TEXT SUMMARIZATION: EXTRACTIVE AND ABSTRACTIVE APPROACHES

Kannav Dhawan

Department of Electrical and Computer Engineering
University of Waterloo, Canada

Abstract- **A significant development in the technology and the continuous availability of the online resources resulted in the huge aggregation of information in the form of natural language on the internet. Availability of the large corpus of textual information on the web from the variety of sources carrying valuable information needs to be effectively summarized for the efficient use. The automatic text summarization plays a big role for summarizing the large and explosive corpus of documents into the meaningful text without human intervention exclusive of any bias. Two known state-of-the-art techniques for the task are extractive and abstractive summarization. The extractive summarization is the task of extracting the most important excerpts from the long text without any rephrasing and the abstractive summarization interprets the text employing linguistic methodologies with the inclusion of new vocabulary in the generated summaries. The approach used for extractive summarization takes the input text, uses the word frequency as the feature for baseline and extended features like sentence length, intersectional n-grams for the improved system and then the summarization is performed using priority queue algorithm for the baseline and using rule-based algorithm for the extended features. Further, the TextRank based summarization is implemented followed by the ensemble model combining the improved model, TextRank model and the summa summarizer with variations on the similarity function of TextRank. For the abstractive summarization, the T-5 transformer and LSTM architecture using GloVe and FastText embeddings are implemented. Polarity based summary classification and the end to end online summarizer is developed using the best extractive model. Models are evaluated using the baseline precision metric for initial evaluation, manual linguistic quality evaluation and the Rouge based evaluation. For the extractive summarization, the implemented ensemble model outperformed the other models on precision whereas for the abstractive summarization, the baseline T-5 transformer outperformed the other implemented LSTM architectures on ROUGE scores.**

Keywords—Text summarization, Abstractive, Extractive, Text Rank, Transformer, LSTM, Rouge

1. INTRODUCTION

The amount of information and the natural language text available online has increased drastically which tends to have increased the need to find the relevant information with ease. Such natural language text generally consists of the long documents which makes it difficult for the individual to read and find the relevance of the document. In order to find the relevant information, user needs to go through the long documents and make an inference for the relevance of the document for the task. Thus, the process becomes time consuming, Automatic text summarization by generating the shorter, meaningful, coherent and concise version of text makes it feasible to understand the long documents in a shorter span of time. The problem of coherency in the generated summaries is still researchable and needs to be addressed. The problem of text summarization is divided into two subproblems viz. single document and multi document text summarization. Single document text summarization tends to make a summary from the single piece of text and hence repetition is not considered as a major issue whereas the multi document text summarization takes multiple documents with the same topic into consideration and generates the summary using both the texts which may lead to the problem of incoherent and repeated information.

There are two approaches for the text summarization viz. Extractive summarization and the Abstractive summarization. The Extractive text summarization also referred as the sentence scoring approach basically generates the summaries which are the subsets of the original text. These subsets are generated employing various features viz. word frequency, sentence length and ranking the text based on these features. These summaries may end up truncating the relevant information from the original text if every sentence in the original text is equally important. Some techniques used for Extractive summarization are sentence scoring and priority queue-based summarization, TextRank based summarization. Abstractive summarization[1] linguistically understands the text and generates the summarized version of the text with new sentences. Transformer based summarization and the sequence to sequence modeling can be considered as the abstractive approaches for text summarization.

In this paper, we develop the extractive summarization system using word frequency as the feature for sentence scoring and then scale the system with additional features viz. sentence length and intersectional n-grams. Further the sentence selection using the scores is done by the priority queue algorithm and rule-based selection criterion based on the features. TextRank summarization using Gensim[2] and the ensemble model with the variation of the similarity function is implemented using the summa summarizer[3]. Further, the text to text transfer transformer is used to leverage the power of abstractive summarization[4]. A more fine-tuned LSTM based automatic text summarization is employed for the abstractive summarization using GloVe and FastText embeddings. Many abstractive summarizations may end up changing the polarity and adding a bias while generating the summary due to the addition of new vocabulary in the summary. Average percentage change in the polarity between the real and the generated summaries is considered for evaluating the bias. Finally, an end to end automatic text summarization system is created to generate the summaries for the text available on the web pages. Different evaluation metrics are used for the evaluating the effectiveness of the systems in both the extractive and abstractive summarization techniques. Extractive models are evaluated using the baseline metric of precision based on the overlapping unigrams. The ROUGE i.e. Recall-oriented understudy for Gisting evaluation[24] is used for evaluating both the approaches.

This paper has five sections. Section II discusses the Literature Review; Section III demonstrates our different techniques for both the approaches viz. extractive and abstractive summarization. Section IV describes the in-depth experimental setup, results obtained and the evaluations. Section V suggests the conclusion and the future work.

2. LITERATURE REVIEW

2.1. Extractive Summarization

The task of text summarization is being researched since a long period of time. There have been significant advancements in both the extractive and abstractive approaches. A lot of research is being done to generate better summaries comparable to the human generated ones without any addition of the significant bias towards the topic.

Deshpande et al. [5] presented an extraction based multiple document text summarization method in which sentence clusters are constructed by combining multiple documents for each query followed by the sentence scoring and cosine similarity calculation and finally picking up top scoring sentences from each cluster arranged in decreasing order of group score resulting in better f1 score, recall and precision than using just document clustering or using summarization based on statistical features. Cui et al. [6] proposed a document clustering algorithm named Particle swarm optimization[2] which executes a global search in solution set instead of local search as in K-means algorithm for clustering of documents which helps avoiding local minima and generating average distance (ADDC) of Euclidean and cosine distance between the cluster and centroid of the cluster where the document belongs to. This in-turn acts as the root for the task of automatic text summarization.

A reformed differential evolution algorithm having chromosomes as integers with introduction of cosine similarity and fitness function generating high density of similar sentences within the cluster and low density among other clusters is presented by Karwa et al in [7]. It is repeatedly used for selection of next generation chromosomes until adequate value of the function is reached signaling generation of competent summary resulting in f1 score, recall and precision of 0.44, 0.74 and 0.32 respectively with cosine similarity and 0.42,0.57 and 0.34 respectively with normalized google distance method.

