The TF-ISF score is computed as -

$$TF - ISF\,(t, s) \; = \; TF\,(t, s) \; * \; log(\frac{1 + n}{1 + sf(t)}) \qquad (1)$$

where TF (t,s) is the frequency of word t in sentence s, n is the total number of sentences in a document, and sf (t) is the sentence frequency of t, i.e., the number of sentences in which t occurs. For multi-document summarization, we calculate the Term Frequency-Inverse Document Frequency (TF-IDF) score [33] for each word in each document. The TF-IDF score is computed as -

$$TF - IDF\,(t, d) \; = \; TF\,(t, d) \; * \; log(\frac{1 + m}{1 + df(t)}) \qquad (2)$$

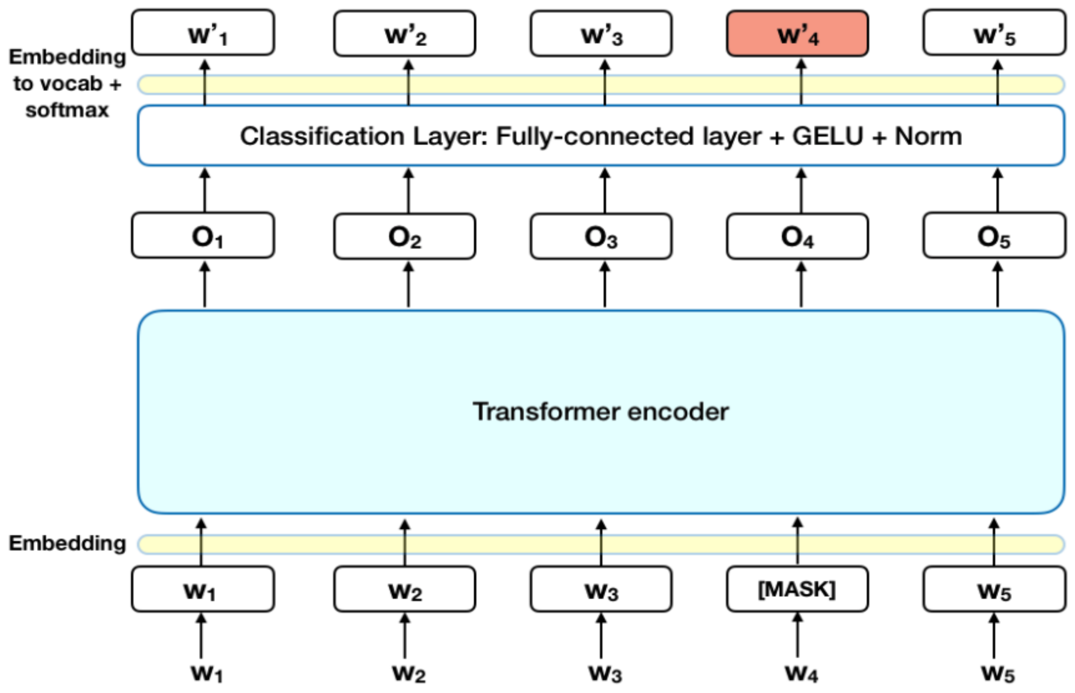
where TF (t,d) is the frequency of word t in document d, m is the total number of documents, and df (t) is the document frequency of t, i.e., the number of documents in which t occurs.

The sentence scoring system is formulated as follows -

$$Score\,(s) \; = \; \sum_{t \in s} Word\,Score\,(t, s) \qquad (3)$$

The top n/k sentences are greedily selected using this score and are fed into the abstractive model, where k is the shrinkage factor we want for our pre-trained BERT encoder model. An important feature to note is that the pre-processing steps are only done for the purpose of feature selection. Although the scores are computed based on the pre-processed sentences, the sentences are selected from the original document. This is done because a BERT model has to learn the context of each word and removing stop-words and stemming changes the context and meaning significantly, decreasing the generated summary’s quality.

