Automatic Text Summarization Model using Seq2Seq Technique

Mrs. Chandrika Prasad
Dept. of Computer Science and Engineering
MS Ramaiah Institute of Technology
Bangalore
(Affiliated to Visvesvaraya Technical University, Belagavi, Karnataka)

Dr. Jagdish S. Kallimani
Dept. of Computer Science and Engineering
MS Ramaiah Institute of Technology
Bangalore
(Affiliated to Visvesvaraya Technical University, Belagavi, Karnataka)

Dr. Divakar Harekal
Dept. of Computer Science and Engineering
MS Ramaiah Institute of Technology
Bangalore
(Affiliated to Visvesvaraya Technical University, Belagavi, Karnataka)

Nicy Sharma
Dept. of Computer Science and Engineering
MS Ramaiah Institute of Technology
Bangalore
(Affiliated to Visvesvaraya Technical University, Belagavi, Karnataka)

***Abstract*— Increasing acquisition of digitization over the information storing and processing in our daily lives has increased the demand of digitization in multiple facets including in investigation processes as well. In fact, for crimes involving computer systems requires the adoption of best practices for the process of evidence extraction from acquired devices from the crime scenes.**

Over the past years, summarization has become a topic of research. Various techniques of Natural Language Processing (NLP) enabling researchers to generate efficient results for a wide spectrum of documents. In the proposed work Seq2Seq Architecture with RNN is used to perform summarization tasks for documents. The nature of the summary is abstractive and allows the generation of internal meaning by the model itself. With refinement and continual work, this model becomes a strong foundation to perform summarization on longer and legal documents. The results are efficient summary generation and ROUGE scores in the range of 0.6 - 0.7.

Keywords— Natural Language Processing, Text summarization, Machine Learning, Tensor Flow, Seq2seq

I. Introduction

The usual mode of human interaction and communication is known as Natural Language. Such text is available around us in multiple forms like email communications, web page content, SMS services etc. However, it must be noted ambiguity is inherent in human language, along with its continuous evolution. Despite being adept at details of speaking or understanding languages, performed sub-optimally at describing formal rules of the same. NLP involves "understanding" spoken language with enough efficiency, to generate intelligible responses to them. Therefore it is a two-way branch where naturally used language-based text will be part of the input and similar text results will be presented as output.

Today, our world is flooded with huge amounts of data. With such a big amount of data circulating in the digital space, there is a need to develop machine learning algorithms that can automatically shorten longer texts and deliver accurate summaries that can fluently pass the intended messages.

Judicial Systems are highly dependent on the manual and laborious summarization of individual cases. All such report generations take long hours and analyzing them to be used as a benchmark is very challenging and time-consuming. Machine Learning techniques are making this task possible without human intervention. With the right collection of data, DL neural network models can be trained and can be used to efficiently generate meaningful summaries of new unseen case studies. With the availability of such a module time, consumption on case studies both in fieldwork and for educational purposes will be highly reduced and can be invested in extracting other hidden meaningful interpretations.

The Proposed System uses DL models based on seq2seq architecture to generate abstractive summaries. It is an attention-mechanism model embedded within the encoder-decoder. The System works to generate abstractive and meaningful summaries out of legal documents and data. The measure of the resultant summary is with reference to ROUGE scores which is a non-differentiable metric.

II. Literature Survey

Gupta, Vanyaa, Neha Bansal, and Arun Sharma [1], Identifying the importance of summarization to ease the task of extraction of relevant information from the plethora of online resources automatic summarization is defined and legal summarization is given slight favor due to the vast scope of using extraction or abstraction in the field. Automatic Text Summarization can be generated in one of two forms i.e. Generic Summary (where no extra information is added)

Kanapala, Ambedkar, Sukomal Pal, and Rajendra Pamula [2], A survey was performed to reveal the progress in the summarization of both single and multi-document related to the legal domain. It was found that in general, the problem of summarization is slightly more challenging when handling legal text as the nature of text differs in size, structure (include status codes), vocabulary, ambiguity and citations. It is a prime concern to handle correctly the hierarchical structure of such text. Hence for the single document summary, the approaches were found and classified based on the overlying technique including Linguistic feature-based approaches like LSA (Latent Semantic Analysis) blending term and sentence description or using Lesk Algorithm, Statistical feature-based approaches where features like term frequency (TF), inverse sentence frequencies(ISF), textual entailment (TE) etc., Language-Independent Approaches where based on word frequency etc. sentence value is computed and then sentences are ranked. Besides these Evolutionary computing-based approaches, and Graph-based approaches like Tree Knapsack were also found.

