Abstractive Summarization for News Headline Generation

Yogalakshmi Saravanan(5497555)
yogalakshmi@knights.ucf.edu

Megala Jeyapal(5481906)
megalajeyapal@Knights.ucf.edu

ABSTRACT :

This report discusses an abstract text summarization model designed to create brief news headlines from news articles, using an encoder-decoder mechanism. The model utilizes a custom attention mechanism to generate the headline one word at a time in an abstract manner. The report provides an overview of the theoretical background of the model, the dataset used, the implementation process, and the evaluation. Additionally, the report discusses the initial results and potential areas for improvement in the future. Overall, this report serves as a detailed guide to the implementation, findings, and potential challenges associated with this project.

Keywords: abstractive summarization, attention mechanism,

1.INTRODUCTION:

With the exponential growth of digital information, it is becoming increasingly difficult for users to efficiently sift through vast amounts of text to identify the most relevant information. This has led to the development of text summarization, which aims to generate a brief summary of a longer document that captures its key concepts.

Automatic text summarization has been widely studied and is typically divided into two main approaches: extraction and abstraction. In extraction-based summarization, important sentences or phrases are extracted from the original document to form the summary, while in abstractive summarization, the summary is generated using new sentences that capture the main idea of the original text. While extraction-based summarization is simpler, abstractive summarization is more challenging and requires the generation of new sentences that are both grammatically correct and semantically accurate.

Abstractive text summarization is a task of particular interest, as it aims to generate a summary that is not simply a selection of existing sentences from the original document, but a compressed paraphrasing of the main contents of the document, potentially using vocabulary not seen in the original text. This can be thought of as a mapping problem, where an input sequence of words from the original document is mapped to a target sequence of words, which form the summary.

Recently, deep learning-based models, specifically sequence-to-sequence models, have shown great success in many natural language processing tasks, such as machine translation, speech recognition, and video captioning. Within the framework of sequence-to-sequence models, a particularly relevant model to abstractive text summarization is the attentional Recurrent Neural Network (RNN) encoder-decoder model proposed by Bahdanau et al. (2014). This model uses an encoder to transform the input sequence into a fixed-length representation, which is then used by the decoder to generate the target sequence word by word, while incorporating an attention mechanism to focus on different parts of the input during the decoding process.

One of the main challenges in abstractive text summarization is to optimize the compression of the original document in a lossy manner such that the key concepts are preserved while ensuring that the summary remains grammatically correct and semantically accurate. Another challenge is the length of the summary, which needs to be short enough to be useful but long enough to contain the main idea of the original text.

The potential applications of abstractive text summarization are numerous, ranging from e-commerce product highlights to social media content targeting. In this report, we present an abstractive text summarization model that generates concise headlines from news articles using a custom attention mechanism. We will describe the theory behind our model, the dataset used, our implementation, and the evaluation metrics used to measure the performance of our model. We will also discuss the initial results of our experiments and the potential issues and future scope of this project.

2.PROBLEM STATEMENT:

The problem addressed in this report is the challenge of generating news headlines of news website using an abstractive text summarization model. While automatic text summarization has become increasingly important in a variety of applications, the traditional extractive methods of summarization may not always capture the essential meaning of a text. Abstractive summarization, on the other hand, aims to create a summary that goes beyond a simple selection of sentences from the original document, by generating new sentences that capture the key ideas and concepts. The main challenge in abstractive summarization is to compress the original document in a way that retains the important information while also ensuring that the summary is grammatically correct and readable.

The proposed solution to this problem is a sequence-to-sequence model that employs a custom attention mechanism to generate a headline word by word. Specifically, the report outlines the use of an attentional Recurrent Neural Network (RNN) encoder-decoder model, which has shown success in other sequence-to-sequence tasks such as machine translation and speech recognition. The report describes the dataset used for training the model, the implementation details, and the evaluation metrics used to assess the performance of the model. The ultimate goal of this project is to improve the accuracy and efficiency of abstractive summarization, which has numerous applications in e-commerce, social media, and other internet-based industries.

3. RELATED WORK:

In the field of abstractive text summarization using sequence-to-sequence (seq2seq) models with LSTM layers, the closest related work to our proposed RNN Encoder-Decoder is the Recurrent Continuous Translation Model (Model 2) introduced by Kalchbrenner and Blunsom in 2013. Their model also includes an encoder and decoder, but with a different architecture. Specifically, they used a convolutional n-gram model (CGM) for the encoder and a hybrid of an inverse CGM and a recurrent neural network for the decoder. While their model shares some similarities with ours, such as the use of an encoder-decoder architecture, the key difference lies in the specific models used for the encoder and decoder.

Our proposed model uses LSTM layers for both the encoder and decoder, while Kalchbrenner and Blunsom used a combination of a convolutional n-gram model and a hybrid inverse CGM-recurrent neural network for their encoder and decoder, respectively. As such, our proposed model may offer different advantages and performance characteristics compared to the Recurrent Continuous Translation Model.