Babar and patil in [8], employed a unique text extraction algorithm where document is preprocessed, features are obtained as 8 vector characteristics signifying presence of title in sentence, discarding brief sentences, location in the paragraph, presence of numerical information, thematic words, term weight, proper nouns and similarity between the sentences followed by score computation which is then fed to fuzzy inference system and a summary is formed. Another summary is formed using semantic analysis followed by term matrix generation and singular value decomposition which is combined with the first summary to give f1 score, recall and precision of 0.67, 0.44 and 0.90 respectively for a dataset when tested on ten different datasets. Scores were calculated on unigrams. Salim et al.[9] presented an abstractive summarization model which at first separates the document into sentences having the document index and location in the corresponding document. Semantic role labelling is applied to designate semantic term viz. core arguments and adjunctive arguments to protect the essential nuance of the sentence. It is followed by application of semantic similarity matrix which calculates similarities of the constructed argument structure using Jian semantic similarity metric defined in[10]. semantic clustering is performed for closely associated predicate arguments. Finally, some predicate arguments are chosen depending on the feature score and application of genetic algorithm which are then fed into SimpleNLG realisation engine[11] resulting in formation of effective summary.

Co-rank model for extractive text summarization proposed by Fang et al. [12] links graph based algorithm, PageRank[13] with the value of words which works by first constructing an undirected graph, calculating the score for each sentence in the document, revising weighted words depending on corresponding sentences followed by recalculating weighted sentences based on the weighted words to complete the loop. This forms an effective scoring mechanism to generate the summary achieving ROUGE-1, ROUGE-2 and ROUGE-L scores of 0.697, 0.606 and 0.661 respectively when applied on the DUC02 Dataset. Semeraro et al. [14] presented a modified centroid based text summarization system in which a vector of real numbers is formed for every constituting word, then a lookup table is constructed where each row corresponds to the word in vocabulary, followed by preprocessing of the document. Then centroid embedding is constructed using the lookup table and finally summary is generated by choosing top sentences out of scored sentences formed with the help of look up table and cosine similarity of the embeddings. This approach resulted in ROUGE-1, ROUGE-2 score of 0.38 and 0.09 which is quite good taking in mind the least complexity of the model when compared with the standard models.

2.2 Abstractive summarization

A neural attention model for abstractive text summarization was proposed by Rush et al.[15] which first forms the neural network language model (NNLM) [16] having word embedding matrix and weight matrices, then an adumbration is created depending upon context using attention based contextual encoder, followed by training the model on random input and output combination and finally summary is generated using beam-search decoder giving ROUGE-1, ROUGE-2 and ROUGE-L score of 0.2818, 0.0849 and 0.2381 respectively when implemented on DUC04 dataset which is excellent for an abstraction based summary.

RNN model with attention mechanism by Xiang et al. [17] outperforms the neural attention model for text summarization proposed by Rush et al [16]. 100 dimensional pretrained Word2Vec word embeddings are employed with trainable parameters set to True. Encoder comprises of bidirectional GRU-RNN[18] with hidden layer having dimension of 200 and decoder comprises of single GRU-RNN. Attention structure based hidden states are used and softmax layer is employed on resultant Large target vocabulary[19] for producing words, followed by beam-search decoder which produced the resultant summary. Maximum ROUGE-1, ROUGE-2 and ROUGE-L score of 0.3525, 0.1798 and 0.3318 are achieved respectively on Gigaword dataset.

Gao et al. [20] proposed a text summarization method which combines the effectiveness of both extraction and

abstraction-based summarization. Initially it determines the keywords by employing TextRank algorithm which are then individually fed to the key-information guide network to generate a key comprising of most recent forward and backward hidden state. It is then fed as an additional dynamic trainable input to the attention structure along with hidden states of the encoder making it consider keys as an important parameter. Keys are utilized to compute probability distribution for the vocabulary and a pointer mechanism can be employed which allows to choose word from either vocabulary or from input document itself followed by training. For inference, apart from using above structure, a prediction guide structure is used which is a feed forward network having single layer with activation function of sigmoid predicting the amount of key based knowledge in the summary. It results in ROUGE-1, ROUGE-2 and ROUGE-L score of 0.39, 0.17 and 0.36 respectively when implemented on Amazon Reviews Dataset with a larger test set.

Barrios et al. [21] introduces substitute functions to the normal similarity function utilized by Text Rank algorithm for the text summarization task. The first one being Longest Common Substring which finds the longest matching collection of words acting as similarity between them. The second one is Cosine distance which uses cosine similarity as notion of similarity. Next substitute is BM25 which is in fact a modification of existing TF-IDF in terms of using notion of probability. Summary generated has ROUGE-1, ROUGE-2 and ROUGE-SU4 score of 0.4042,0.1831 and 0.2018 which is in fact a progress of 2.92% from baseline Text Rank algorithm.

2.3 Transformer based summarization

Pretrained BERT employed by Devlin et al.[22] for embeddings on humongous dataset improved the extractive text summarization with respect to coherence between the sentences. Yang Liu [23] presented a different variety of BERT namely- BERTSUM to be utilized for extractive text summarization. First sentences are encoded by inserting tokens at either end of the sentence to mark its start and the end. For separating numerous sentences in a text, interval segment embeddings are utilized followed by improving the results by stacking up layers on it using binary crossentropy loss. Plain classifier with only one stacked up layer on the result of BERT along with application of sigmoid activation function gave ROUGE-1,ROUGE-2 and ROUGE-L score of 0.4323, 0.2022, 0.3960 whereas BERT result followed by normal Transformer layers gave ROUGE-1,ROUGE-2 and ROUGE-L score of 0.4325, 0.2024, 0.3963 and finally BERT result followed by LSTM layers gave ROUGE-1,ROUGE-2 and ROUGE-L score of 0.4322, 0.2017 and 0.3959 respectively outperforming statistical models.

3. APPROACH

After the brief analysis of the methods and approaches discussed in the previous section, two pipelines were constituted consisting of the extractive and abstractive approaches for text summarization.

3.1. Dataset

The dataset used for the summarization is ‘BBC news corpus’ and ‘Amazon Food Reviews’. ‘BBC News corpus’ consists of articles in 5 different topics viz. business, entertainment, politics, sports and tech. The gold summaries are available with the dataset which are human generated. All the articles and summaries in each section inclusively are of different length. ‘Amazon Food Reviews dataset’ consists of the 568454 reviews, for which the summaries are also provided. We have used the entire datasets for the task and the overall evaluation of the systems was performed on the whole dataset to introduce generality of the approaches followed. Whereas the detailed qualitative and quantitative analysis was performed on sample articles. The ‘BBC News corpus’ was used for the extractive summarization whereas the ‘Amazon Food Reviews dataset’ was used for the abstractive approaches considering the training time and model complexity. Moreover, Wikipedia articles were also fetched for the end to end summarization using the best extractive system. Figure 1 shows the data distribution among the topics and the article length in the ‘BBC corpus’. It can be seen that the mean article length is 2000 words approximately with the article length ranging between 1000 to 5000 words per article for all the categories with few anomalies having higher number of words. In order to have the better evaluations, the anomalies in the dataset were removed.