The extracted sentences are then fed into a transformer model with BERTSUM [26] as the network. The standard BERT architecture is as follows -



w'1
w'2
w'3
w'4
w'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
w1
w2
w3
[MASK]
w5
w1
w2
w3
w4
w5

Figure 1: BERT Architecture with masked LM

[Source: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270]

BERT makes use of the transformer encoder to generate contextual embeddings for each word in the input sequence. The transformer network is as follows -

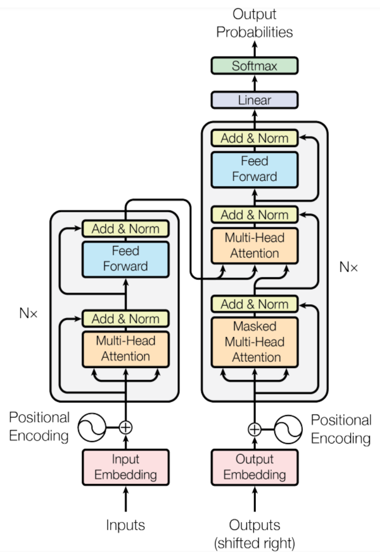


Figure 2: Transformer Network [18]

The transformer encoder is on the left side and the decoder is on the right. The inputs are fed into an embedding layer which generates word embeddings, which are then concatenated with positional encodings which are indicative of the position of the word in the sentence. The vectors are then fed into a multi-head attention network which generates the output as follows -

$$Attention\ (Q, K, V)\ =\ Softmax(\frac{QK^T}{\sqrt{d_k}}) \qquad (4)$$

where Q is the query matrix, K is the key matrix, and d_k is the dimension of the key queries. These are then fed into a feed forward neural network with residual skip connections, and repeated N times to form the encoder network.

The decoder network then follows a similar embedding mechanism with masked attention, which only takes into account the words in the output sequence that come before the next predicted word for the self-attention calculation. After that, there is an additional multi-head attention layer with residual skip connections. However, the key and value pairs for this come from the encoder network's output. The output probabilities are maximized with respect to the actual words in the training set.

BERT makes use of the bidirectional encoder architecture of the transformer, which aims to learn a language model using 2 types of pre-training - Masked LM and Next Sentence Prediction [21]. The idea is that once the encoder has been trained for such a task, it can be applied to downstream NLP applications such as text summarization, question answering, named entity recognition etc. BERTSUM is an extension of BERT, which inserts multiple [CLS] symbols for the purpose of learning sentence representations, and uses segment embeddings to distinguish alternate sentences. The architecture is as follows -

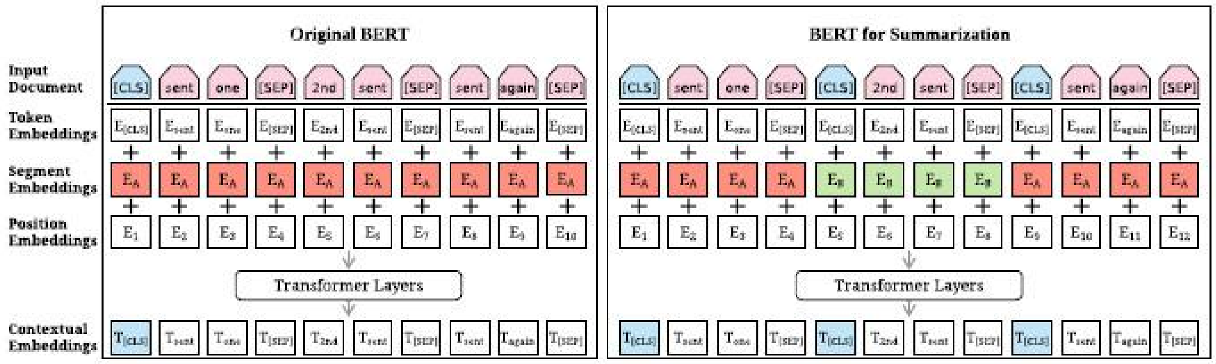


Figure 3: Architecture of Original BERT (left) vs BERTSUM (right) [26]

For each sentence, $T_{[CLS]}$ becomes the vector representation for the sentence output by the BERTSUM model, which is the output corresponding to the [CLS] symbol which aggregates information for the following sentence based on the entire article. Positional encodings are added to $T_{[CLS]}$ for indicating sentence position, and many inter-sentence transformer layers (similar to that of the encoder part) are stacked on top.

$$\hat{h}^{l} = LN(h^{l-1} + MHAtt(h^{l-1})) \qquad (5)$$

$$h^{l} = LN(\hat{h}^{l} + FFN(\hat{h}^{l})) \qquad (6)$$

where c^{l} is the output of the Multi-Head Attention Network in the l'th layer of the transformer and $\hat{h}^{l}$ is the output of the entire layer. $h^{0} = PosEmb(T)$ where T represents the sequence of sentence vectors output by BERTSUM and PosEmb is a function that adds sinusoidal positional embeddings [18] to T. LN represents the layer normalization operation and MHAtt is the multi-head attention operation.