Attention layer we used in our model is based on the Bahdanau et al.(2014). The Bahdanau attention mechanism has been shown to be particularly effective in handling longer input sequences and capturing important contextual information from the input text. This has been demonstrated in various studies on abstractive text summarization, where the Bahdanau attention mechanism has been used to significantly improve the quality and coherence of generated summaries.

For example, in a recent study by Li et al. (2021), the Bahdanau attention mechanism was used in a neural network-based model for abstractive text summarization of scientific papers. The authors found

that incorporating the Bahdanau attention mechanism improved the overall performance of the model, particularly in terms of accurately capturing key information from the input text. Similarly, in a study by Nallapati et al. (2016), the Bahdanau attention mechanism was used in a sequence-to-sequence model for abstractive text summarization of news articles. The authors found that the Bahdanau attention mechanism was able to improve the model's ability to generate informative and coherent summaries, particularly for longer input texts.

Overall, the Bahdanau attention mechanism has been shown to be an effective technique for improving the quality of abstractive text summarization. This mechanism can be applied in various contexts and has been demonstrated to provide significant improvements in the accuracy and coherence of generated summaries.

4. DATASET :

The dataset used for our model training is a combination of two datasets from Kaggle, with a total of over one million data points. The data was primarily scraped from the Inshorts news app and includes short summaries of larger articles, along with their respective headlines. Due to resource constraints, we initially used only the short summaries for testing our model's performance before incorporating the complete articles. The final dataset has 102,915 samples and includes columns for the author's name, headlines, article URL, short text, and the complete article.

	author	date	headlines	read_more	text	ctext
0	Chhavi Tyagi	03 Aug 2017,Thursday	Daman & Diu revokes mandatory Rakshabandhan in...	http://www.hindustantimes.com/india-news/raksh...	The Administration of Union Territory Daman an...	The Daman and Diu administration on Wednesday ...
1	Daisy Mowke	03 Aug 2017,Thursday	Malaika slams user who trolled her for 'divorc...	http://www.hindustantimes.com/bollywood/malaik...	Malaika Arora slammed an Instagram user who tr...	From her special numbers to TV?appearances, Bo...
2	Arshiya Chopra	03 Aug 2017,Thursday	'Virgin' now corrected to 'Unmarried' in IGNMS...	http://www.hindustantimes.com/patna/bihar-igm...	The Indira Gandhi Institute of Medical Science...	The Indira Gandhi Institute of Medical Science...
3	Sumedha Sehra	03 Aug 2017,Thursday	Aaj aapne pakad liya: LeT man Dujana before be...	http://indiatoday.intoday.in/story/abu-dujana-...	Lashkar-e-Taiba's Kashmir commander Abu Dujana...	Lashkar-e-Taiba's Kashmir commander Abu Dujana...
4	Aarushi Maheshwari	03 Aug 2017,Thursday	Hotel staff to get training to spot signs of s...	http://indiatoday.intoday.in/story/sex-traffic...	Hotels in Maharashtra will train their staff t...	Hotels in Mumbai and other Indian cities are t...

Fig 1. News dataset

5. PREPROCESSING :

Below are the preprocessing steps applied for both Headline and Text.

- Converted the input to lowercase.
- Removed punctuation, special characters and double quotes.
- Replaced contractions with expanded forms.
- Eliminated apostrophe and non-alphabetical characters.
- Removed stop words and very short words.
- Dropped all data samples with empty ‘text’ and ‘headline’.
- Tokenized both ‘text’ and ‘headline’ and created two different vocabularies.
- Padded short sequence of words with 0’s.

5.1. Merging datasets:

We combined two Kaggle datasets that had columns named 'headline' and 'text'. The resulting merged dataset has a specific size as below,

```
data.shape
```

```
(102915, 6)
```

Fig 2. Final data size

5.2. Train-Test data splitting:

Merged dataset ‘cleaned_data’ split into two sets as Training data(80%) and Testing data(20%). Training data is used for Parameter tuning and model training. Finally tested the model predictions and performances with Testing data.

6. EXPLORATORY DATA ANALYSIS:

6.1 Word distribution:

We performed an exploratory data analysis (EDA) on a cleaned dataset especially on two columns, 'text' and 'headline', representing the body and the headline of articles respectively. Using the Python library Matplotlib, we created histograms to visualize the distribution of word counts for both columns. By analyzing the histograms, we were able to determine the maximum number of words in 'text' and 'headline' which were found to be 53 and 14 respectively.