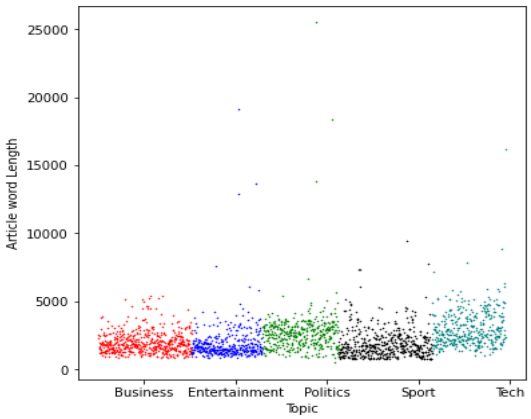

Fig. 1. Article length (BBC News Corpus)

3.2 Preprocessing

For the initial preprocessing, the documents from the BBC news corpus, Amazon food reviews are cleaned for the unwanted tags and the documents fetched from web were parsed using Beautiful soup library and the text was extracted using all the instances of the <p> tags. The unparsed article is represented in Figure 2.

```
</p><p>Increasing transistor count in
<a href="/wiki/Digital_electronics"
title="Digital electronics">digital
electronics</a> provided more process
power that enabled the development of
```

Fig 2. Sample Text

Further, for all the input documents, the text was converted into lowercase, all the text inside the parenthesis was removed, basic preprocessing of removal of punctuations, non-ASCII characters was done. For the Extractive approaches, the removal of the stopwords was done for the implementation of the sentence scoring. If stopwords are not excluded for scoring, the sentences with more stopwords may end up achieving the high importance scores and hence effecting the quality of the extractive system. Stopwords were included at the time of summary generation.

For abstractive summarization using LSTM, Word index dictionary was formed using keras "fit_on_texts" followed by the text sequencing where the input text is converted into corresponding index from the word index dictionary and the sequences are truncated and padded given the fixed length.

3.3 Extractive Summarization

For the extractive summarization, we developed four systems to compare the performance of the approach. There exist multiple methods for selecting the important sentences, sentence scoring is one of them which further includes multiple hand-crafted feature generation techniques for creating the scoring system.

Initially, we developed a system where the preprocessed input document is tokenized by sentences followed by the word tokenization for each sentence, which further is used to create different features for sentence scoring. For the baseline, a single feature with word frequency was generated where a dictionary with different words in the vocabulary of that article and their counts was created. The values were normalized by the division with the maximum count. Further, a sentence dictionary carrying sentence scores was created where the value suggests the total frequency count of the word impressions in the sentence summed from the dictionary carrying word frequencies. Thus, the top n sentences are selected using the priority queue algorithm which selects the sentences based on the sentence scores generated.

Further in our second system , some additional features viz. sentence length and the intersectional n-grams were generated to control the length of the sentences in the summary and the redundancy in the summaries generated.

1. Sentence Length

A new feature with the sentence length is generated where the count of words in a tokenized sentence is divided by the length of the longest sentence in the document. The feature score is given by the formula:

$$Feature(Si) = \frac{word\ count\ in\ ith\ sentence.}{maximum\ word\ count\ in\ longest\ sentence.} \quad (1)$$

This feature for scoring was selected to control the effect of extremely long and short sentences appearing out to be important.

2. Intersectional n-grams

A novel intersectional n-grams feature determines the sentences which may have the redundant information in other sentences. We have considered bigrams for developing a more informed feature not neglecting the important information which may be possible by selecting unigrams. The calculation for the same is performed as follows:

$$\text{Feature}(S_i) = \text{Sum of Bigram similarity count for each } i^{th} \text{ sentence with all the other sentences.} \quad (2)$$

This can help in dropping one of the sentences with high scores and hence preventing the inclusion of redundant information in the generated summary.

After the generation of the three features viz. word frequency, sentence length and the intersectional n-grams, the selection of the top n sentences was carried out based on the algorithm Shown in figure 3.

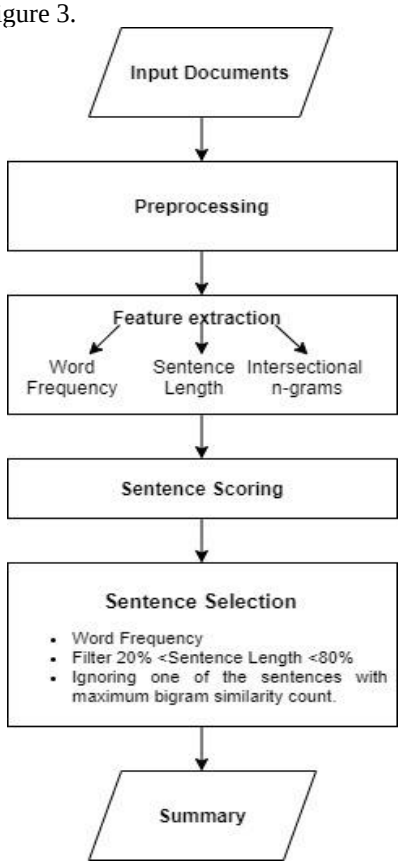

Fig. 3. System 2 with sentence length and intersectional n-grams for features

It suggests that after the scoring and feature extraction is performed, each sentence carries three features which can be utilized for sentence selection. We select the sentences by giving the highest importance to the word frequency followed by the sentence length where we ignore all the sentences having the word count score of less than 0.20 and more than 0.80 to prevent extremely long and short sentences included in the summary. We ignore one of the sentences having the maximum sum count of bigrams which helps in eliminating the sentences where the redundancy can be seen and hence improving the quality of the summaries.

In our third extractive summarization system, we implemented TextRank, a graph-based algorithm which categorizes the sentences into nodes where each edge connecting the nodes represents the weights defining the similarity between the nodes i.e. sentences which in-turn determines the importance of the sentences that may be used for the sentence selection[2]. Features used for defining the weights for edges include the 'percentage of the word overlapping in any two sentences', longest common substring and the cosine similarity calculated on the TF-IDF of the sentences within the document. Gensim's implementation of the TextRank was used for the task where the preprocessed input documents from the BBC corpus were fed into the system and the summaries were generated.