For Multi Documents a similar study revealed several approaches wherein linguistic features important terms are extracted and a co-occurrence base for the terms is created. Another promising approach is the Latent Dirichlet Allocation which is used to score sentences based on certain topic words of prime importance. Even graph-based approaches have been studied as a viable option.

Jain, Aditya, Divij Bhatia, and Manish K. Thakur [3], Proposed Extraction based Summarization Using Word Vector Embedding. For every sentence, a similarity score is calculated to create the labelled training data using 100-dimensional glove vectors. For feature extraction Mean TF-ISF i.e. Term Frequency Inverse Sentence Frequency was evaluated, further sentence length was considered giving less weight to shorter sentences. For summarization MLP i.e. multi-layer perceptron, a three-layer fully connected feed-forward neural network was used. The results of the summarization with its ROUGE scores for the first 284 documents were found as ROUGE1: 0.36625, ROUGE2: 0.15735 and ROUGEL: 0.34410.

Yousefi-Azar, Mahmood, and Len Hamey [4], It was shown that a deep autoencoder (AE) can be suitably used for query-based summary generation based on extraction based on the term frequency (tf) input. Introduction of random noise to local tf additionally led to the proposal of Ensemble Noisy Auto-Encoder (ENAE). The training has been done in pre-training and fine-tuning stages. For parameter discovery, a generative model (RBM) is used. RBM consists of a neural network of two layers primary RBM being Gaussian-Bernoulli and latter as Bernoulli-Bernoulli (except for hidden units). The experimentation was carried out on SKE and BC3 email corpus and an 11.2% improvement in the ROUGE-2 recall was found by using AE.

Song, Shengli, Haitao Huang, and Tongxiao Ruan [5], Abstractive Text Summarization (ATS) is resolved by exploring the more minute semantic phrases within a document proposing an LSTM-CNN baes network. Experimental results are seen on the CNN and DailyMail datasets. It was found that the proposed LSTM-CNN network outperformed other models like LEAD, Moses, ABS and CoRe yielding higher ROUGE Scores. ROUGE-1 score is 34.9% and ROUGE-2 score is 17.8%.

Nema, Preksha, Mitesh Khapra, Anirban Laha, and Balaraman Ravindran [6], Simple attention and RNN based models provide promising results for shorter texts. However, to tackle the problem with a larger input size of documents an intra-attention model attending input and continuous output is proposed. For training supervised word prediction and reinforcement learning (RL) are combined. In conclusion, ROUGE-1 score is 42.94% and ROUGE-2 score is 26.02%.

Paulus, Romain, Caiming Xiong, and Richard Socher [7], Focusing on Query Based Summarization the problem of the encode-attend-decode returning repeated words were identified and hence an attention model (query) and another diversity-based model are proposed. In conclusion, it was found that diverse vectors could be generated during consecutive steps while attention was paid to words should a later requirement come up. An absolute gain of 28% was found in ROUGE-L score and state-of-the-art models were also outperformed.

Liao, Pengcheng, Chuang Zhang, Xiaojun Chen, and Xiaofei Zhou [8], Proposes an aggregation mechanism using the Transformer model to handle the same. This model can review information giving the encoder more memory capacity. The memory retention is achieved by adding the aggregation layer between the encoder and decoder. Aggregation mechanism rebuilds the final hidden states of the encoder using the history information. Rouge scores for aggregation model are as follows: ROUGE-1 score is 42.94% and ROUGE-2 score is 26.02%.

