News Article Summarization using Deep Learning

Vaidehi Parikh **Shaival Shah**

1 Introduction

The current expanse in digital media and the information available on the web is tremendous. Therefore there is a need for some form of information compression which can be achieved by various mining tasks like classification, clustering and summarization. Vast amount of content on the web is news and news websites are overwhelmed with news articles. According to a study, roughly 6 out of 10 people read only the news headlines. Hence, summarization becomes a crucial task.

Text Summarization is one of the most effective and simplest technique for giving the central idea from large amount of information. It aims to compress the source text into a more concise form with preserving its information content and overall meaning [1]. Manually summarizing a text is quite cumbersome and time-consuming. Hence, our primary idea is to develop an automatic text summarizer for all kinds of news articles on the web. One of the major applications of this is, it would allow organizations to further enrich newsletters with a stream of summaries (versus a list of links), which can be a particularly convenient format in mobile.

There are two different techniques used for this: Extractive Summarization and Abstractive Summarization. Here, we are making use of abstractive summarization in order to generate multi-sentence summaries with the use of Deep Learning techniques.

Our proposed basline model was a recurrent-neural-network (RNN)-based sequence- to-sequence (seq2seq) model with attention mechanism. A bi-directional LSTM cell is used for the encoding along with a uni-directional LSTM for the decoding stage. It is capable of capturing the context from both the directions and results in a better context vector.

This model has several shortcomings, including the inability to handle OOV (out of vocabulary) words. The model also has issues with repetition of words and phrases. Here, we plan to use Transformers to better the results of our baseline model. The main issues with RNN is, it does not provide parallelization and we cannot compute the value of the next timestep, without having the output of the current. The transformers make use of the "self-attention" mechanism - it assigns attention weights to all the words at once, which leads to parallelization. It also makes the task computationally more efficient.

We have used the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [3] scores for evaluating our summaries. They determine the quality of summarization by counting the number of overlapping units between machine-generated and human- written summaries. ROUGE-1(unigram), ROUGE-2(bigram) and ROUGE-L (longest common subsequence) are most widely used for single document abstractive summarization.

2 Related Work

Text summarization has been in existence since 1958. Luhn [2] proposed a heuristic method for text summarization. He proposed that the importance of each word in a document is indicated by its significance. The idea was that any sentence with the highest frequency words (Stopwords) and the fewest occurrences is not more important to the meaning of the document than others.

Historically, most summarization tasks were extractive, involving scanning the source document for significant sentences or passages and reproducing these as summaries [4] [5] [6]. Traditional phrase- table-based machine translation techniques [7], weighted tree-transformation compression

[8], and the quasi-synchronous grammar approach have all been used to tackle the task of abstractive summarization [9].

Deep learning has emerged as a viable alternative for many NLP tasks [10] since 2011, and researchers have begun to consider this framework as an appealing, fully data-driven alternative to abstractive summarization. In 2015, Rush [11] used convolutional models to encode the source and a context sensitive attentional feed-forward neural network to generate the summary. Chopra [12] extended this work by using a similar convolutional model for the encoder but replacing the decoder with an RNN, resulting in improved performance. To address critical problems in abstractive text summarization, Nallapati [13] introduced RNN encoder-decoder architecture. He implemented unidirectional encoder and decoder along with attention mechanism. However, the model is not efficient in generating multi-sentence summaries. To overcome the shortcomings of Nallapati's model, Kamal Al-Sabahi et al.[17] introduced bidirectional encoder-decoder based RNN. One of the advantages of using bidirectional encoder-decoder is at the time of generating summaries, the model's prediction depends on understanding future along with previous one.

A lot of neural machine translation models have been proposed, including RNNencdec, RNNSearch, ConvS2S and Transformer[18]. Out of all these, Transformer model have achieved state-of-the-art performance. It minimizes the path length between long-distance dependencies in the text, which contributes its exceptional performance.

3 Methods

3.1 Dataset

We have used the **CNN/DailyMail** dataset for this project.

The CNN / DailyMail Dataset is an English-language dataset containing just over 300k unique news articles as written by journalists at CNN and the Daily Mail. There are two features:

- article: text of news article, used as the document to be summarized
- highlights: joined text of highlights with <s> and </s> around each highlight, which is the target summary

The CNN/DailyMail dataset has 3 splits: train, validation, and test, with 287,113, 13,368, 11,490 number of instances in the respective split.