The final output layer is a fully-connected layer with sigmoid as the activation function -

$$\hat{y}_{i} = \sigma(W_{o}h_{i}^{L} + b_{o}) \qquad (7)$$

where h_{i}^{L} is the output vector for the i'th sentence, after applying L layers of the transformer. L = 2 was found to perform the best. W_{o} and b_{o} are the weights and biases of the output layer, and $\hat{y}_{i}$ is the output probability of the sentence being included in the final summary. The loss of the model is calculated using binary cross-entropy using prediction $\hat{y}_{i}$ and gold label y_{i} which are generated using a greedy algorithm which maximizes the ROGUE-2 score [34].

During test time, the model is used to generate probabilistic scores for each sentence in the input document. The sentences are ranked from highest to lowest, and the top 3 are chosen as the summary output. During sentence selection, trigram blocking is used to reduce redundancy [35], which basically means that if a candidate sentence has a trigram overlap with the already selected sentences, it is skipped. We try three BERT variants - the BERT-Base, DistilBERT and the MobileBERT variants and report qualitative and performance results for the same.

VI) EVALUATION METRICS

The evaluation of text summarization is a challenging task and a research field in itself. It is complex since computers have to identify the content and phrases that are similar to the human-generated summary and score them based on their similarity. Two sentences phrased differently can have the same meaning. Also, placement of key phrases is indicative of content expression. Hence, traditional evaluation metrics like accuracy, precision, recall of F1-score cannot be used to judge the quality of a summary.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [36] is a series of evaluations designed specifically for ATS and Machine Translation tasks. It compares a machine-generated candidate summary with that of the human references. There are 5 measures - ROGUE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. Among these, we will be using ROUGE-N and ROUGE-L for evaluation of our generated summaries.

ROUGE-N score for a reference and candidate summary is defined as -

$$ROUGE - N = \frac{Count_{match}\,(n-gram)}{Count\,(n-gram)} \qquad (8)$$

where $\text{Count}_{\text{match}}$ (n-gram) is the number of matching n-grams between the reference and candidate summary, and Count (n-gram) is the number of n-grams in the reference summary. The final ROUGE-N score is the maximum score among all of the references.

ROUGE-L score for a reference and candidate summary is defined as -

$$ROUGE - L = \frac{LCS\,(reference, candidate)}{Count\,(reference)} \qquad (9)$$

where LCS (reference, candidate) is the length of the longest common subsequence between reference and candidate summary, and Count (reference) is the length of the reference summary (the number of unigrams).

We will be using ROUGE-1, ROUGE-2 and ROUGE-L for evaluation of our model-generated summaries and the results will be presented accordingly.

VII) RESULTS

We perform a qualitative as well as a performance analysis of our model for different values of k, which is the shrinkage hyperparameter. The inference process is carried out in a non-GPU enabled Kaggle environment with a Intel Xeon processor running at 2.3 GHz and a total RAM size of 30 GB. The qualitative analysis is done using the ROUGE-1, ROUGE-2 and ROUGE-L metrics and the unit we use for performance measurement is the time it takes in milliseconds to perform a single prediction.

1. BERT-Base -

k	ROUGE-1	ROUGE-2	ROUGE-L	Time (in ms)
1	0.38	0.17	0.36	1580
2	0.36	0.15	0.33	1450
3	0.34	0.13	0.31	1210
4	0.32	0.12	0.29	1250
5	0.30	0.11	0.28	1120

Table 1: BERT-Base Comparison

2. DistilBERT -

k	ROUGE-1	ROUGE-2	ROUGE-L	Time (in ms)
1	0.40	0.18	0.37	1190
2	0.37	0.15	0.34	1050
3	0.34	0.13	0.31	1120
4	0.32	0.12	0.29	1020
5	0.31	0.11	0.28	980

Table 2: DistilBERT Comparison

3. MobileBERT -

k	ROUGE-1	ROUGE-2	ROUGE-L	Time (in ms)
1	0.36	0.15	0.33	775
2	0.35	0.14	0.32	714
3	0.32	0.12	0.30	719
4	0.30	0.11	0.28	649
5	0.29	0.10	0.26	575