III. **Proposed Work**

The data is collected from Harvard NLP project which basically comprises of two datasets- Gigaword dataset and CNN/DM dataset. CNN/DM dataset have been used which comprises of news articles and its corresponding hand written summary. There are several different themes that the news articles are comprised of, for eg. Business, Politics, sports, finance.

The flowchart in figure 1 shows the sequence in which the data is imported, cleaned, preprocessed, trained and tested using DL neural networks such as seq2seq architecture.

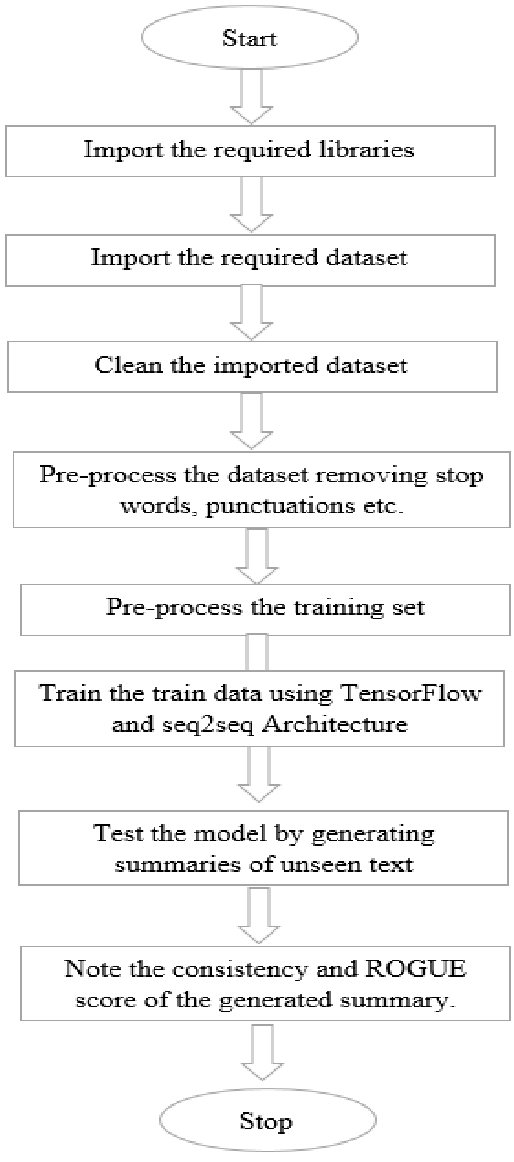

Fig.1. Flowchart for the training and testing of the dataset

The first step is to import the required libraries required by the python program to simplify some of the works. Then the next step is to import the dataset. The dataset is raw and needs preprocessing to extract only known characters and meaningful words while discarding unnecessary input. Dataset needs to be cleaned because the text data obtained is highly unstructured. The words may be shortened, spelling mistakes, not standardized, contains dates and other numerical information and legal jargon. Stemming or lemmatizing of words can also help at this step. The pre-processing step is dedicated exclusively for this purpose.

After the training and testing processes, note down the metrics that are used to analyze the performance of our model such as ROUGE Score.

Analyzing the limitations of the networks-- Since the training models are not perfect, there is a need to understand why they are not and find if any improvements can be done on these classifiers. This is represented in figure 2.

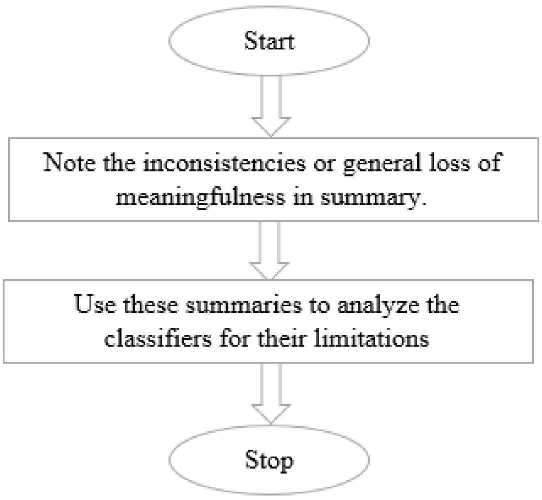

Fig.2. Analyzing the limitations of the classifier

The grammar inconsistencies and meaningless words are noted down from the trained dataset. This data is analyzed further to find any patterns or the limitations of the classifiers.