We created an ensemble technique for our fourth system, in which we combine the output summaries of our second system where we used the intersectional n-grams and

sentence length features along with the baseline word frequency, TextRank based summarizer and the summa summarizer[3] which is build combining several features viz. Position scoring, centroidal cosine similarity etc. We generate the summaries at the compression ratio of 0.4 for each of these underlying systems and select the intersectional sentences from all the summaries. Figure 4 depicts the ensemble system.

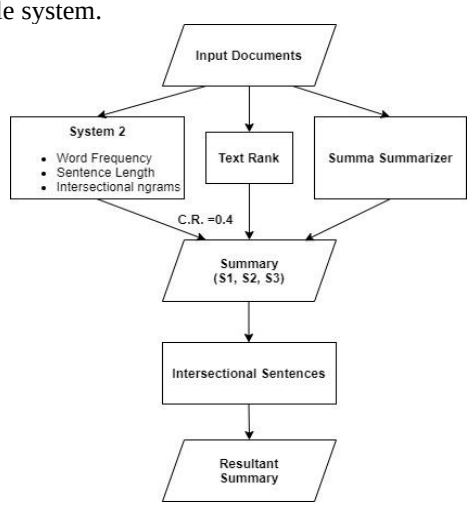

Fig. 4. Ensemble System

3.4. Abstractive Summarization

The dataset utilized for realization of the abstractive text summarization task by our model is Amazon Reviews dataset which encompasses the customer reviews for variety of products purchased and their corresponding precise and compact summaries. We are using text to text transfer transformer[4] as our baseline. Raffel et al. implemented the novel T-5 transformer architecture where the positional embeddings are formed which implies that a single value is appended to corresponding logit for calculating weights of the attention structure. Embedding parameters are shared across the layers of our model while at the same time differing position embedding is used by the attention heads in a given layer. These embedding sequences are fed as an input to the transformer model which has encoder-decoder mechanism where encoder has the normalized attention layer followed by the Neural network. Decoding is autoregressive in nature which means that it cares only about previously generated outputs, which is followed by the addition of fully connected output layer with softmax activation function. Finally, model is trained using various hyperparameters including dropouts to avoid overfitting.

Further two LSTM models with different embeddings viz. GloVe and FastText having separate embedding matrices for encoder and decoder are implemented. 300- dimensional GloVe and FastText embeddings are utilized as learning model to transform words into their respective vector representations. These are the unsupervised techniques used for creation of embeddings and hence embedding matrices which are separately utilized in the embedding layer for both encoder and decoder architecture of our sequence to sequence model. Instead of using a single embedding matrix for both encoder and decoder component of our LSTM, we decided to use separate embedding matrix for both to counter the fact that review and its summary have differing amount of words and hence different lengths of vocabulary to be utilized, which in turn increases the performance of our model in terms of space and time complexity.

We first use keras tokenizer to form the word index dictionary while at the same time cleaning the textual data for both our reviews and summaries, which is followed by the generation of sequenced data by replacing the input text with the corresponding values form the word index dictionary and pad sequencing the data individually for both.

While generating the embedding matrix, out of vocabulary words when encountered are replaced by special <UNK> symbol. <TOS> symbol is appended to all the sentences in reviews as well as summaries to signify its end i.e. termination of the sentence. <PAD> symbol is used to signify the padding value that will be used at the end of each vector representation in order to have all the input sequences with a definite length. The values used for the <UNK>, <TOS> and <PAD> symbols in our model are chosen to be 1,2 and 0 respectively. Intuitively we have used a special symbol <SOD> with a value of 3 for the first step in decoder mechanism to transfer a signal for starting the decoding process. We have fixed maximum vocabulary size for review and the summary to be 40000 and 15000 to have an efficient memory utilizing system. While generating the output of the decoder, we need to store tensor of size given by the equation:

$$tensor_{size} = X_{size} \times t \times V \tag{3}$$

Where X is the training data size, t is the number of tokens(length of the current summary) and V is the generated vocabulary size. We will then arrange the data in form of batches ready to be utilized in the LSTM model.

We have chosen LSTM because of its ability to apprehend longstanding dependency while avoiding the problem of vanishing gradient to great extent for the given task.

For the sequence to sequence model utilization for the task of text summarization, encoder-decoder architecture is used in which the encoder is responsible for generating the fixed length vectors carrying contextual information for the input sequence when fed with one word to the encoder at each timestamp. Whereas, the decoder component which receives the encoder output, decodes the sequence and makes the predictions for the next word at each timestep. This decoder is initialized by using the hidden and cell state of the concluding time step of the encoder component. Figure 5 depicts the LSTM"s encoder-decoder architecture where the left branch represents the encoder component and the right branch represents the decoder component.

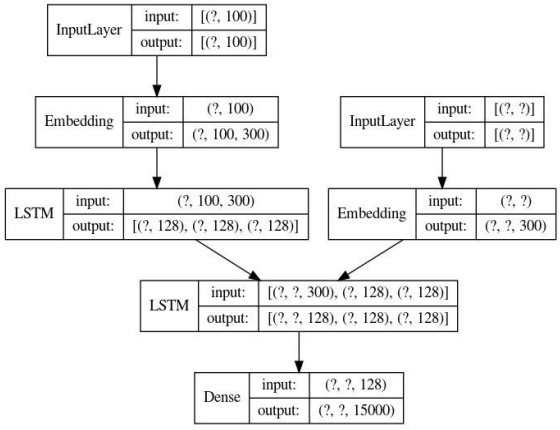

Fig. 5. LSTM Architecture

3.5. Evaluation metrics

Initially, a baseline metric of precision for extractive summarization was setup to evaluate the quality of the summary generated. The calculation is performed as

$$Precision(r,h) = \frac{O(r,h)}{c} \quad (4)$$

Where the r is the reference summary, h is the generated hypothetical summary, c represents the word count in the hypothetical summary and the O(r, h) calculates the unigram overlapping between the reference and the hypothetical summary.

For the both the extractive and abstractive summarization, we used the Rouge scores for the quantitative analysis of the summaries generated and for the qualitative analysis, summaries were evaluated making comparisons from the individual's perspective. ROUGE i.e. recall-oriented understudy for gisting evaluation[24] is a metric which evaluates the system's performance by comparing the summary generated by the model with the reference summary. Using this metric, we calculate the precision, recall and F-scores for unigrams, bigrams and the longest common subsequence. The precision, recall and F-score for the n-grams is calculated as

$$Precision = \frac{O_{grams}(r,h)}{gc} \quad (5)$$

$$Recall = \frac{O_{grams}(r,h)}{rc} \quad (6)$$

$$F-Score = \frac{2*(Precision)*(Recall)}{Precision+Recall} \quad (7)$$

Where the $O_{grams}(r, h)$ represents the n-gram overlapping of the words in reference summary and the hypothetical generated summary. Word count in the generated summary is represented by "gc" whereas the word count in the reference summary is given by "rc".