Table 3: MobileBERT Comparison

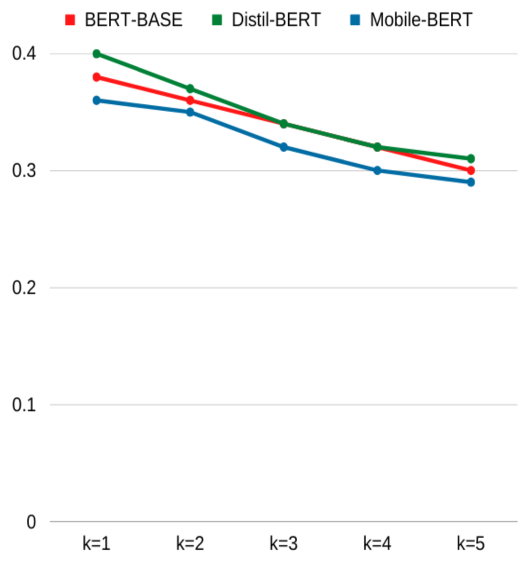


Figure 4: Qualitative curve of ROUGE-1 vs k

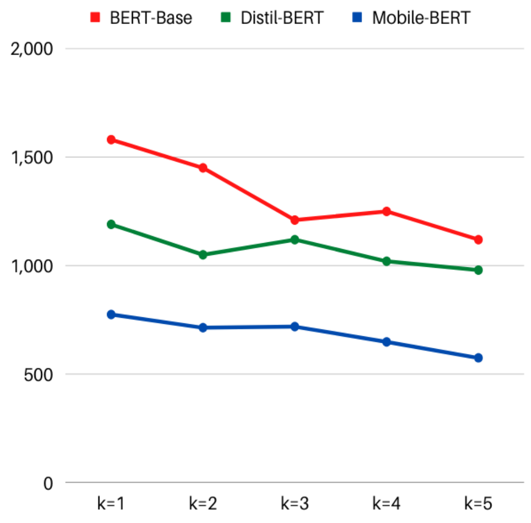
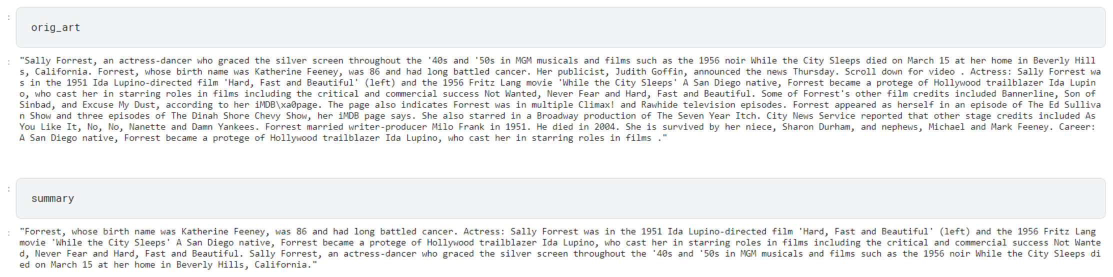


Figure 5: Performance curve of Time taken (in ms) vs k

DistilBERT performs the best, based on qualitative comparison, whereas MobileBERT performs the best in terms of performance comparison. k represents a tradeoff between summary quality and performance of the model. Increasing the value of k results in shorter input summaries to the BERT model, hence resulting in better performance. However, the decreased input content results in a decrease in the quality of the output summaries. BERT-Base surprisingly has slightly worse results than DistilBERT, which might be due to the fact that a simpler model might work with the shrunk input in a better way. Our methodology performs close to the SOTA for our task, while being much more computationally efficient than other methods, hence making it ideal for deployment on low-resource devices and embedded systems.