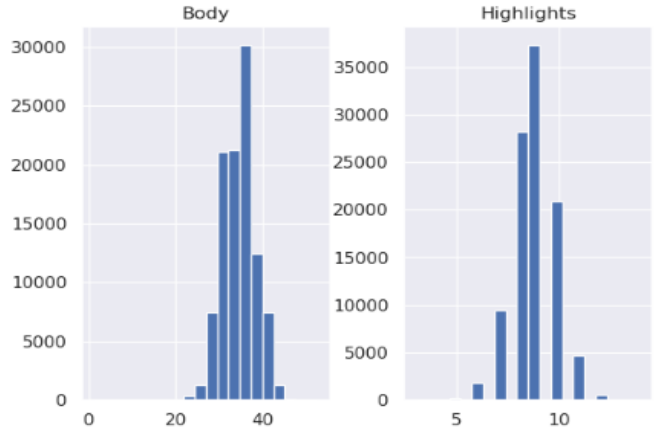

Fig 3. Word Distribution of ‘text’ and ‘headline’ respectively

```
print(news_length, headline_length)
```

```
53 14
```

Fig 4. Maximum length of ‘text’ and ‘headline’

6.2 Feature Distribution:

- 35 words and 218 characters per row are the mean values of ‘text’
- 9 words and 52 characters per row are the mean values of ‘headline’ and they are very close to the median values.
- Standard deviations are quite small.
- The maximum number of words or characters are far away from the mean values in both ‘text’ and ‘headline’
- Indicating that there are some registers with values out of range or outliers.

	text_word_count	text_char_count	text_stopw_count	text_punc_count	text_word_density
count	102915.000000	102915.000000	102915.000000	102915.0	102915.000000
mean	34.483195	217.560472	0.040237	0.0	0.158776
std	3.698919	28.916186	0.295104	0.0	0.012692
min	1.000000	4.000000	0.000000	0.0	0.113043
25%	32.000000	198.000000	0.000000	0.0	0.149798
50%	34.000000	218.000000	0.000000	0.0	0.158103
75%	37.000000	238.000000	0.000000	0.0	0.166667
max	53.000000	319.000000	6.000000	0.0	0.220588

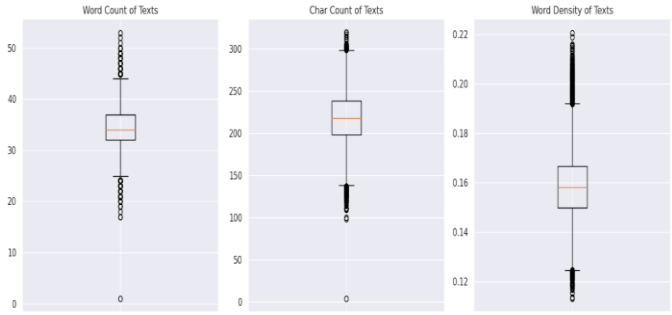

Fig 4. Feature distribution of 'text'

	headline_word_count	headline_char_count	headline_stopw_count	headline_punc_count	headline_word_density
count	102915.000000	102915.000000	102915.000000	102915.0	102915.000000
mean	8.789681	51.496206	0.012904	4.0	0.168062
std	1.104305	5.869791	0.160614	0.0	0.017304
min	3.000000	21.000000	0.000000	4.0	0.083333
25%	8.000000	48.000000	0.000000	4.0	0.155556
50%	9.000000	52.000000	0.000000	4.0	0.166667
75%	10.000000	56.000000	0.000000	4.0	0.180000
max	14.000000	71.000000	4.000000	4.0	0.243902

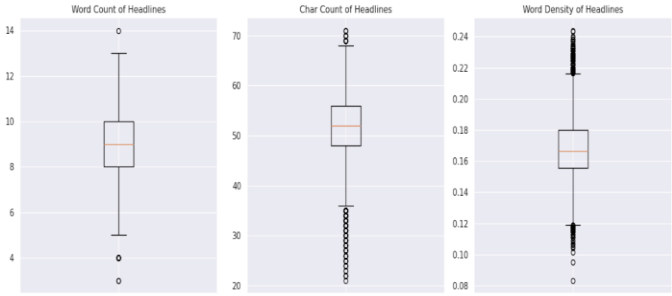

Fig 5. Feature distribution of 'headline'

7. MODEL BUILDING AND ATTENTION MECHANISM:

7.1. Sequence to Sequence model:

Sequence-to-sequence (seq2seq) model is a type of neural network architecture designed for processing sequences of input data and generating sequences of output data. The seq2seq model consists of two main components: an encoder and a decoder. The encoder takes in a sequence of input data and produces a fixed-length vector representation of the entire sequence. This vector is then passed to the decoder, which generates a sequence of output data based on the input vector and an initial "start" token. During training, the input and output sequences are provided as pairs, and the model is trained to predict the output sequence given the input sequence. The loss function is typically a measure of the difference between the predicted output sequence and the true output sequence.

The reason why we chose the seq2seq model is that it can handle variable-length input and output sequences. This makes it well-suited for our task news headline generation, where the length of the input and output sentences may vary widely. Additionally, by using a fixed-length vector representation of the input sequence, the seq2seq model can capture the semantic meaning of the entire sequence, rather than just the meaning of individual words or tokens.

7.2. Encoder-Decoder Architecture:

Both the encoder and the decoder are LSTM models. The LSTM model is a type of neural network that is commonly used in sequence-to-sequence models. Both the encoder and decoder in the LSTM model are composed of LSTM cells.