IV. Implementation

1. Data Cleaning

This module deals with the cleaning of the dataset, which removes noise from unstructured text. This module will be called while loading training data, test data and for viewing sample output of cleaning. The algorithm is presented below.

Input: A text dataset.
Output: A text dataset which is cleaned.
1. Import the required libraries.
2. Import the dataset.
3. For every sample in the dataset:
   a. Remove the unwanted characters and replace with #.

Any other substitution may be made if they seem fit.
4. Output the cleaned dataset.

Following this set dictionary building for our model will begin which will be a straightforward word dictionary containing (word, position) entries and a reverse dictionary containing (position, word) entries.

2. Building Dictionary and Data Preprocessing--

As mentioned above two dictionaries are generated, with the following steps. In terms of abstraction, the seq2seq algorithm necessitates these steps however for ETS this dictionary building is not required.

Here 4 built-in words are added before proceeding any further, they are <*padding*> which will be used to make sequences of the same length, <*unk*> to identify word not found in the dictionary, <*s*> identifies the beginning of a sentence, </*s*> marks the end of a sentence.

Input: A cleaned dataset
Output: A Document Term Matrix (Dictionary)
1. Import the labelled cleaned dataset.
2. Tokenize the sentences into the words forming it, with built-in nltk function.
3. Select the most common words only from the above tokens.
4. Add the 4 built-in words.
5. Iterate over the words to build a dictionary and store it by pickling.
6. Create a reverse dictionary using the above.
7. Define maximum summary and article lengths.

3. Building the seq2seq Abstractive Text Summarizer (ATS) model.

In the following section, a Recurring Neural Network (RNN) will be built, arranged as an Encoder/Decoder architecture called the seq2seq model. The seq2seq will be made as a bidirectional structure where the RNN cell will become an LSTM cell and will include an attention mechanism for better encoder/decoder interfacing and include a beam search concept. The building will be proceeded with in the form of the following blocks:
1. Initialization Block: Define the TensorFlow (tf), placeholders, variables and RNN cell.
2. Embedding Block: Embedding matrix is defined for further to be used in Encoder/Decoder.
3. Encoder Block: Multilayer bidirectional RNN is defined.
4. Decoder Block: Add the attention mechanism and the beam search.
5. Loss Block: Only for the training of ATS, to apply clipping to gradients, and use Adam Optimizer.

3.1 Initialization Block

Aim: Create Initialization Block.
1. Import the required libraries.
2. Build a Model Class which takes an object called args containing several parameters.
3. Initialize all the parameters like embedding size, num_hidden, num_layers, Learning Rate and BeamWidth.
4. Also reversed dictionary and maximum article, summary length parameters must also be initialized.
5. Define the testing phase to use LSTM as a cell.
6. Define batch size for data.
7. Define decoder using summary length.
8. Define global step beginning from zero.

3.2. Embedding Block

The algorithm to be followed is defined below:
Input: A dictionary
Output: Embedded Matrix

4. Else randomly define word2vector for testing
5. Define embeddings for both encoder and decoder.

3.3. Encoder Block:

The steps to be followed are:

1. Define forward and backward cells.
2. Connect them by using stack_bidirectional_dynamic_rnn with the parameters:
   Forward Cells, Backward Cells, Encoder embedded input (word2vec format), X_len (length of articles), time_major (if True avoids transposes).
3. Generate the output at encoder_output (used in attention calculation) and encoder_state (used in an initial state of the decoder).
4. To generate encoder_state both forward and backward using LSTMStateTuple must be combined.

3.4. Decoder Block:

The Decoder Block will be defined in two parts:

Training Part A: Attention Model

Input: Encoder Output (from the previous step) and Decoder Input (summary sentence in training)

1. The attention structure used here is BahdanauAttention.
2. Encoder_output Is used for attention calculation.
3. The decoder cell is defined as multilayer LSTM to which attention is added using AttentionWrapper.
4. Define a helper function to combine the two inputs.
5. Use all RNN outputs and get logits, by transposing decoder output.
6. Reshape the logits.