We explore the quality of the generated summary for different values of k for one particular article and observe the following outputs for DistilBERT model -

For k = 1,



```
orig_art
```

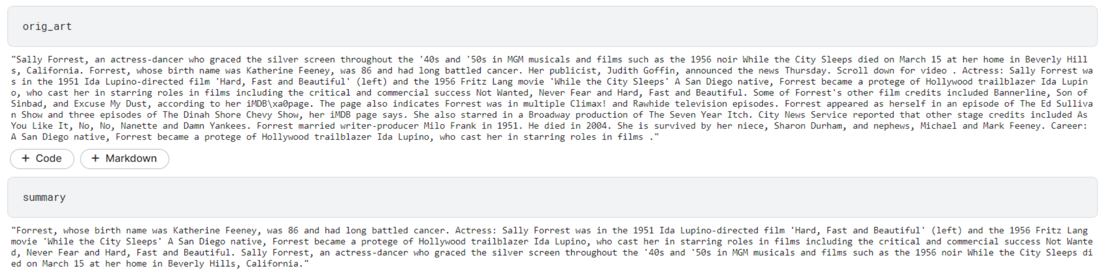
```
"Sally Forrest, an actress-dancer who graced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hill
s, California. Forrest, whose birth name was Katherine Feeney, was 86 and had long battled cancer. Her publicist, Judith Goffin, announced the news Thursday. Scroll down for video . Actress: Sally Forrest wa
s in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupin
o, who cast her in starring roles in films including the critical and commercial success Not Wanted, Never Fear and Hard, Fast and Beautiful. Some of Forrest's other film credits included Bannerline, Son of
Sinbad, and Excuse My Dust, according to her iMDB\xa0page. The page also indicates Forrest was in multiple Climax! and Rawhide television episodes. Forrest appeared as herself in an episode of The Ed Sulliva
n Show and three episodes of The Dinah Shore Chevy Show, her iMDB page says. She also starred in a Broadway production of The Seven Year Itch. City News Service reported that other stage credits included As
You Like It, No, No, Nanette and Damn Yankees. Forrest married writer-producer Milo Frank in 1951. He died in 2004. She is survived by her niece, Sharon Durham, and nephews, Michael and Mark Feeney. Career:
A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films ."
```

```
summary
```

```
"Forrest, whose birth name was Katherine Feeney, was 86 and had long battled cancer. Actress: Sally Forrest was in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang
movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films including the critical and commercial success Not Wante
d, Never Fear and Hard, Fast and Beautiful. Sally Forrest, an actress-dancer who graced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps di
ed on March 15 at her home in Beverly Hills, California."
```

Fig. 6: Summary generated for k = 1

For k = 2,



```
orig_art
```

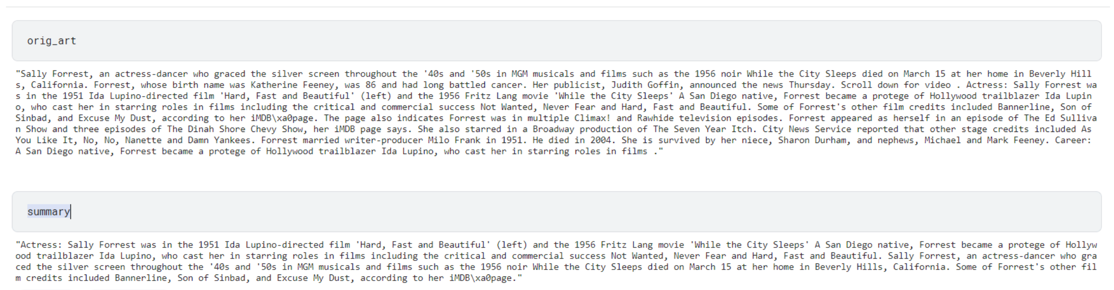
```
"Sally Forrest, an actress-dancer who graced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hill
s, California. Forrest, whose birth name was Katherine Feeney, was 86 and had long battled cancer. Her publicist, Judith Goffin, announced the news Thursday. Scroll down for video . Actress: Sally Forrest wa
s in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupin
o, who cast her in starring roles in films including the critical and commercial success Not Wanted, Never Fear and Hard, Fast and Beautiful. Some of Forrest's other film credits included Bannerline, Son of
Sinbad, and Excuse My Dust, according to her iMDB\xa0page. The page also indicates Forrest was in multiple Climax! and Rawhide television episodes. Forrest appeared as herself in an episode of The Ed Sulliva
n Show and three episodes of The Dinah Shore Chevy Show, her iMDB page says. She also starred in a Broadway production of The Seven Year Itch. City News Service reported that other stage credits included As
You Like It, No, No, Nanette and Damn Yankees. Forrest married writer-producer Milo Frank in 1951. He died in 2004. She is survived by her niece, Sharon Durham, and nephews, Michael and Mark Feeney. Career:
A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films ."
```

```
summary
```

```
"Forrest, whose birth name was Katherine Feeney, was 86 and had long battled cancer. Actress: Sally Forrest was in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang
movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films including the critical and commercial success Not Wante
d, Never Fear and Hard, Fast and Beautiful. Sally Forrest, an actress-dancer who graced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps di
ed on March 15 at her home in Beverly Hills, California."
```