3.2 Evaluation Metrics

We have used the Recall-Oriented Understudy for Gisting Evaluation (ROUGE)[4] scores for evaluating our summaries. They determine the quality of summarization by counting the number of overlapping units between machine-generated and human- written summaries.

ROUGE has the following metrics:

- **ROUGE-N**: Overlap of n-grams between the predicted and original summaries.

$$\mathrm{ROUGE-N} = \frac{\sum_{\mathrm{S}\in\{\mathrm{Reference\_Summaries}\}} \sum_{\mathrm{gram_n}\in\mathrm{S}} \mathrm{Count_{match}(gram_n)}}{\sum_{\mathrm{S}\in\{\mathrm{Reference\_Summaries}\}} \sum_{\mathrm{gram_n}\in\mathrm{S}} \mathrm{Count(gram_n)}}$$

  Here, reference summaries are the original summaries of the dataset and n_gram means the sequence of n words.

  - **ROUGE-1** refers to the overlap of unigram (each word) between the predicted and original summaries.

$$\mathrm{ROUGE-1} = \frac{\sum_{\mathrm{S}\in\{\mathrm{Reference\_Summaries}\}} \sum_{\mathrm{gram_1}\in\mathrm{S}} \mathrm{Count_{match}(gram_1)}}{\sum_{\mathrm{S}\in\{\mathrm{Reference\_Summaries}\}} \sum_{\mathrm{gram_1}\in\mathrm{S}} \mathrm{Count(gram_1)}}$$

    Here, reference summaries are the original summaries of the dataset and 1_gram / unigram means the sequence of one word.

– **ROUGE-2** refers to the overlap of bigrams between the predicted and original summaries.

$$\text{ROUGE} - 2 = \frac{\sum_{\text{S} \in \{\text{Reference\_Summaries}\}} \sum_{\text{gram}_2 \in \text{S}} \text{Count}_{\text{match}}(\text{gram}_2)}{\sum_{\text{S} \in \{\text{Reference\_Summaries}\}} \sum_{\text{gram}_2 \in \text{S}} \text{Count}(\text{gram}_2)}$$

Here, reference summaries are the original summaries of the dataset and 2_gram / bigram means the sequence of two words.

- **ROUGE-L**: It is a Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically. Below is the formula to calculate ROUGE-L.

$$\text{ROUGE} - \text{L} = \frac{\sum_{\text{s}_\text{i} \in \text{S}_1} \max \text{s}_\text{j} \in \text{S}_2 \ \text{LCS}(\text{s}_\text{i}, \text{s}_\text{j}) + \sum_{\text{s}_\text{j} \in \text{S}_2} \max \text{s}_\text{i} \in \text{S}_1 \ \text{LCS}(\text{s}_\text{i}, \text{s}_\text{j})}{\sum_{\text{s}_\text{i} \in \text{S}_1} \text{length}(\text{s}_\text{i}) + \sum_{\text{s}_\text{j} \in \text{S}_2} \text{length}(\text{s}_\text{j})}$$

Here, s_i refers to predicted summary and s_j refers to the original summary. LCS() represents the longest common subsequence received from predicted and original summaries.

3.3 Bidirectional LSTM with Custom Attention Mechanism

Deep learning models that take a sequence of items as input and produce another sequence of items as output are known as sequence-to-sequence (Seq2Seq) models. Utilizing Long Short-Term Memory (LSTM) networks, which are a type of recurrent neural network, capable of learning long-term dependencies, is particularly effective in applications where sequences are interdependent, such as text summarization.

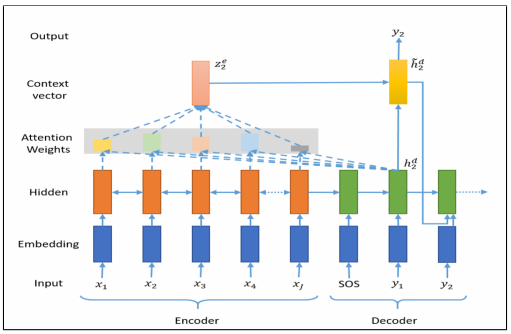

Figure 1: Bidirectional LSTM with attention mechanism

The drawback of LSTM model is that, sometimes it predicts the wrong sequence due to ambiguity. Hence, we have combined the "attention" mechanism along with the existing model. It is based on this exact concept of directing the focus on important factors while predicting the output in Sequence to Sequence models. This way, it helps the encoder in searching most relevant information.