Part B: Applying BeamSearch

- The main goals of this phase are: dividing the encoder output, encoder states and x_len such that beam search can be applied.
- Build a input independent decoder, as testing phase does not include input summary.

3.5. Loss Block

The training is performed in this block through loss calculation, gradient calculations ad clipping, and optimizer application. The steps to be followed have been enlisted below.

1. Define in variable scope a variable for embedding.
2. If in the training phase and args.glove is enabled use tf.constant as no change is required.
3. get_init will be used to return a vector for each word.

1. Define the name scope and define a block to be used only while training.
2. Perform loss calculation (softmax function).
3. Calculate the gradients.
4. Apply clipping on gradients to solve exploding gradient error.
5. Apply Adam Optimizer. (previously defined learning rate is used)

V. Results and Discussion

Results obtained for the proposed system--The metrics have been observed with our proposed method for both seq2seq and text-rank. The trend in which they are changing is investigated. Starting to evaluate from the beginning, first, a dataset is loaded so that it can be prepared for further processing and looks as shown in figure 3.

With this clean dataset, dictionary building takes place required for mapping of words to their corresponding integer values and vice-versa.

Now perform numerical analysis on the articles as well as their corresponding summaries. It is observed that 99 percentile articles have a length of 28 and summaries of length 11.

After training, five random inputs are selected from our testing dataset and generate their summary. The generated summary is compared with a human generated summary.

Article1: "general motors corp. said Wednesday it' sales fell ##.# percent in December and four percent in #### with the biggest losses coming from passenger car sales ."

> Model output: gm us sales down # percent in december

> Actual title: gm december sales fall # percent

```
In [19]: # Inspecting some of the reviews
         for i in range(5):
             print("Article #",i+1)
             print(reviews.Summary[i])
             print(reviews.Text[i])
             print()
```

```
Article # 1
australian current account deficit narrows sharply

australia 's current account deficit shrunk by a record #.## billion dollars -lrb- #.## billion us -rrb- in the jun
e quarter due to soaring commodity prices , figures released monday showed .


Article # 2
at least two dead in southern philippines blast

at least two people were killed in a suspected bomb attack on a passenger bus in the strife-torn southern philippin
es on monday , the military said .


Article # 3
australian stocks close down #.# percent

australian shares closed down #.# percent monday following a weak lead from the united states and lower commodity p
rices , dealers said .
```

Fig.3. Inspecting some reviews

```
Summaries:
              counts
count    2225.000000
mean        7.737528
std         1.709674
min         3.000000
25%         7.000000
50%         8.000000
75%         9.000000
max        13.000000

Texts:
              counts
count    2225.000000
mean       20.443596
std         3.040285
min         6.000000
25%        19.000000
50%        21.000000
75%        22.000000
max        34.000000
```

Fig.4. Numerical Analysis of Articles and summaries

Article2: "japanese share prices rose #.## percent thursday to <unk> highest closing high for more than five years as fresh gains on wall street fanned upbeat investor

the sentiment , dealers said ."

> Model output: Tokyo shares close # percent higher

> Actual title: Tokyo shares close up # percent

Article3: "hong kong share prices opened #.## percent higher thursday on follow-through interest in properties after wednesday 's sharp gains on abating interest rate worries, dealers said."

> Model output: Hong Kong shares open higher

> Actual title: Hong Kong shares open higher as rate worries ease

Article4: "the dollar regained some lost ground in Asian trade Thursday in what was seen as a largely technical rebound after weakness prompted by expectations of a shift in us interest rate policy, dealers said."

> Model output: dollar stable in Asian trade

> Actual title: dollar regains ground in Asian trade

Article5: "the final results of Iraq's december general elections are due within the next four days, a member of the Iraqi electoral commission said on thursday."