Fig. 7: Summary generated for k = 2

For k = 3,



```
orig_art
```

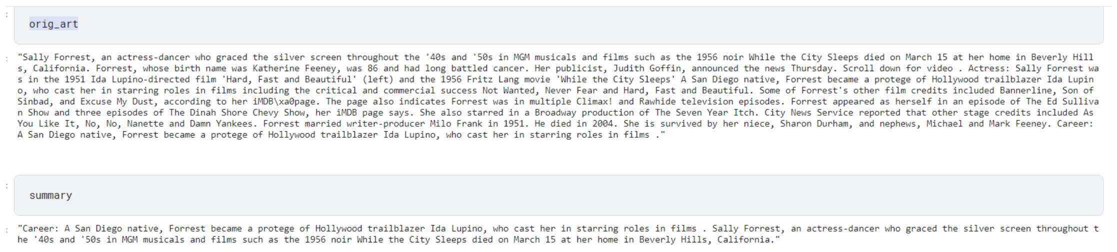
```
"Sally Forrest, an actress-dancer who graced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hill
s, California. Forrest, whose birth name was Katherine Feeney, was 86 and had long battled cancer. Her publicist, Judith Goffin, announced the news Thursday. Scroll down for video . Actress: Sally Forrest wa
s in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupin
o, who cast her in starring roles in films including the critical and commercial success Not Wanted, Never Fear and Hard, Fast and Beautiful. Some of Forrest's other film credits included Bannerline, Son of
Sinbad, and Excuse My Dust, according to her iMDB\xa0page. The page also indicates Forrest was in multiple Climax! and Rawhide television episodes. Forrest appeared as herself in an episode of The Ed Sulliva
n Show and three episodes of The Dinah Shore Chevy Show, her iMDB page says. She also starred in a Broadway production of The Seven Year Itch. City News Service reported that other stage credits included As
You Like It, No, No, Nanette and Damn Yankees. Forrest married writer-producer Milo Frank in 1951. He died in 2004. She is survived by her niece, Sharon Durham, and nephews, Michael and Mark Feeney. Career:
A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films ."
```

```
summary
```

```
"Actress: Sally Forrest was in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollyw
ood trailblazer Ida Lupino, who cast her in starring roles in films including the critical and commercial success Not Wanted, Never Fear and Hard, Fast and Beautiful. Sally Forrest, an actress-dancer who gra
ced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hills, California. Some of Forrest's other fil
m credits included Bannerline, Son of Sinbad, and Excuse My Dust, according to her iMDB\xa0page."
```

Fig. 8: Summary generated for k = 3

For k = 4,



```
orig_art
```

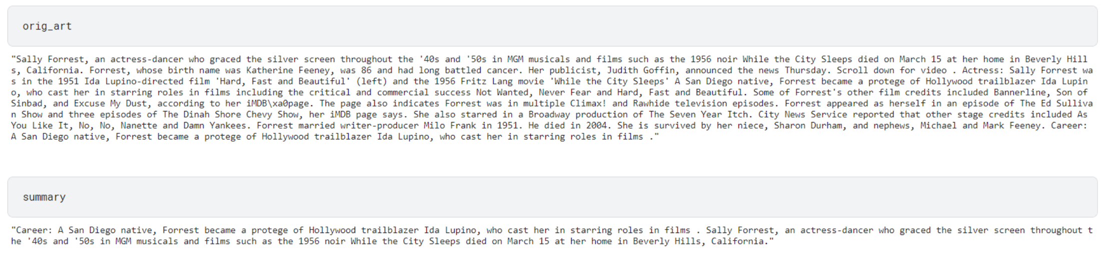
```
"Sally Forrest, an actress-dancer who graced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hill
s, California. Forrest, whose birth name was Katherine Feeney, was 86 and had long battled cancer. Her publicist, Judith Goffin, announced the news Thursday. Scroll down for video . Actress: Sally Forrest wa
s in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupin
o, who cast her in starring roles in films including the critical and commercial success Not Wanted, Never Fear and Hard, Fast and Beautiful. Some of Forrest's other film credits included Bannerline, Son of
Sinbad, and Excuse My Dust, according to her iMDB\xa0page. The page also indicates Forrest was in multiple Climax! and Rawhide television episodes. Forrest appeared as herself in an episode of The Ed Sulliva
n Show and three episodes of The Dinah Shore Chevy Show, her iMDB page says. She also starred in a Broadway production of The Seven Year Itch. City News Service reported that other stage credits included As
You Like It, No, No, Nanette and Damn Yankees. Forrest married writer-producer Milo Frank in 1951. He died in 2004. She is survived by her niece, Sharon Durham, and nephews, Michael and Mark Feeney. Career:
A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films ."
```

```
summary
```

```
"Career: A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films . Sally Forrest, an actress-dancer who graced the silver screen throughout t
he '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hills, California."
```