3.4 Transformer Model

A transformer is a deep learning model that uses the self-attention mechanism to weight the relevance (importance) of each element of the input data differently. It is widely used in natural language processing (NLP) and computer vision (CV).

Transformers, like recurrent neural networks (RNNs), are built to handle sequential input data, such as natural language, for tasks like machine translation and text summarization. Transformers, unlike RNNs, do not always process data in the same sequence. The attention mechanism, on the other hand, offers context for any place in the input stream. If the input data is a natural language sentence, for

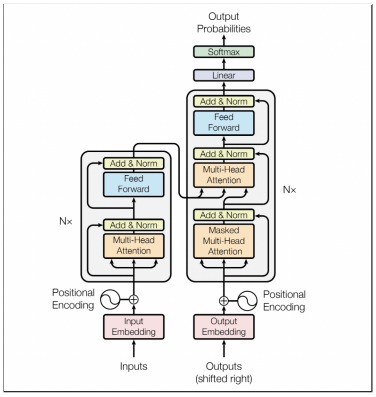

Figure 2: The transformer architecture [18]

example, the transformer does not need to process the beginning of the phrase before the conclusion. Rather, it determines the context that gives each word in the phrase meaning. This feature allows for more parallelization than RNNs, resulting in shorter training durations.

Multi-head Self Attention in transformer decomposes the attention in multiple heads for parallel and independent computations.

$$\text{Multihead}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{W}^0\ [\text{head}_1; \text{head}_2; ....; \text{head}_\mathrm{i}]$$

$$\text{where, } \text{head}_\mathrm{i} = \text{Attention}(\mathrm{W}_\mathrm{i}^\mathrm{Q}\mathrm{Q}, \mathrm{W}_\mathrm{i}^\mathrm{K}\mathrm{K}, \mathrm{W}_\mathrm{i}^\mathrm{V}\mathrm{V})[18]$$

3.5 T5 Model

Text-to-Text Transfer Transformer (T5) is a transformer based architecure which uses text as an input to the model and training it generates some text. The T5 model is based on the concept of transfer learning. The model was first trained on a task with a lot of text available on the **Common Crawl** website before being fine-tuned on a downstream task to learn general-purpose abilities and information that can be applied to tasks like summarization. T5 employs a sequence-to-sequence generation approach, in which the encoded input is fed to the decoder via cross-attention layers and the decoder output is autoregressive. The encoder receives a sequence of tokens that are mapped to a sequence of embeddings as input. In the encoder block, there is a self-attention layer and a feed forward network. The basic structure is similar to the Vanilla Transformer model, with the exception that after each self-attention layer, there is a generalized attention mechanism. This enables the model to just work with the previous outputs. The output of the last decoder block is routed into another layer. The final layer is a thick layer with a SoftMax activation function. The weights from this layer's output are supplied into the embedding matrix's input. It is trained using the "teacher forcing" algorithm which requires both the input sequence and corresponding target sequence.

3.6 Greedy Algorithm for Decoding Summary

Greedy Approach takes the list of potential outputs and the probability distribution that is already calculated, and chooses the option with the highest probability(argmax) and returns the word. It picks most likely token according to the model at each decoding time step t[21].

$$\hat{\mathrm{y}_\mathrm{t}} =_{\mathrm{w}\in\mathcal{V}}\ [\mathrm{P}_\mathrm{r}(\mathrm{y}_\mathrm{t} = \mathrm{w}|\hat{\mathrm{y}}_{<\mathrm{t}}, \mathrm{x}, \phi)]$$

$$\text{where } \mathrm{P}_\mathrm{r}(\mathrm{y}_\mathrm{t} = \mathrm{w}|\hat{\mathrm{y}}_{<\mathrm{t}}, \mathrm{x}, \phi) \text{ is the probability over items in the vocabulary.}[22]$$

The probability of y_t depends on the token seen upto this points as well as the input sequence x.