Precision determines how precisely the system generates the summary by finding the overlapping n-grams divided by the total word count of generated summary. It is independent of the length of the reference summary; thus, it identifies the quality and the relevance of the text generated by the system. Recall is the quantitative value determining how well the generated summary captures the reference summary. F-Score is calculated by using the formula given in equation [7]. We have taken unigram, bigram and Longest common subsequence into the consideration which are represented by ROUGE-1, ROUGE-2 and ROUGE-L.

A major problem in the human generated summaries is the bias towards the topic which can be added by the individual intentionally or unintentionally. The System generated automatic text summarization is considered to be free from such bias. Extractive summarization tends to have minimal bias depending upon the sentences selected by the sentence scoring algorithm whereas due to the addition of the new text, abstractive summarization may end up having the bias. For the qualitative analysis, we classify the real and the system generated summaries on polarity by taking the range between [-1,+1] and the overall percentage change in the polarity was calculated to analyze the models.

Finally, the best extractive model is used to develop an end-to-end summarizer which automatically summarizes the web articles from any given web portal. We use the Wikipedia articles and parse the articles using beautiful soup, preprocess the article text and then feed into our model to generate the summaries.

4. EXPERIMENTAL RESULTS

We compare the extractive and abstractive summarization approaches separately because the abstractive summarization generates the new text while summarization which may or may not be present in the real summary. Thus, the ROUGE scores cannot be comparable for both the approaches.

For the extractive summarization on 'BBC news corpus', the baseline model using word frequency as a feature and priority queue for sentence selection resulted in the precision values shown in Figure 6 for each article in each topic using the baseline metric and the mean Rouge scores are shown in Figure 7.

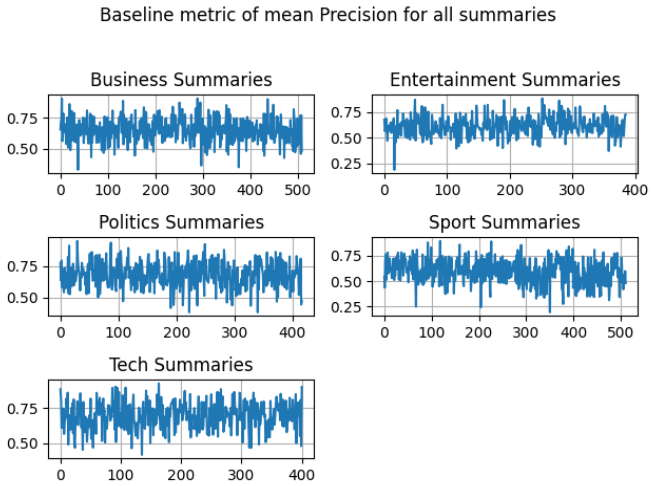

Fig. 6. Baseline precision metric for baseline model.

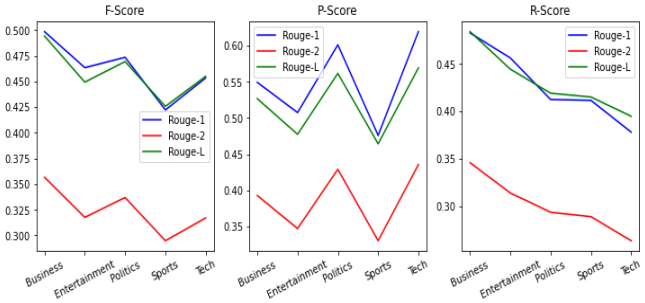

Fig. 7. Rouge Scores for baseline model

Baseline precision metric provides the mean unigram precision for business summaries as 0.64, entertainment summaries as 0.61, political summaries as 0.68, sports summaries as 0.59 followed by the tech summaries as 0.69. Thus, the average precision achieved on the whole dataset for the baseline metric is 0.64 and it is independent of the compression ratio as discussed. It can be considered as a descent score for the extractive summarizer.

Thus, we can consider the model for our baseline and improve upon. For the ROUGE scores, average F score for ROUGE-1 and ROUGE-2 for the whole dataset is averaged at 0.46 and 0.32

respectively. This depicts that the generated summary is required to be improved on the bigram's similarity for precision and recall.
For the qualitative analysis, Gold summary for a sample article is shown in Table 1 along with the baseline generated summary in Table 2.

Summary: timewarner said fourth quarter sales rose 2% to $11.1bn from $10.9bn.for the full-year, timewarner posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn.quarterly profits at us media giant timewarner jumped 76% to $1.13bn (£600m) for the three months to december, from $639m year-earlier.however, the company said aol's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues.its profits were buoyed by one-off gains which offset a profit dip at warner bros, and less users for aol.for 2005, timewarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.it lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters.time warner's fourth quarter profits were slightly better than analysts' expectations.

Table 1. Gold Summary

for the full year, timewarner posted a profit of 3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to 42.09bn. but its film division saw profits slump 27% to 284m, helped by box office flops alexander and catwoman, a sharp contrast to year earlier, when the third and final film in the lord of the rings trilogy boosted results. ad sales boost time warner profit quarterly profits at us media giant timewarner jumped 76% to 1.13bn 600m for the three months to december, from 639m year earlier. for 2005, timewarner is projecting operating earnings growth of around 5%, and expects higher revenue and wider profit margins.

Table 2. Baseline generated summary

The summary suggests to be meaningful from the individual's perspective, but the long sentences contribute a lot to the summary which is due to the increased sentence frequency count in the feature because of the large number of contributing tokens in the sentence.
The quantitative results for the second model where the word frequency, sentence length and the intersectional n-grams are considered as features are shown in the figure 8.