During encoding, the input sequence is processed element by element to produce an internal state vector, which consists of a hidden state vector and a cell state vector. These internal states aim to summarize the information from the input sequence and help the decoder make accurate predictions for the output sequence. The hidden states h_i are computed using the formula:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

The outputs generated by the encoder are typically discarded, and only these internal states are retained. The aim of these internal states is to capture the relevant information from all the input elements in a compact and meaningful way, which can then be used by the decoder to generate accurate predictions for the output sequence. By discarding the output sequence and only preserving the internal states, the encoder can produce a fixed-length representation of the input sequence.

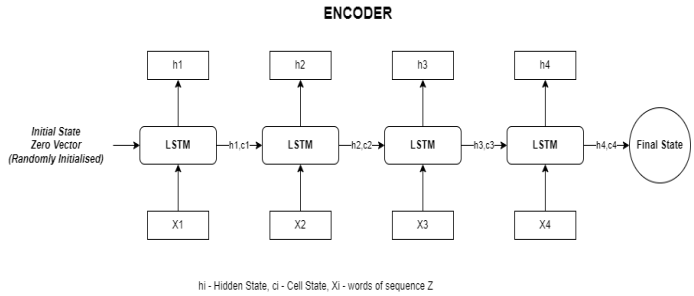

Fig 6. Encoder architecture in Seq-to-Seq model

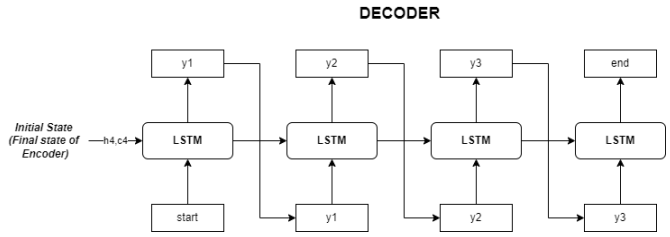

Fig 7. Decoder architecture in Seq-to-Seq model

The decoder generates the output sequence element by element, where each element is predicted by a separate recurrent unit in the LSTM network. Each unit accepts the hidden state from the previous unit and generates an output and its own hidden state using the formula:

$$h_t = f(W^{(hh)}h_{t-1})$$

The final output is obtained by applying the softmax function to the hidden state at the current time step multiplied by the respective weight. The softmax function creates a probability vector that helps in determining the final output.

$$y_t = softmax(W^S h_t)$$

7.3. Attention Layer:

Attention was introduced in the paper[1] to address the fixed representation problem in the encoder-decoder architecture.

An attention mechanism is a part of a neural network. At each decoder step, it decides which source parts are more important. In this setting, the encoder does not have to compress the whole source into a single vector - it gives representations for all source tokens.

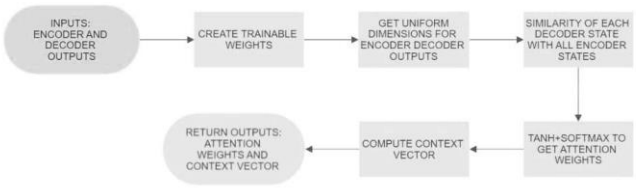

Fig 8. Attention mechanism

Our attention layer is based on the Bahdanau attention mechanism. The attention layer is defined separately to calculate the attention weights. The layer inherits from Layer in keras. The attention layer will take in two inputs, namely the output hidden states from the encoder and decoder separately. The layer makes use of three trainable weights w1,w2, and w3 that are initiated in the build function. The control will then enter the call function that contains the two main methods required for calculating the weights: the energy step and context step. The energy step first dot produces the encoder and decoder outputs with trainable weights to get fixed dimensionality. Each decoder state is compared with all the encoder states to get similarity scores through dot product. The values are passed through a tanh activation function followed by softmax. After these operations, a probability distribution of the attention scores(energy scores) is computed. Using these attention scores the context vector will be calculated by taking the weighted sum of the attention weights with the encoder states. The context vector and the attention weights are returned as output. This provides more information for the model to judge which parts of the input sentence need more attention.

8.MODEL TRAINING AND ARCHITECTURE:

Below is our basic model architecture used for this project

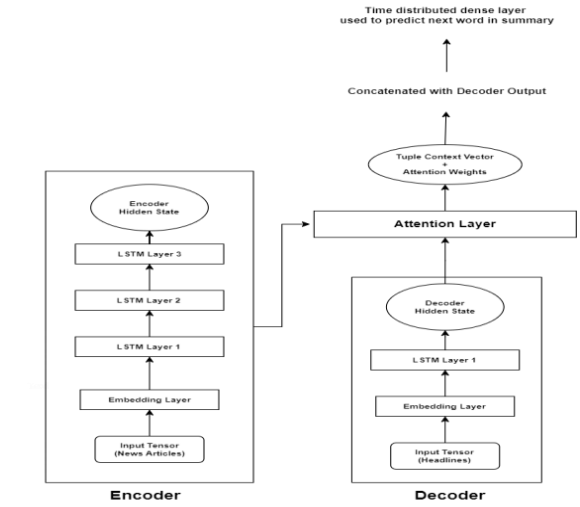

Fig 8. Model Basic Architecture

We used three different models for headline generation.