Fig. 9: Summary generated for k = 4

For k = 5,



```
orig_art
```

```
"Sally Forrest, an actress-dancer who graced the silver screen throughout the '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hill
s, California. Forrest, whose birth name was Katherine Feeney, was 86 and had long battled cancer. Her publicist, Judith Goffin, announced the news Thursday. Scroll down for video . Actress: Sally Forrest wa
s in the 1951 Ida Lupino-directed film 'Hard, Fast and Beautiful' (left) and the 1956 Fritz Lang movie 'While the City Sleeps' A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupin
o, who cast her in starring roles in films including the critical and commercial success Not Wanted, Never Fear and Hard, Fast and Beautiful. Some of Forrest's other film credits included Bannerline, Son of
Sinbad, and Excuse My Dust, according to her iMDB\xa0page. The page also indicates Forrest was in multiple Climax! and Rawhide television episodes. Forrest appeared as herself in an episode of The Ed Sulliva
n Show and three episodes of The Dinah Shore Chevy Show, her iMDB page says. She also starred in a Broadway production of The Seven Year Itch. City News Service reported that other stage credits included As
You Like It, No, No, Nanette and Damn Yankees. Forrest married writer-producer Milo Frank in 1951. He died in 2004. She is survived by her niece, Sharon Durham, and nephews, Michael and Mark Feeney. Career:
A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films ."
```

```
summary
```

```
"Career: A San Diego native, Forrest became a protege of Hollywood trailblazer Ida Lupino, who cast her in starring roles in films . Sally Forrest, an actress-dancer who graced the silver screen throughout t
he '40s and '50s in MGM musicals and films such as the 1956 noir While the City Sleeps died on March 15 at her home in Beverly Hills, California."
```

Fig. 10: Summary generated for k = 5

We explore the quality of summaries generated for different articles by varying the value of k used and we find that the quality starts dipping rapidly from k = 3 onwards. Using these observations, we propose using a value of k = 2 as the optimal value of k which balances quality and performance of summarization.

VIII) REFERENCES

1. Radev, D. R., Hovy, E. & McKeown, K. (2002). Introduction to the special issue on summarization. Computational Linguistics, 28(4), 399-408. doi: 10.1162/089120102762671927.

2. Moratanch, N. & Chitrakala, (2017). A Survey on Extractive Text Summarization.Paper presented at the International Conference on Computer,Communication and Signal Processing (ICCCSP), Chennai.

3. Hou, Liwei & Hu, Po & Bei, Chao. (2018). Abstractive Document Summarization via Neural Model with Joint Attention. 10.1007/978-3-319-73618-1_28.
4. Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. Journal of Emerging Technologies in Web Intelligence, 2(3), 258–268.
5. H. P. Luhn, ``The automatic creation of literature abstracts," IBM J. Res. Develop., vol. 2, no. 2, pp. 159165, Apr. 1958.
6. Edmundson, H.P. (1969) New Methods in Automatic Extracting. Journal of the ACM (JACM), 16, 264-285. https://doi.org/10.1145/321510.321519
7. Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez and Krys Kochut, “Text Summarization Techniques: A Brief Survey” International Journal of Advanced Computer Science and Applications(IJACSA), 8(10), 2017. http://dx.doi.org/10.14569/IJACSA.2017.081052
8. Leonhard Hennig, Winfried Umbrath, and Robert Wetzker. 2008. An ontology based approach to text summarization. In Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT’08. IEEE/WIC/ACM International Conference on, Vol. 3. IEEE, 291–294.
9. Ping Chen and Rakesh Verma. 2006. A query-based medical information summarization system using ontology knowledge. In Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on. IEEE, 37–42.
10. Elena Baralis, Luca Cagliero, Saima Jabeen, Alessandro Fiori, and Sajid Shah. 2013. Multi-document summarization based on the Yago ontology. Expert Systems with Applications 40, 17 (2013), 6976–6984.
11. Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web. ACM, 697–706.
12. Ganesan, Kavita & Zhai, ChengXiang & Han, Jiawei. (2010). Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference. 2.