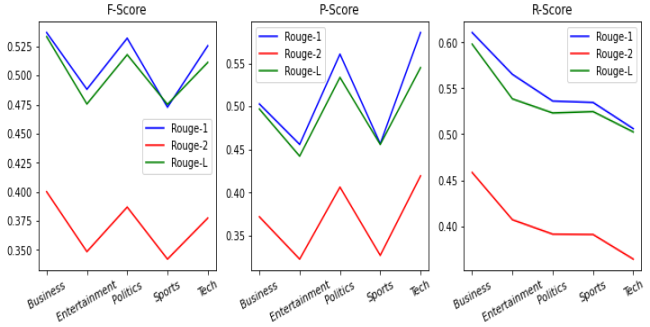

Fig. 8. Rouge scores for Model 2

The overall F-score for unigram ROUGE-1 and bigram ROUGE-2 for the dataset turns out to be 0.51 and 0.37 which is an increase of 11% and 16% in the scores from the baseline. Thus, ignoring the extremely long and short sentences improves the ROUGE scores as more comparable summaries to the real summaries are generated.
The sample summary generated by the system is depicted in Table 3. A shorter, concise summary is generated which is much better than the one generated by the baseline. It is the result of the sentence length and the intersectional n-gram

quarterly profits at us media giant timewarner jumped 76% to 1.13bn 600m for the three months to december, from 639m year earlier. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins. but its own internet business, aol, had has mixed fortunes. it lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding

Table 3. Summary generated by System 2

filtering where we ignore one of the sentences with the maximum bigram similarity count.
In system 3, where the implementation of the TextRank algorithm was used, a further increase of 16% in F-score can be seen for the unigrams and an increase of 23% can be seen for the bigrams when compared to the previous model. Thus, the TextRank implementation outperforms the previous models. The average ROUGE-1 score, and ROUGE-2 score stands at 59.2 and 45.6 respectively. Rouge scores are shown in Figure 9.

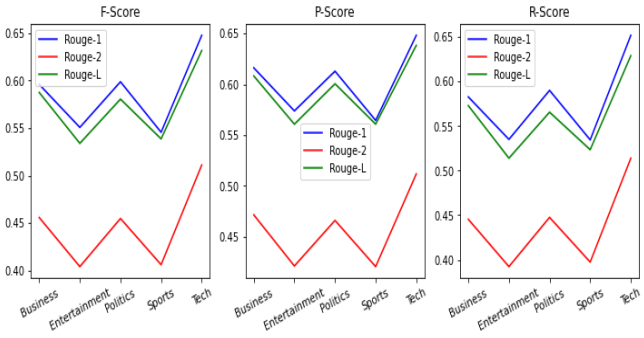

Fig. 9. Text Rank Rouge scores

The sample summary generated by this system is shown in Table 4. A much similar summary to the previous model can be seen. As this algorithm doesn't penalize the length of the sentences, thus the differing sentences appear to be longer which are contributing more towards the scores. This summary is more generalized from the others.

quarterly profits at us media giant timewarner jumped 76% to 1.13bn 600m for the three months to december, from 639m year earlier. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins. timewarner said fourth quarter sales rose 2% to 11.1bn from 10.9bn. however, the company said aol's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. time warner's fourth quarter profits were slightly better than analyst's expectations.

Table 4. TextRank generated summary

For the fourth system which is an ensemble architecture, where the sentences in the resultant summary are chosen on the basis of intersectional overlapping, we use the precision as the metric for making the comparisons as the resultant summary length will be less than the length of the summaries generated by the other systems with the compression ratio of 0.4. The precision score is shown in Figure 10.

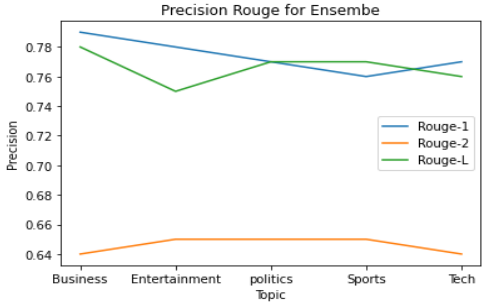

Fig. 10. Precision Score for Ensemble

Thus, an average of 0.77 ROUGE-1 and 0.65 ROUGE-2 for precision is achieved which is much better than the one's achieved by the individual models. The reason can be considered as the reduction in the summary length and the most relevant sentences to be the part of the summary which have the highest probability of appearance in the reference summary. The sample generated summary is shown in Table 5 which turns out to be the concise summary carrying no noisy irrelevant information but missing some important information which can surely be incorporated if the compression ratio for the contributing models is increased from 0.4.

quarterly profits at us media giant timewarner jumped 76% to 1.13bn 600m for the three months to december, from 639m year earlier. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.

Table 5. Ensemble Summary

For the overall cross-sectional comparison on precision and recall, the ensemble model turns out to generate the precise summaries, which is obvious due to the incorporation of the intersectional sentences in the resultant summary. The recall for the ensemble cannot be compared with other models as the length of the generated summary plays an important role for the recall calculation. Whereas, the PageRank and the improved model 2 generates the summaries with high recall depicting the models capturing the information from the reference summary in a better way.

For the Abstractive summarization, we used the "Amazon Food Reviews " dataset and the baseline model is considered to be the text to text transfer transformer. For this task, pretrained transformer model was used. The model configuration is shown in Table 6.

Parameters	Configuration
Num_beams	5
no_repeat_ngram_size	2
min_length	3
max_length	10
early_stopping	True
skip_special_tokens	True

Table 6. Transformer Model configuration

In the LSTM model, the input shape of the encoder component is taken as the maximum length of padded review i.e. 100 which we obtained by calculating the 75th percentile of the review lengths in our dataset. We define the embedding layer with input dimensions to be the maximum vocabulary size that we define for the review data which is 40000 ,output dimensions to be 300 which is according to the 300 dimensional GloVe and FastText embedding that we are employing, trainable parameter to be False due to the limitation of the computational resources and preventing the overfitting of the model on training data because of the changing weights at the embedding layer, mask_zero parameter is set to be True to mask(ignore some computations) the succeeded LSTM layer having 128 units with return_state set to be True which represents the cell state of the LSTM, dropout for the layer to be 0.3.

Decoder will have None in the Input Shape which serves us a special purpose of letting us choose the number of tokens during the training phase and the inference phase . During the training phase, we require this value to be equal to the maximum summary length that is defined as 5 by taking the 75th percentile of all the summary lengths whereas at the time of prediction on test set, the value of this parameter is required to be 1 so that we are able to decode one token at a moment thereby increasing the flexibility of the LSTM model. The embedding layer for the decoder setup is defined with input dimensions as 15000 for the same reason as stated above in encoder component, output dimension is set to 300, trainable parameter is set to be False and mask_zero parameter is set to be True. LSTM layer for decoder component with 128 units, dropout is set to be 0.3. Return sequences is set to be true because for decoding, both the hidden states and the cell states of the LSTM model are required. The decoder output is calculated followed by the dense layer having softmax activation function which returns the resultant value of decoder output.