8.1. Model 1 :

We first implemented a basic encoder decoder model. The text and the summaries are passed through an embedding layer from Keras. The encoder is built using two LSTM layers and the decoder is built using one LSTM layer. Embedding dimensions and latent dimensions are 200 and 300 initially. The output from the encoder and the decoder is fed into the attention layer. The attention layer outputs are concatenated with the decoder output and a time-distributed dense layer with softmax function is applied. This will return a probability distribution of all the vocabulary in the input giving a score for each word. From these scores the highest score is taken for the next word generation. This model is compiled using rmsprop optimizer for 10 epochs. The decode sequence function is used to generate the headlines for 50 input texts from the test data and the BLEU and ROUGE scores are calculated.

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 53)]         0           []

embedding (Embedding)           (None, 53, 200)      13925400    ['input_1[0][0]']

input_2 (InputLayer)            [(None, None)]       0           []

lstm (LSTM)                     [(None, 53, 300),    601200      ['embedding[0][0]']
                                 (None, 300),
                                 (None, 300)]

embedding_1 (Embedding)         (None, None, 200)    5979600     ['input_2[0][0]']

lstm_1 (LSTM)                   [(None, 53, 300),    721200      ['lstm[0][0]']
                                 (None, 300),
                                 (None, 300)]

lstm_2 (LSTM)                   [(None, None, 300),  601200      ['embedding_1[0][0]',
                                 (None, 300),                     'lstm_1[0][1]',
                                 (None, 300)]                     'lstm_1[0][2]']

attention_layer (AttentionLaye  ((None, None, 300),  180300      ['lstm_1[0][0]',
r)                               (None, None, 53))               'lstm_2[0][0]']

concat_layer (Concatenate)      (None, None, 600)    0           ['lstm_2[0][0]',
                                                                 'attention_layer[0][0]']

time_distributed (TimeDistribu  (None, None, 29898)  17968698    ['concat_layer[0][0]']
ted)

==================================================================================================
Total params: 39,977,598
Trainable params: 39,977,598
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig 9. Model 1 Architecture

8.2. Model 2 :

The second model is a fine tuned version of model one. The number of encoder lstm layers is increased to three and the latent dimension increased to 400. Instead of rmsprop optimizer adam optimizer is used and the number of epochs is increased to 10 to allow the model to converge better. Similarly after prediction the evaluation metrics are applied.

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 53)]         0           []

embedding (Embedding)           (None, 53, 200)      13925400    ['input_1[0][0]']

lstm (LSTM)                     [(None, 53, 400),    961600      ['embedding[0][0]']
                                 (None, 400),
                                 (None, 400)]

lstm_1 (LSTM)                   [(None, 53, 400),    1281600     ['lstm[0][0]']
                                 (None, 400),
                                 (None, 400)]

input_2 (InputLayer)            [(None, None)]       0           []

lstm_2 (LSTM)                   [(None, 53, 400),    1281600     ['lstm_1[0][0]']
                                 (None, 400),
                                 (None, 400)]

embedding_1 (Embedding)         (None, None, 200)    5979600     ['input_2[0][0]']

lstm_3 (LSTM)                   [(None, 53, 400),    1281600     ['lstm_2[0][0]']
                                 (None, 400),
                                 (None, 400)]

lstm_4 (LSTM)                   [(None, None, 400),  961600      ['embedding_1[0][0]',
                                 (None, 400),                     'lstm_3[0][1]',
                                 (None, 400)]                     'lstm_3[0][2]']

attention_layer (AttentionLaye  ((None, None, 400),  320400      ['lstm_3[0][0]',
r)                               (None, None, 53))               'lstm_4[0][0]']

concat_layer (Concatenate)      (None, None, 800)    0           ['lstm_4[0][0]',
                                                                 'attention_layer[0][0]']

time_distributed (TimeDistribu  (None, None, 29898)  23948298    ['concat_layer[0][0]']
ted)

==================================================================================================
Total params: 49,941,698
Trainable params: 49,941,698
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig 10. Model 2 Architecture

8.3. Model 3:

The third model is different from the other two in the sense that it uses pretrained embeddings. Instead of using the embedding layer from keras we tried using GloVe to create a pretrained embedding layer. After downloading GloVe we created an embedding matrix and then

used that to form our embedding layer. The model has 2 lstm layers for encoder and one lstm layer for decoder. The latent dimensions are reduced to 256. Adam optimizer is used and the model is trained for 3 epochs and the evaluation is performed.