> Model output: Iraqi election results due in the next four days

> Actual title: Iraqi election final results out within four days

Figure 5 shows the result of how the metrics vary when checked it with the same five random articles from the validation dataset using seq2seq.

Article #	Rouge-1			Rouge-2			Rouge-l		
	f	p	r	f	p	r	f	p	r
1	0.71	0.63	0.83	0.16	0.15	0.2	0.57	0.50	0.67
2	0.83	0.83	0.83	0.59	0.6	0.6	0.83	0.83	0.83
3	0.71	1.0	0.56	0.67	1.0	0.5	0.71	1.0	0.56
4	0.72	0.8	0.67	0.44	0.5	0.4	0.73	0.8	0.67
5	0.62	0.63	0.63	0.29	0.29	0.29	0.63	0.63	0.63

Fig.5. Rouge Score for seq2seq

Similarly, results for extractive summary for the Hindi language are generated. During the prediction, needed to provide the percent by which wanted to shorten our summary or in other words the percentage of summary wanted of the original text. The predicted summary is stored in a file and for compression rate of 25%, 50% and 75% following results are obtained in figure 6,7,8 and 9.

VI. Conclusion and Future Enhancements

Conclusion

In this work, an abstractive summarization platform is attempted to create wherein, tried to incorporate a

प्रधानमंत्री नरेंद्र मोदी के चीन दौरे के बाद एक खुशखबरी आई है।
भारत और चीन के बीच नाथुला सीमा से मंगलवार को द्विपक्षीय व्यापार शुरू हो गया है।
बीते साल दोकलम विवाद की वजह से यह व्यापार बंद हो गया था।
इस मौके पर दोनों देशों के व्यापारियों और अधिकारियों ने एक दूसरे को गिफ्ट और बधाइयां देकर जश्न मनाया।
हालांकि यह एक अनौपचारिक मुलाकात थी और इस दौरान कोई लिखित समझौता नहीं हुआ था।
व्यापारियों ने उम्मीद जताई कि इस साल भारत और चीन के बीच किसी भी तरह की समस्या नहीं आएगी और व्यापार जारी रहेगा।

Fig.6. Hindi news article

प्रधानमंत्री नरेंद्र मोदी के चीन दौरे के बाद एक खुशखबरी आई है
भारत और चीन के बीच नाथुला सीमा से मंगलवार को द्विपक्षीय व्यापार शुरू हो गया है
बीते साल दोकलम विवाद की वजह से यह व्यापार बंद हो गया थाव्यापारियों ने उम्मीद जताई कि इस साल भारत और चीन के बीच किसी भी तरह की समस्या नहीं आएगी और व्यापार जारी रहेगा

Fig.7. Summary at 25% compression

प्रधानमंत्री नरेंद्र मोदी के चीन दौरे के बाद एक खुशखबरी आई है
बीते साल दोकलम विवाद की वजह से यह व्यापार बंद हो गया थाव्यापारियों ने उम्मीद जताई कि इस साल भारत और चीन के बीच किसी भी तरह की समस्या नहीं आएगी और व्यापार जारी रहेगा

Fig.8. Summary at 50%

व्यापारियों ने उम्मीद जताई कि इस साल भारत और चीन के बीच किसी भी तरह की समस्या नहीं आएगी और व्यापार जारी रहेगा

Fig.9. Summary at 75%

scope for the usage of Legal or Judicial Data.
Such proceedings involve sensitive information and due to unavailability of public data on the same, chose to go with data of similar nature i.e. collected from the news.

Besides forming the abstraction, an extractive module is also included, which allowed us to see the comparison between the two techniques. The seq2seq architecture is followed, and as a further attempt, different models and architectures can be chosen.

The summarization results are also incorporated on Hindi Language and news articles by following the same procedure of word embeddings followed by encoder-decoder with an attention mechanism.

Future Enhancements

Having worked only with seq2seq architecture, whereas, pointer generator network is another promising architecture, being used for summarization. Further with the availability of more data, the network train can be streamlined on different data and generate domain-specific summaries.