For the training of the sequence to sequence model, the optimizer is taken as Adam with learning rate=0.002, loss as categorical crossentropy. Model is trained for 50 epochs using fit_generator instead of fit because values are fed in batches for effective utilization of memory. The LSTM model is implemented using GloVe and FastText embeddings separately. The evaluation results for Transformer and LSTM models for 2.5% of the test data with average summary length of 5 words are shown in Table 7, Table 8 and Figure 11.

	F-Score	P-Score	R-Score
T5 Transformer	0.729	0.74	0.72
LSTM (GloVe)	0.664	0.68	0.65
LSTM (FastText)	0.704	0.72	0.69

Table 7. Rouge-L Scores Abstractive Techniques (2.5% Test data)

Review: great full bodied organic coffee with medium acidity the addition of yerba mate to the arabica beans gives this coffee a nice mildly sweet finishing note. the cost is reasonable for the coffee i drink many types of organic gourmet coffee this rate as one of my top three all time favorites. **Gold Summary:** full bodied coffee. **Transformer Summary:** bodied coffee **LSTM summary(GloVe):** great bodied UNK **LSTM summary(Fast Text):** great bodied coffee
Review: after trying many other brands we found ancient harvest linguini finally a gluten free pasta that does n t stick clump or disintegrate the pricing on amazon is about 60 cents better per box than any grocery store pricing i ve seen we cook two boxes for our family of five and have less than one serving left over we also love the elbow noodles the spaghetti is passable but the consistency of the linguini is better in my opinion if you eat gluten free this pasta is for you **Gold Summary:** Best GF pasta on the market! **Transformer Summary:** gluten free pasta **LSTM summary(GloVe):** best UNK pasta ever **LSTM summary(FastText):** best gluten pasta ever

Table 8. Summaries Abstractive Techniques

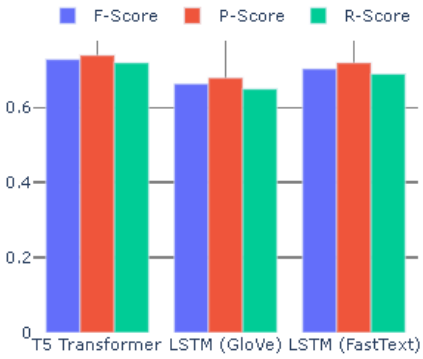

Fig. 11. Rouge-L scores

From Table 7, We can conclude that F-Score for the pretrained transformer model which is taken as baseline is very high which is due to the fact that the minimal sized test set with summary length averaged at 5 was taken. Thus, the short summaries generated by the system are much comparable to the real summaries for the given dataset. For the LSTM with GloVe embeddings, descent results can be seen with no improvement on the baseline. Further when the FastText is used for the embedding, a slight increase in the scores can be seen. For the qualitative analysis, it can be seen in Table 8 that summaries generated by the GloVe LSTM contain the <UNK> words i.e. the vocabulary formed wasn't good enough. This problem was tackled by FastText to some extent. Time taken by the Pretrained T5 transformer to generate the summaries is 583 minutes for the test data comprising of 2.5% of the whole dataset whereas the LSTM GloVe and LSTM FastText takes approximately 291 and 312 minutes respectively to train and generate the summaries.
A change in the polarity between the reference summary and the generated summary can be seen from the example summaries in Table 8. Thus, average percentage change in polarity was calculated between the real and the generated summaries to check for the bias added by abstractive models. The sample polarity results are shown in Table 9.

Gold Summary	Real	T5	LSTM(GloVe)	LSTM(FastText)
"full bodied coffee"	0.35	0	0.7	0.8
"Best GF pasta on the market!"	1.0	0.4	1.0	1.0

Table 9. Sample polarity change in summaries

On average, it was seen that the baseline model T5 provides the best results with the least deviation in the overall polarity from the real summaries. Average percentage change in the polarity with different models for the complete test set of 14211 summaries i.e. 2.5% of the dataset is shown in Figure 12. LSTM with GloVe embeddings tend to have the highest change in the polarity which is 77% whereas the LSTM with FastText tends to have the polarity change of 56% which is much lower in comparison to the GloVe.

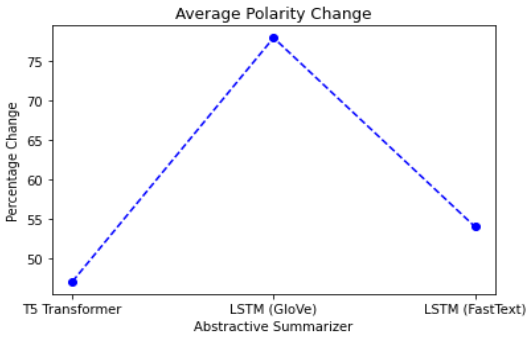

Fig. 12. Average Polarity change

For the end-to-end automatic text summarizer, Ensemble model was used for the extractive summarization. We used our ensemble model to generate the summary because of the long article size and coherence can be better preserved with the extractive summarization for the long sequences and the best model to generate the concise summary obtained was the ensemble model. The compression ratio was set at 0.02.

The generated summary is shown in Table 10 which covers the important information in the article. Further, the better summary containing more relevant information can be obtained by increasing the compression ratio of the models used in ensemble architecture individually.

Url: https://en.wikipedia.org/wiki/Coronavirus

Ensemble Summary:
Coronaviruses constitute the subfamily Orthocoronavirinae, in the family Corona viridae order Nidovirales, and realm Riboviria.[6][7] They are enveloped viruses with a positive-sense single-stranded RNA genome and a nucleocapsid of helical symmetry.[8] The genome size of coronaviruses ranges from approximately 26 to 32 kilobases, one of the largest among RNA viruses.[9] They have characteristic club-shaped spikes that project from their surface, which in electron micrographs create an image reminiscent of the solar corona, from which their name derives.[10]
The name "coronavirus" is derived from Latin corona, meaning "crown" or "wreath", itself a borrowing from Greek κορώνη korṓnē, "garland, wreath".[11][12] The name was coined by June Almeida and David Tyrrell who first observed and studied human coronaviruses.[13] The word was first used in print in 1968 by an informal group of virologists in the journal Nature to designate the new family of viruses.[10] The name refers to the characteristic appearance of virions (the infective form of the virus) by electron microscopy, which have a fringe of large, bulbous surface projections creating an image reminiscent of the solar corona or halo.[10][13] This morphology is created by the viral spike peplomers, which are proteins on the surface of the virus.[14]