3.7 Beam Search Algorithm for Decoding Summary

The beam search approach performs the summarization word by word from left to right while maintaining a constant number (beam) of active candidates at each time step. By increasing the beam size, translation performance may be improved although decoder speed is greatly reduced.

4 Experiments and Results

Figure below is the pipeline of our project.

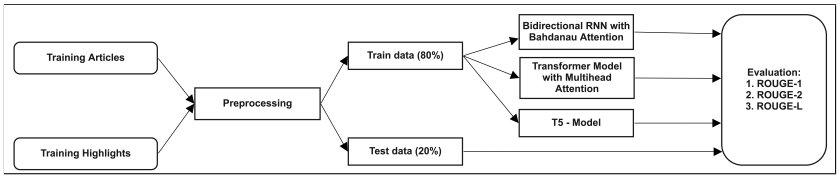

Figure 3: Pipeline: Text Summarization

4.1 Environment

All the implementation is done on Google Colab Pro with HIGH-RAM(27GB CPU), Tesla P100-PCIE-16GB GPU,and Kaggle with 13GB CPU, Tesla P100-PCIE-16GB GPU.

4.2 Preprocessing

This step involves preparing our data for the model. It involved converting text into lower case, removing punctuation, special characters, html links, urls etc. In order to understand the distribution of sentences' length of all articles and highlights, we decided to plot a graph and in order to decide the 'max_length' parameter, we got the percentile values of word count of our articles and highlights data. Hence, we truncated our articles to a maximum length of 1000 words and highlights to a word length of 90. One additional step in **Transformer Model** is, we formed a pipeline with the help of

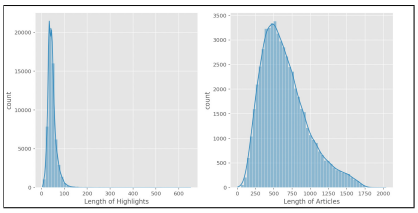

Figure 4: Frequency of articles and summaries

Tensorflow's Dataset API. Herein, we make a batch of data and then randomly shuffle it based on the 'BUFFER_SIZE' which is a hyperparameter. Generally it's value should be more or equal to the total number of datapoints.

4.3 Tokenizing

We make use of keras Tokenizer in order to generate tokens in both baseline and advanced models. After converting text sequences to integer sequences, we pad/truncate our data till the maximum length of article and highlights respectively.

4.4 Embedding

Baseline: To obtain vector representations for words, we used GloVe embedding which is an unsupervised learning algorithm. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space [20]. It is pre-trained on Wikipedia 2014 and Gigaword 5 having 6 Billion tokens, 400k vocab and 50d, 100d, 200d, and 300d vectors. For our baseline, we used 300d vectors.

Transformer: For, our advanced model, we used keras embedding. We converted the input tokens and output tokens to vectors of dimension $d_{model} = 128$. Learned linear transformation and softmax function were used to convert the decoder output to predicted next-token probabilities.

4.5 Train-Test Split

Train-test split for each of the models are mentioned below:

Table 1: Train-Test Split

Model	Train data points	Test data Points
Bidirectional LSTM with Attention	38,797	9700
Transformer	45,938	11,485
T5	22,969	5742

4.6 Hyperparameter Tuning

4.6.1 Bidirectional LSTM RNN with Attention Mechanism

The Architecture contains a single Bidirectional LSTM encoder and a single unidirectional LSTM decoder.

We have implemented Bahdanau attention which is of type additive i.e., linear combination of encoder and decoder. It was proposed to address the performance of standard encoder-decoder architectures. It consist of attention layer, attention weights and context vector. Here, h_t is an input sequence, h_s represents the decoder's previous hidden state, and c_t is a context vector.

$$\text{Attention weights} : \alpha_{ts} = \frac{\exp(\text{score}(h_t, \bar{h}_s))}{\sum_{s'=1}^{S} \exp(\text{score}(h_t, \bar{h}_{s'}))}$$

$$\text{Context vector} : c_t = \sum_s \alpha_{ts} \bar{h}_s$$

$$\text{Attention vector} : a_t = f(c_t, h_t) = \tanh(W_c[c_t; h_t])$$

The latent dimensions are 500, embedding dimensions are 300 and number of hidden units are 500. The total number of trainable parameters for the model are 73,734,000.