REFERENCES

[1] Gupta, Vanyaa, Neha Bansal, and Arun Sharma. "Text summarization for big data: A comprehensive survey." In International Conference on Innovative Computing and Communications, pp. 503-516. Springer, Singapore, 2019.

[2] Kanapala, Ambedkar, Sukomal Pal, and Rajendra Pamula. "Text summarization from legal documents: a survey." Artificial Intelligence Review 51, no. 3 (2019): 371-402.

[3] Jain, Aditya, Divij Bhatia, and Manish K. Thakur. "Extractive text summarization using word vector embedding." In 2017 International Conference on Machine Learning and Data Science (MLDS), pp. 51-55. IEEE, 2017.

[4] Yousefi-Azar, Mahmood, and Len Hamey. "Text summarization using unsupervised deep learning." Expert Systems with Applications 68 (2017): 93-105.

[5] Song, Shengli, Haitao Huang, and Tongxiao Ruan. "Abstractive text summarization using LSTM-CNN based deep learning." Multimedia Tools and Applications 78, no. 1 (2019): 857-875.

[6] Nema, Preksha, Mitesh Khapra, Anirban Laha, and Balaraman Ravindran. "Diversity driven attention model for query-based abstractive summarization." arXiv preprint arXiv:1704.08300 (2017).

[7] Paulus, Romain, Caiming Xiong, and Richard Socher. "A deep reinforced model for abstractive summarization." arXiv preprint arXiv:1705.04304 (2017).

[8] Liao, Pengcheng, Chuang Zhang, Xiaojun Chen, and Xiaofei Zhou. "Improving Abstractive Text Summarization with History Aggregation." arXiv preprint arXiv:1912.11046 (2019).

[9] Chopra, Sumit, Michael Auli, and Alexander M. Rush. "Abstractive sentence summarization with attentive recurrent neural networks." In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93-98. 2016..

[10] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[11] Zhang, Yizhe, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. "Deconvolutional paragraph representation learning." In Advances in Neural Information Processing Systems, pp. 4169-4179. 2017.

[12] Cohan, Arman, and Nazli Goharian. "Revisiting summarization evaluation for scientific articles." arXiv preprint arXiv:1604.00400 (2016).

[13Gupta, Vanyaa, Neha Bansal, and Arun Sharma. "Text summarization for big data: A comprehensive survey." In International Conference on Innovative Computing and Communications, pp. 503-516. Springer, Singapore, 2019.

[14] Ferilli, Stefano, and Andrea Pazienza. "An abstract argumentation-based approach to automatic extractive text summarization." In Italian Research Conference on Digital Libraries, pp. 57-68. Springer, Cham, 2018.

[15] Sehgal, Sunchit, Badal Kumar, Lakshay Rampal, and
Ankit Chaliya. "A modification to graph based approach for extraction based automatic text summarization." In Progress in advanced computing and intelligent engineering, pp. 373-378. Springer, Singapore, 2018.

[16] Nagwani, N. K. "Summarizing large text collection using topic modeling and clustering based on MapReduce framework." Journal of Big Data 2, no. 1 (2015): 6.

[17] Khan, Atif, Naomie Salim, Haleem Farman, Murad Khan, Bilal Jan, Awais Ahmad, Imran Ahmed, and Anand Paul. "Abstractive text summarization based on improved semantic graph approach." International Journal of Parallel Programming 46, no. 5 (2018): 992-1016.

[18] Sinha, Aakash, Abhishek Yadav, and Akshay Gahlot. "Extractive text summarization using neural networks." arXiv preprint arXiv:1802.10137 (2018).

[19] Lopyrev, Konstantin. "Generating news headlines with recurrent neural networks." arXiv preprint arXiv:1512.01712 (2015).

[20] Colmenares, Carlos A., Marina Litvak, Amin Mantrach, and Fabrizio Silvestri. "Heads: Headline generation as sequence prediction using an abstract feature-rich space." (2015).

[21] Zhu, Xiaodan, Gerald Penn, and Frank Rudzicz. "Summarizing multiple spoken documents: finding evidence from untranscribed audio." In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pp. 549-557. Association for Computational Lin