Table 10. Wikipedia Summary by Ensemble

5. CONCLUSION

This work presented two pipelines viz. extractive and abstractive summarization for the task of automatic text summarization. Feature extraction techniques demonstrated in model 2 resulted in an increase of 11% in the ROUGE-1 F-score and an increase of 16% in the ROUGE-2 F-score from the baseline setup. TextRank algorithm implementation further improved the results obtained in model 2 by giving the ROUGE F-scores of 0.59 and 0.45 for unigram and bigram similarity which is an increase of 16% and 23% from the model 2. Thus, the existing implementation of TextRank produces better summaries than the defined model 2 for the given dataset. Finally, the ensemble model, which generates the summaries by selecting the sentences using intersectional overlapping of the sentences in different summaries generated by Model 2, TextRank and Summa provided the precision score of 0.77 for unigrams and 0.65 for bigrams. This P-score for ROUGE-1 and ROUGE-2 is better than the other models as most of the sentences generated in the summary are present in the reference summary. Thus, for the extractive summarization we can conclude that the Ensemble model works better holistically when the compression ratio of the underlying models is increased to generate the comparable number of sentences to the reference summary. For the abstractive summarization, the baseline transformer model generates much better summaries than the implemented GloVe LSTM with least polarity change in the generated summaries. FastText LSTM performs better than the GloVe as the summaries doesn't include large number of <UNK> tokens which degrades the F-Score of GloVe LSTM by 9% from the T5 transformer model. The time complexity for T-5 transformer turns out to be much greater than the LSTM variants exceeding approximately by 271 minutes . Thus, we conclude that the T-5 transformer generates better summaries which are comparable to the reference summaries whereas the FastText LSTM also performs better when the time complexity is a constraint. Finally, the end to end web

summarizer implemented using ensemble model properly summarizes the web articles.

The embeddings have a large impact on the summarizer system, which can be further explored by employing the embeddings trained on larger corpus. The coherency in the long summaries by abstractive models is still researchable and the implementation of the global and local attention mechanism can be explored to improve the resu

REFERENCES

[1] Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock, “Headline generation based on statistical translation,” In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL ’00). Association for Computational Linguistics, USA, 318–325, DOI:https://doi.org/10.3115/1075218.1075259, 2000.

[2] Federico Barrios, Federico López, Luis Argerich and Rosa Wachenchauzer, “Variations of the Similarity Function of TextRank for Automated Summarization,” arXiv: 1602.03606, 2016.

[3] Saggion, Horacio. "Creating Summarization Systems with SUMMA," LREC, 2014.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li and Peter J. Liu, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” arXiv: 1910.10683, 2019.

[5] Anjali R. Deshpande , Lobo L. M. R. J. , "Text Summarization using Clustering Technique," International Journal of Engineering Trends and Technology (IJETT), V4(8):3348-3351, ISSN:2231-5381, www.ijettjournal.org. published by seventh sense research group, Jul 2013.

[6] X. Cui, T. E. Potok and P. Palathingal, "Document clustering using particle swarm optimization," Proceedings, IEEE Swarm Intelligence Symposium, SIS, Pasadena, CA, USA, pp. 185-191, doi: 10.1109/SIS.2005.1501621, 2005.

[7] S. Karwa and N. Chatterjee, "Discrete Differential Evolution for Text Summarization," International Conference on Information Technology, Bhubaneswar, pp. 129-133, doi: 10.1109/ICIT.2014.28, 2014.

[8] Babar, Samrat and Patil, Pallavi, “Improving Performance of Text Summarization,” Procedia Computer Science. 46. 10.1016/j.procs.2015.02.031, 2015.

[9] Atif Khan, Naomie Salim, Yogan Jaya Kumar, “A framework for multi-document abstractive summarization based on semantic role labelling,” Applied Soft Computing, Volume 30, Pages 737-747, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2015.01.070, 2015.

[10] Jay J. Jiang and David W. Conrath, “Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy,” arXiv: cmp-lg/9709008, 1997.

[11] Gatt, Albert & Reiter, Ehud, “SimpleNLG: A realisation engine for practical applications,” Proceedings of the 12th European Workshop on Natural Language Generation, ENLG, 90-93. 10.3115/1610195.1610208, 2009.

[12] Changjian Fang, Dejun Mu, Zhenghong Deng, Zhiang Wu, "Word-sentence co-ranking for automatic extractive text summarization," Expert Systems with Applications,Volume 72, Pages 189-195, ISSN 0957-4174, 2017.

[13] Page,Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry, “The PageRank Citation Ranking: Bringing Order to the Web, Technical Report. Stanford InfoLab, 1999.

[14] Rossiello, Gaetano and Basile, Pierpaolo and Semeraro, Giovanni, “Centroid-based Text Summarization through Compositionality of Word Embeddings,” Proceedings of the MultiLing Workshop on Summarization and Summary Evaluation Across Source Types and Genres, 2017.

[15] Alexander M. Rush and Sumit Chopra and Jason Weston, “A Neural Attention Model for Abstractive Sentence Summarization,” arXiv: 1509.00685, 2015.

[16] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, null (3/1/2003), 1137–1155, 2003

[17] Ramesh Nallapati and Bowen Zhou and Cicero Nogueira dos santos and Caglar Gulcehre and Bing Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," arXiv:1602.06023, 2016.

[18] Junyoung Chung and Caglar Gulcehre and KyungHyun Cho and Yoshua Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, arXiv: 1412.3555, 2014.

[19] Sebastien Jean and Kyunghyun Cho and Roland Memisevic and Yoshua Bengio, “On Using Very Large Target Vocabulary for Neural Machine Translation,” arXiv: 1412.2007, 2014.

[20] Li, Chenliang, Weiran Xu, Si Li, and Sheng Gao, “Guiding generation for abstractive text summarization based on key information guide network," In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 55-60, 2018.

[21] Federico Barrios and Federico López and Luis Argerich and Rosa Wachenchauzer, “Variations of the Similarity Function of TextRank for Automated Summarization,” arXiv: 1602.03606, 2016.

[22] Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv: 1810.04805, 2018.

[23] Yang Liu, “Fine-tune BERT for Extractive Summarization,” arXiv: 1903.10318, 2019.

[24] Lin,Chin-Yew, “ROUGE: A Package for Automatic Evaluation of Summaries”, Association for Computational Linguistics, Barcelona, Spain, pp. 74-81, https://www.aclweb.org/anthology/W04-1013, 2004.