The table below represents the optimal hyperparameters received for the model.

Table 2: Baseline hyperparameters

Hyperparameters	Value
Number of Hidden Units	500
LSTM stacks at encoder	1
LSTM stacks at decoder	1
Batch Size	16
Optimizer	RMSProp
Learning Rate	0.001
Loss Function	sparse categorical cross entropy loss
Number of Epochs	5
Total Training Time	10 hours

4.6.2 Transformer Model

One of the Seq2Seq Model's shortcomings is that it cannot manage long-term dependencies. Hence, we implemented a transformer Architecture to mitigate this issue. The transformer's goal is to prevent recursion in order to facilitate parallel computing (reduce training time).

In transformer Model,

The Encoder consist a stack of $N = 4$ identical layers having multi-head self attention and a fully connected network in each with residual connections. All the sub layers and the embedding produces the output of dimension $d_{model} = 128$.

The Decoder also consist a stack of $\mathrm{N} = 4$ identical layers. Along with the sub layers of the encoder, decoder also takes masked multi-head self attention which will ensure that the predictions for position i can depend only on the known outputs at positions less than i.

Multi-head Self Attention: For this model, we employed $\mathrm{h} = 4$ attention heads and $\mathrm{d_k}, \mathrm{d_v}, \text{and } \mathrm{d_{model}} = 128$.

The table below depicts the optimal hyper-parameter retrieved for transformer.

Table 3: Transformer hyperparameters

Hyperparameters	Value
Encoder Layers	4
Decoder Layers	4
Multihead Attention: Number of heads	4
Embedding dim	128
FeedForward Network dimension	512
dropout_rate	0.2
batch_size	16
optimizer	Adam
learning_rate	CustomSchedule(Based on training steps)
Loss function	sparse categorical cross entropy loss
n_epochs	15
Total training time	15 hours

4.6.3 T5 Model

The T5 Model was pretrained using the AdaFactor optimizer. It can be trained or fine-tuned in a supervised or unsupervised fashion. Here, we are feeding the actual highlights and hence we are doing a supervised training of the model. We have implemented **"Beam Search"** decoding while generating the predicted summaries. T5 comes in five different sizes, but we have trained the "t5-base" model using the "T5ForConditionalGeneration", which includes a language modeling head on top of the decoder. We use the "T5FastTokenizer" which is based on the vocabulary created by the sentencepiece file. The "Fast" implementation allows a significant speed-up when using batched tokenization and additional methods to map original character and the token space. We also utilize "batch encoding" which returns the input_ids, token_type_ids and attention_mask as a list for each input sentence.

The table below depicts the hyperparameters used for the T5 training.

Table 4: T5 hyperparameters

Hyperparameters	Value
Repetition Penalty	2.5
Length Penalty	1
Number of beams	4
Train Batch Size	2
Val Batch Size	2
Optimizer	Adam
Learning Rate	1e-4
Loss Function	Negative Log-Likelihood Loss
Number of Epochs	2
Total Training Time	5 hours

Below shown table gives the Results that we obtained after optimal hyper-parameter tuning:

Table 5: Results

	ROUGE-1	ROUGE-2	ROUGE-L
Baseline (Bidirectional LSTM with attention)	26.62	7.26	19.06
Transformer	20.67	1.8	14.3
T5	34.6	12.5	21.5

5 Conclusion and Future Work

We came to the following conclusion after working with Bidirectional LSTM, transformer, and T5 model implementations:

- T5 model outperformed both Transformer and LSTM models. The main reason for this is that it is pre-trained on a very big corpus of data with millions of parameters. It is incredibly computationally costly to train LSTM and transformer with such large amounts of data and parameters.
- When the results of LSTM with attention and Transformer were compared, LSTM with attention outperformed the transformer model. The reason is that due to limitations of compute and time, proper hyperparameter is not done but the parallelism architecture of the transformer significantly decreased training time.
- ROUGE scores for all techniques are average. This is because the ROUGE scores do not accurately reflect content coverage for a variety of reasons: ROUGE does not capture concepts that are synonymous. On a surface level, it compares n-gram overlap of words. As a result, even if the term is semantically accurate, ROUGE will ignore it. PEGASUSLARGE is the current state-of-the-art model for text summarization, with a ROUGE score of 44.17 on cnn/dailymail data. As a result, when compared to PEGASUS, the models performed descent.
- Another explanation for average scores is that the model requires a lot of training, which involves a significant amount of data as well as compute resources. We were unable to adequately train our model on massive amounts of data due to a lack of sufficient computational resources. It would have performed much better if it had been trained on a much larger corpus.