13. Gupta, S., & Gupta, S. K. (2019). Abstractive summarization: An overview of the state of the art. Expert Systems with Applications, 121, 49–65. https://doi.org/10.1016/j.eswa.2018.12.011.

14. Lin, H. & Ng, V. (2019). Abstractive summarization: A survey of the state of the art. Paper presented at the The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19).

15. Okumura, N. & Miura, T. (2015). Automatic labelling of documents based on ontology. Paper presented at the 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM).

16. Sutskever, Ilya & Vinyals, Oriol & Le, Quoc. (2014). Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems. 4.

17. Hou, L., Hu, P. & Bei, C. (2017). Abstractive document summarization via neural model with joint attention. Paper presented at the Natural Language Processing and Chinese Computing, Dalian, China.

18. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

19. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana.

20. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. In CoRR, abs/1704.01444, 2017.

21. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota.

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

23. Xingxing Zhang, FuruWei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

24. Raffel, Colin & Shazeer, Noam & Roberts, Adam & Lee, Katherine & Narang, Sharan & Matena, Michael & Zhou, Yanqi & Li, Wei & Liu, Peter. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

25. Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter Liu. "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization." In *International Conference on Machine Learning*, pp. 11328-11339. PMLR, 2020.

26. Liu, Yang & Lapata, Mirella. (2019). Text Summarization with Pretrained Encoders.

27. Lewis, Mike & Liu, Yinhan & Goyal, Naman & Ghazvininejad, Marjan & Mohamed, Abdelrahman & Levy, Omer & Stoyanov, Veselin & Zettlemoyer, Luke. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. 7871-7880. 10.18653/v1/2020.acl-main.703.

28. S. Takase and S. Kiyono, “Rethinking perturbations in encoder-decoders for fast training,” in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5767–5780, Mexico City, Mexico, 2021.

29. X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, and T. Qin, “Rdrop: Regularized Dropout for Neural Networks,” 2021, https://arxiv.org/pdf/2106.14448.pdf.

30. Wang, S., Zhao, X., Li, B., Ge, B. & Tang, D. (2017). Integrating extractive and abstractive models for long text summarization. Paper presented at the 2017 IEEE International Congress on Big Data (BigData Congress).

31. Bhat, I. K., Mohd, M. & Hashmy, R. (2018). SumItUp: A hybrid single-document text summarizer. In M. Pant, K. Ray, T. K. Sharma, S. Rawat & A. Bandyopadhyay (Eds.),

Soft computing: Theories and applications: Proceedings of SoCTA 2016, Volume 1 (pp. 619–634). Singapore: Springer Singapore.

32. El-Kassas, Wafaa & Salama, Cherif & Rafea, Ahmed & Mohamed, Hoda. (2020). Automatic Text Summarization: A Comprehensive Survey. Expert Systems with Applications. 165. 113679. 10.1016/j.eswa.2020.113679.
33. SPARCK JONES, K. (1972), "A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL", Journal of Documentation, Vol. 28 No. 1, pp. 11-21. https://doi.org/10.1108/eb026526
34. Nallapati, Ramesh & Zhai, Feifei & Zhou, Bowen. (2016). SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. Proceedings of the AAAI Conference on Artificial Intelligence. 31. 10.1609/aaai.v31i1.10958.
35. Paulus, Romain & Xiong, Caiming & Socher, Richard. (2017). A Deep Reinforced Model for Abstractive Summarization.
36. Lin, Chin-Yew. (2004). ROUGE: A Package for Automatic Evaluation of summaries. Proceedings of the ACL Workshop: Text Summarization Braches Out 2004. 10.