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_2 (InputLayer)           [(None, None)]       0           []

 input_1 (InputLayer)           [(None, 14)]         0           []

 embedding_2 (Embedding)        multiple             15262000    ['input_1[0][0]',
                                                                  'input_2[0][0]']

 lstm (LSTM)                    [(None, 14, 256),    467968      ['embedding_2[0][0]']
                                 (None, 256),
                                 (None, 256)]

 lstm_1 (LSTM)                  [(None, 14, 256),    525312      ['lstm[0][0]']
                                 (None, 256),
                                 (None, 256)]

 lstm_2 (LSTM)                  [(None, None, 256),  467968      ['embedding_2[1][0]',
                                 (None, 256),                     'lstm_1[0][1]',
                                 (None, 256)]                     'lstm_1[0][2]']

 attention_layer (AttentionLaye  ((None, None, 256),  131328     ['lstm_1[0][0]',
 r)                              (None, None, 14))                'lstm_2[0][0]']

 concat_layer (Concatenate)     (None, None, 512)    0           ['lstm_2[0][0]',
                                                                  'attention_layer[0][0]']

 time_distributed (TimeDistribu  (None, None, 76310)  39147030   ['concat_layer[0][0]']
 ted)

==================================================================================================
Total params: 56,001,606
Trainable params: 56,001,606
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig 11. Model 3 Architecture

9. EVALUATION METRIC:

The two main evaluation metrics we have used are the BLEU and rouge scores to evaluate how similar our predicted headlines are from the actual headlines.

9.1. BLEU Score :

BLEU (Bilingual Evaluation Understudy) score is a metric that is commonly used to evaluate the quality of machine-generated text. It measures the similarity between the machine-generated text and a set of human-generated reference texts. The score ranges from 0 to 1, where a score of 1 means the machine-generated text is identical to the human-generated reference texts. Mathematically, BLEU Score is given as follows:

$$\text{BLEU Score} = \text{BP} * \exp\left(\sum_{n=1}^{4} \cdot \frac{1}{n} P_n\right)$$

BP is the brevity penalty which is used to penalize the machine-generated translations that are shorter than the reference translations.

P_n is the n-gram modified precision score.

9.2. ROUGE Score:

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used to evaluate the quality of machine-generated summaries or texts. ROUGE measures the overlap between the machine-generated text and a set of human-generated reference texts in terms of n-grams or sequences of words. There are several variants of ROUGE, such as ROUGE-1, ROUGE-2, and ROUGE-L, which measure the overlap between unigrams, bigrams, and the longest common subsequence (LCS) of words, respectively. The score ranges from 0 to 1, where a score of 1 means the machine-generated text is identical to the human-generated reference texts. The formula for ROUGE score is as follows:

$$\text{ROUGE} = \sum (\text{Recall of n-grams})$$

Recall of n-grams is the fraction of n-grams that are present in both the generated summary and the reference summaries, divided by the total number of n-grams in the reference summaries. It measures the ability of the generated summary to capture the important information present in the reference summaries.

10. EXPERIMENTAL RESULTS AND DISCUSSION:

We have used both BLEU and ROUGE-1 score to evaluate the quality of our model predictions.

10.1. Model 1 Predictions:

The predictions generated by Model 1 are not accurate, as it repeatedly uses the same phrases or sentences for all input information.

	Information	Actual Headline	Predicted Headline
0	qualcomm monday announced chinese court order banning import sale apple iphone iphone models china due software patent violations court found apple violated two qualcomm software patents around resizing photographs managing applications touchscreen apple however said iphones remain sale china	<START> qualcomm wins import ban apple iphones china <END>	google shares crore film
1	congress appointed year old amit chavda new chief gujarat unit replacing year old bharatsinh solanki held post since december comes days congress president rahul gandhi said younger generations come forward take party leadership inspired gandhi words year old shantaram naik resigned goa congress chief	<START> congress appoints old amit chavda new gujarat head <END>	man man india india
2	male nurse employed delhi institute liver biliary sciences booked allegedly stealing government hospital stents selling private hospitals kerala hospital provided stents subsidised rate man associates allegedly helped fabricate false documents hide tracks	<START> nurse sells stents worth stolen hospital <END>	man man world india
3	finance minister arun jaitley announced government develop scheme assign every major small enterprise india unique presenting budget said scheme along lines aadhaar provided identity every indian eased delivery public services	<START> govt announces aadhaar like unique businesses <END>	govt shares crore crore
4	international organization migration slammed social media giants like facebook failing tackle human trafficking platform smugglers often use facebook reach migrants false promises jobs europe iom spokesperson leonard doyle said traffickers use whatsapp send videos tortured migrants families extortion added	<START> slams facebook failing tackle human trafficking <END>	google shares film film

Fig 12. Model 1 predictions

10.1.1. BLEU and ROUGE score for Model 1:

	Actual Headline	Predicted Headline	BLEU Score
27	[[kerala, govt, challenge, sabarimala, verdict]]	[man, govt, world, world]	0.550695
3	[[govt, announces, aadhaar, like, unique, businesses]]	[govt, shares, crore, crore]	0.428882
5	[[everything, get, triple, talaq, bill, passed, govt]]	[govt, govt, crore, crore]	0.334014
32	[[apple, google, used, car, test, self, driving, tech]]	[google, shares, film, film]	0.26013
22	[[first, ever, hat, trick, cricket, world, cup, taken, indian]]	[man, govt, world, world]	0.202589
36	[[pak, govt, army, want, civilised, ties, india, imran, khan]]	[man, man, world, india]	0.202589

Fig 13. Model 1 BLEU Score for each prediction

```
r1_scores_m1.mean()

0.030952380952380953
```