5.1 Future Scope

- In this project, we used word level embedding. We'd like to add sentence level embeddings to the models to improve them even more.
- As ROUGE performs better for extractive summarization, we would like to add pointing copying mechanism which helps in extracting the salient information from the extraction process in the summaries, along with a coverage mechanism which helps the attention mechanism to remember the past alignment information.
- We would also like to explore Diverse Beam Search for decoding summaries.
- The majority of evaluation techniques, such as ROUGE and BERTScore, are insufficient to assess the overall quality of produced summaries. We still need to gain access to essential qualities of created summaries by human specialists, such as factual correctness, fluency, and relevancy. Building improved assessment algorithms that go beyond present metrics to capture the most significant features that agree with people is thus a potential research direction along this line.

References

[1] H. Dave and S. Jaswal, "Multiple Text Document Summarization System using hybrid Summarization technique," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015, pp. 804-808, doi: 10.1109/NGCT.2015.7375231.

[2] H. P. Luhn, "The Automatic Creation of Literature Abstracts," in IBM Journal of Research and Development, vol. 2, no. 2, pp. 159-165, Apr. 1958, doi: 10.1147/rd.22.0159.

[3] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[4] Joel Larocca Neto, Alex Alves Freitas, and Celso A. A. Kaestner. 2002. Automatic text summarization us- ing a machine learning approach. In Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence, pages 205–215.

[5] Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008a. Extractive summarization using supervised and semisupervised learning. In Proceedings of the 22Nd International Conference on Computational Linguistics Volume 1, pages 985–992.

[6] Katja Filippova and Yasemin Altun. 2013. Overcom- ing the lack of parallel data in sentence compression. In Proceedings of the 2013 Conference on Empiri- cal Methods in Natural Language Processing, pages 1481–1491.

[7] Michele Banko, Vibhu O. Mittal, and Michael J Wit- brock. 2000. Headline generation based on statistical translation. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 22:318–325.

[8] Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In Proceedings of the 22Nd International Conference on Computa- tional Linguistics - Volume 1, pages 137–144.

[9] Kristian Woodsend, Yansong Feng, and Mirella Lap- ata. 2010. Title generation with quasi- synchronous grammar. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pages 513–523, Stroudsburg, PA, USA. Association for Computational Linguistics.

[10] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. CoRR, abs/1103.0398.

[11] Alex and er M.Rush,Sumit Chopra,and Jason Weston.2015.A neural attention model for abstractive sentence summarization. CoRR, abs/1509.00685.

[12] Sumit Chopra,Michael Auli,and Alex and er M.Rush.2016.Abstractive sentence summarization with attentive recurrent neural networks. In HLT-NAACL.

[13] Nallapati,R.,Zhou,B.,Santos,C.N.,Gülçehre,Ç.,Xiang,B.(2016).Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. CoNLL.

[14] Abigail See,Peter J.Liu,and Christopher D.Manning.2017.Get to the point:Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 1073– 1083.

[15] Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In HLT-NAACL.

[16] Jiang, Jiawen Zhang, Haiyang Dai, Chenxu Zhao, Q. Feng, Hao Ji, Zhanlin Ganchev, Ivan. (2021). Enhancements of Attention-Based Bidirectional LSTM for Hybrid Automatic Text Summarization. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3110143.

[17] Al-Sabahi, Kamal et al. "Bidirectional Attentional Encoder-Decoder Model and Bidirectional Beam Search for Abstractive Summarization." ArXiv abs/1809.06662 (2018): n. pag.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of NIPS

[19] Zhang, J., Luan, H., Sun, M., Zhai, F., Xu, J., Zhang, M., Liu, Y. (2018). Improving the Transformer Translation Model with Document-Level Context. EMNLP.

[20] https://nlp.stanford.edu/projects/glove/

[21] https://towardsdatascience.com/the-three-decoding-methods-for-nlp-23ca59cb1e9d

[22] https://www.borealisai.com/en/blog/tutorial-6-neural-natural-language-generation-decoding-algorithms/

[23] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. ACL.

[24] Shi, T., Keneshloo, Y., Ramakrishnan, N., Reddy, C.K. (2021). Neural Abstractive Text Summarization with Sequence-to-Sequence Models. ACM Transactions on Data Science, 2, 1 - 37.

[25] Raffel, C., Shazeer, N.M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. ArXiv, abs/1910.10683.

6 Appendix

Github Repo Link:https://github.com/shaival99/News-Summarization-using-Abstractive-Techniques

Here, we are showing our extracted summaries from the 3 models that we implemented in this project:

6.1 Inference Results: Bidirectional LSTM RNN with Attention

Original summary: nikki wright was threatened by pc boulton in row outside bar the incident took place in trinity street stokeontrent staffordshire police suspended the officer after the incident in july but now despite finding he used threatening language he has escaped with just written warning and returned to work today miss wright blasted decision to let him off with slap on the wrist

Predicted summary: the yearold was suspended after he was suspended from work in the early hours of the morning he was suspended after the row at the row of the yearold was suspended after the row at the row of the yearold was suspended after the row at the row of the yearold was suspended after the row at the row of the yearold was suspended after the row at the row of the yearold was suspended after the row at the row of the yearold was suspended after the row

Original summary: saudi arabia is preparing for an onslaught of religious pilgrims the health ministry wants to prevent the spread of the new respiratory disease mers mers kills more than half the people it is known to infect but it is unlikely to become pandemic health officials say

Predicted summary: saudi arabia is offering to cover their aid to millions of muslim worried the problem is caused by the death of the disease the yearold has saudi arabia and saudi arabia the problem is caused by the death of the disease

6.2 Inference Results: Transformer Model

Original summary: Thierry Henry says Arsene Wenger deserves praise for the job he's done . The 37-year-old believes Arsenal fans demand a lot from the Gunners . But he thinks this is down to the success Wenger brought to the club . The Frenchman won the Premier League twice during his time at Arsenal . He believes Wenger's current crop are also capable of winning it .

Predicted summary: arsenal beat arsenal 2 0 in the fa cup on sunday the midfielder has been linked with a move to arsenal in the summer the midfielder has been linked with a move to arsenal in the summer

Original summary: The mother, who cannot be named, pleaded guilty to producing and disseminating child abuse material . Her husband said she should not be imprisoned but instead be allowed back home . She took naked photos of her daughters allegedly at the request of their Sydney dance teacher Grant Davies . Davies ran a studio in Sydney's inner west and trained performers for his productions like Billy Elliot .

Predicted summary: the former pm was accused of having sex with her mother in a young girl he was jailed for four years for four years for sex offences the teenager was jailed for four years for the assault.

6.3 Inference Results: T5 Model

Original summary: New York (CNN) – The World Food Programme is ramping up its Syria operations in anticipation of greater demand from inside and outside the country, but lack of access is proving problematic, the U.N. groupś executive director told reporters Tuesday. "Most of my time has been spent with donors as well as the representatives from the neighboring countries regarding our operation in Syria," Ertharin Cousin said at the Social Good Summit

Predicted summary: U.N. Secretary-General Ertharin Cousin said access to Syria is a serious concern for the United Nations, which has 80 international staffers and 300 national staffers supporting the region. "Itś not an easy task," Cousin said in a statement on CNNś "State of the Union." "We are very concerned that we will not be able to reach our targets as quickly as possible,"

Original summary: (CNN) – A cricketer who punched a rival international player in a bar has been fined and banned ahead of one of the sport's oldest and most renowned competitions. Australia batsman David Warner struck young England counterpart Joe Root in an Australian-themed bar in the English city of Birmingham in the early hours of Monday morning. Warner had to sit out Wednesday's Champions Trophy match with New Zealand, and Cricket Ausrtalia announced Thursday that

Predicted summary: Australian cricketer David Warner has been banned and fined for hitting a rival player in an Australian-theme bar. The batsman, who will be eligible for the forthcoming Ashes series against England, punched England's Joe Root in the back of the head in an Australian-themed bar on Saturday night. Warner was fined $11,000 and banned until July 10 by Cricket Australia. Scroll down for video. David Warner has been banned