Fig 14. Model 1 overall ROUGE Score

10.2. Model 2 Predictions:

The predictions generated by Model 2 are more accurate compared to Model 1

	Information	Actual Headline	Predicted Headline
0	qualcomm monday announced chinese court order banning import sale apple iphone iphone models china due software patent violations court found apple violated two qualcomm software patents around resizing photographs managing applications touchscreen apple however said iphones remain sale china	<START> qualcomm wins import ban apple iphones china <END>	qualcomm fixes bug behind samsung rival android devices
1	congress appointed year old amit chavda new chief gujarat unit replacing year old bharatsinh solanki held post since december comes days congress president rahul gandhi said younger generations come forward take party leadership inspired gandhi words year old shantaram naik resigned goa congress chief	<START> congress appoints old amit chavda new gujarat head <END>	congress leader quits congress leader
2	male nurse employed delhi institute liver biliary sciences booked allegedly stealing government hospital stents selling private hospitals kerala hospital provided stents subsidised rate man associates allegedly helped fabricate false documents hide tracks	<START> nurse sells stents worth stolen hospital <END>	mumbai police stations get electric vehicles
3	finance minister arun jaitley announced government develop scheme assign every major small enterprise india unique presenting budget said scheme along lines aadhaar provided identity every indian eased delivery public services	<START> govt announces aadhaar like unique businesses <END>	govt allocates crore gst tax arun jaitley
4	international organization migration slammed social media giants like facebook failing tackle human trafficking platform smugglers often use facebook reach migrants false promises jobs europe iom spokesperson leonard doyle said traffickers use whatsapp send videos tortured migrants families extortion added	<START> slams facebook failing tackle human trafficking <END>	india born embassy made negotiations

Fig 15. Model 2 predictions

10.2.1 BLEU and ROUGE score for Model 2:

	Actual Headline	Predicted Headline	BLEU Score
46	[[metro, cards, swiped, pay, bus, fares, delhi]]	[delhi, metro, launch, metro, fares, due, heavy, rainfall]	0.782542
13	[[rape, case, cong, mla, complained, blackmail]]	[bjp, mla, bawana, ved, arrested, stalking, case]	0.731110
27	[[kerala, govt, challenge, sabarimala, verdict]]	[kerala, temple, bans, trekking, test]	0.668740
47	[[loveratri, demeaning, towards, culture, salman]]	[cannot, take, part, films, salman, khan]	0.638943
23	[[year, old, buried, avalanche, minutes, survives]]	[old, girl, dies, falling, train, delhi]	0.638943
25	[[never, dmk, political, enemy, dhinakaran]]	[dissolve, maun, vajpayee, niece, jaya, mortal, dmk]	0.614788
3	[[govt, announces, aadhaar, like, unique, businesses]]	[govt, allocates, crore, gst, tax, arun, jaitley]	0.614788

Fig 16. Model 2 BLEU Score for each prediction

```
r1_scores_m2.mean()
```

```
0.15572222222222223
```

Fig 17. Model 2 overall ROUGE Score

10.3. Model 3 Predictions:

The predictions generated by Model 3 are accurate compared to Model 1 and Model 2.

	Information	Actual Headline	Predicted Headline
0	qualcomm monday announced chinese court order banning import sale apple iphone iphone models china due software patent violations court found apple violated two qualcomm software patents around resizing photographs managing applications touchscreen apple however said iphones remain sale china	<START> qualcomm wins import ban apple iphones china <END>	qualcomm rejects china ban iphone sales china
1	congress appointed year old amit chavda new chief gujarat unit replacing year old bharatsinh solanki held post since december comes days congress president rahul gandhi said younger generations come forward take party leadership inspired gandhi words year old shantaram naik resigned goa congress chief	<START> congress appoints old amit chavda new gujarat head <END>	congress worker quits party somnath paper
2	male nurse employed delhi institute liver biliary sciences booked allegedly stealing government hospital stents selling private hospitals kerala hospital provided stents subsidised rate man associates allegedly helped fabricate false documents hide tracks	<START> nurse sells stents worth stolen hospital <END>	delhi student alleges molestation hospital
3	finance minister arun jaitley announced government develop scheme assign every major small enterprise india unique presenting budget said scheme along lines aadhaar provided identity every indian eased delivery public services	<START> govt announces aadhaar like unique businesses <END>	govt allocates lakh jobs without size notes

Fig 18. Model 3 Predictions

10.3.1. BLEU and ROUGE score for Model 3:

	Actual Headline	Predicted Headline	BLEU Score
42	[[complaint, diljit, dosanjh, glorifying, dog, fights]]	[complaint, diljit, dosanjh, glorifying, dog, fights]	1.000000
0	[[qualcomm, wins, import, ban, apple, iphones, china]]	[qualcomm, rejects, china, ban, iphone, sales, china]	0.809107
46	[[metro, cards, swiped, pay, bus, fares, delhi]]	[delhi, metro, passengers, get, bus, service, tax]	0.809107
47	[[loveratri, demeaning, towards, culture, salman]]	[producers, salman, khan, produce, film, loveratri]	0.759836
43	[[snapdeal, sued, sellers, non, payment, dues]]	[petition, filed, snapdeal, dues, pnb, scam]	0.759836
25	[[never, dmk, political, enemy, dhinakaran]]	[never, appointed, yes, prakash, ttv, dhinakaran]	0.759836
49	[[mirinda, launches, releasethepressure, campaign]]	[launches, campaign, teach, gay, military, campaign]	0.759836
41	[[notices, issued, noida, housing, complexes, air, pollution]]	[notice, issued, notices, repair, sheets, repair, sheets]	0.731110

Fig 19. Model 3 BLEU Score for each prediction

```
r1_scores_m3.mean()
```

```
0.19702380952380955
```

Fig 20. Model 2 overall ROUGE Score

10.4. Average scores for all models:

Based on the scores in Table 1, it appears that Model 2 and Model 3 have higher BLEU and ROUGE scores compared to Model 1. BLEU score measures the similarity between the predicted text and the reference text, with higher scores indicating better similarity. ROUGE score, on the other hand, measures the overlap of n-grams (n consecutive words) between the predicted and reference text. In this case, Model 2 has the highest BLEU score, indicating that it has the closest similarity to the reference text. Model 3 has a higher ROUGE score, which suggests that it has better overlap of n-grams with the reference text.

	Model	BLEU Score	ROUGE Score
1	Model 2	0.324671	0.155722
2	Model 3	0.319631	0.197024
0	Model 1	0.315809	0.030952

Table 1. Model scores comparison

11. INFERENCES :

11.1 Model 1:

Model 1 was a base model that we used to see where to proceed. The accuracy of the model was not that high and after examining individual BLEU scores there were many predictions that were scored 0. There were also predictions that were able to get the theme and keywords but some of them were completely off topic. The highest similarity scores were still less than 50% There was room for improvement.

11.2. Model 2:

- Increasing the number of epochs helped in this model. After running for 20 epochs the loss value was decreasing constantly and the model was converging
- Adam optimizer performed better than 'rmsprop' optimizer with lower loss values.
- Increasing the number of LSTM layers increased the model size and this was computationally more expensive.
- The BLEU scores increased and the number of predictions that was rated 0 decreased. Individual scores of predictions even reached around 80% in some cases.

11.3. Model 3 :

- Pretrained embeddings improved the overall quality of the predictions.

- The number of individual predictions that was scored 0 was the minimum of all three models.
- There was not much improvement in overall BLEU scores, but the predictions were not so varied. Almost all of the predictions captured the main themes and keywords.
- This model produced the most balanced results compared to the other two models.

12. CHALLENGES:

- The primary challenge we faced was trying to run complex models with limited GPU and resources. This inhibited many of our ideas.
- We tried to implement pretrained models for abstractive summarization such as T5, BART, BART-base, GPT-2 and distilbert. The main issue we ran into was an OOM error that indicated the kernel was out of memory.
- We also tried to implement our model with the standard CNN news dataset which led to a similar setback due to the large number of input.
- We found that the quality of available datasets could be improved as there are not many benchmark datasets with high-quality summaries to test the models on.

13. FUTURE SCOPE:

- With more computational resources the challenges faced with pretrained models/bigger datasets can be overcome.
- There is a possibility our model may overfit the data hence increasing the data through new datasets or data augmentation can increase the performance of the model.
- We have as of now executed news headline generation using shorter versions of the complete article due to computational costs. It can be expanded to feed in entire articles as inputs.
- The accuracy of these predictions has further scope to be improved with better quality of predictions. Transformer based models can be experimented with to see if any better improvements.
- The attention mechanism implemented is a global attention mechanism where the similarity of the decoder state is computed for the entire encoder state. Local attention layer.

- where comparison is made with only a fixed window in the encoder states can be attempted to see if there is improvement in the predictions.

13. CONCLUSION:

In summary, the development of an effective abstractive text summarization model has the potential to revolutionize the way we process and consume information, making it easier to quickly and accurately identify the most important information from large amounts of text. We believe that our proposed model has the potential to contribute to this field and advance the state of the art in abstractive text summarization.

14. REFERENCES:

[1] Dzmitry Bahdanau,KyungHyun Cho Yoshua Bengio 2015. Neural Machine Translation by Jointly Learning to Align and Translate.

[2] Alexander M Rush, Sumit Chopra, and Jason Weston 2015. A neural attention model for abstractive sentence summarization. In Empirical Methods in Natural Language Processing

[3] Jun Suzuki and Masaaki Nagata. 2016. RNN-based encoder-decoder approach with word frequency estimation

[4] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos,C¸ aglar Gulc¸ehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Computational Natural Language Learning.

[5] Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In North American Chapter of the Association for Computational Linguistics

[6] Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) - Network Alex Sherstinsky, 2018

[7] Attention Is All You Need - Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

Project Contribution :

- Megala Jeyapal : Data Proprocessing, Exploratory Data Analysis, Tokenization and Model 1 .
- Yogalakshmi Saravanan : Model 2, Model 3, Attention Mechanism, Evaluation metrics.