Text Summarization in English

Asmita Mukherjee[1] **Dhiraj Jain**[2] **Shashwat Vaibhav**[3]
Department of Computer Science and Engineering
IIIT-D

{ [1]asmita21115@iiitd.ac.in, [2]dhiraj21030@iiitd.ac.in, [3]shashwat21082@iiitd.ac.in }

Abstract

Summarization as a task can be seen in day-to-day life. Automatic text summarization had been explored before the deep-learning era, using extractive methods to find the most relevant information from a collection of sentences. Deep learning has helped to generate abstractive summaries with the sequence-to-sequence paradigm using the encoder and decoder approach. With the arrival of transformers and advanced architectures based on them, abstractive summarization has seen improved performances. As a task, we have combined both the extractive and the abstractive approach in a single pipeline. We found that performing the extractive-then-abstractive method performed par with the abstractive-only summarization. It also helps reduce the overall training time.

1 Introduction

Text summarization has been present in different walks of life. It helps us to get the overall gist concisely and in a shorter period. In the age of an overwhelming amount of information, summarization plays an essential role in getting the knowledge shared across. This is evident from the fact that multiple applications and websites provide news summaries, novel bird-eye-view, scientific paper analysis, and other information in few-minute-reads type of blogs.

Automated text summarization is not a novel task in Natural Language Processing(NLP). The classical approaches, with their limitation in grasping the semantics and capturing long-term dependencies in extensive texts, most of the earlier works majorly focused on Extractive Summarization. Extractive summarization uses several approaches to find the most fruitful sentences from the document. It employs entropy-based filtering, Pointwise Mutual Information, filtering sentences based upon which entails the given document the most, etc., for the purpose.

With deep learning coming into the picture, the RNN-based encode-attend-decode(1) paradigm helped overcome several limitations of the classical methods. But, it still faced the issues of computation resource wastage and lack of parallelization. Transformers (8) revolutionized the field of NLP. It provided long-sought parallelizability and simultaneously encouraged the creation of language-agnostic representation using its novel attention mechanism. Architectures based upon it, such as BERT, BART, GPT, vision transformer, Audio Transformers have awed the world with their skyrocketing advancement in text, vision, and acoustics.

This paved the way for Abstractive Summarization. Now the model had to understand, infer the semantics and then generate the summaries. Abstractive summarization is one of the most challenging tasks in NLP. Some of the other challenges it faces are a need for good gold summaries or the ground truth, flawed evaluation metrics, and an extremely slow reward-based policy search if Reinforcement learning is included.

In this task, we combined the extractive and abstractive techniques. We have used unsupervised learning for extractive summarization based on LexRank(3). Then we used abstractive summarization over the outputs from the previous step. We also performed extractive-only and abstractive-only methods for comparison and analysis.

The dataset used is Indian Language Summarization (ILSUM-2022). There were 10K(10,052) training instances and 2.5K(2,513) test instances. The training data had news articles, their headlines, and annotated summaries. Test data did not have

any gold summary. For our analysis part, we took 2K(2,011) instances from training data for validation of our results but used the whole training dataset for training the model.

In the subsequent sections, we discuss the related works, the methodology used, experimental results, analysis of the results, conclusion, and future work.

2 Related works

In the extractive summarization domain, earlier works rely on rule-based systems. The rules and design features were manually designed by experts having domain knowledge. Some of the earlier abstractive summarization tasks were done based on the coherence of sentences. They mainly modified and combined the extracted phrases and sentences to generate summaries.

Neural network-based summarizers find newer representations of sentences and then find the relevance of those representations for an extractive summary. The vanilla transformer-based models have performed well in beating the earlier baselines for abstractive summaries, but they are computationally much more expensive, both in terms of time and space. The advanced architectures based on transformers have performed better and are mostly employed for this task.

PageRank is a standard algorithm for identifying essential web pages. It maps the whole web as a graph and finds the most influential nodes. LexRank, a successor to PageRank, is a graph-based method for summarizing text. It is an unsupervised paradigm that scores sentences based on the eigenvector centrality in a graph representation of sentences. It incorporates similarity metrics. The approach proposed in LexRank is insensitive to the noise present in input data arising from the imperfect clustering of sentences.

Bidirectional Encoder Representation from Transformers(BERT)(2) is based upon transformers. It used 12 layers of stacked encoders. It is designed to read the textual sequence in both directions simultaneously, thus giving it an edge over previous models in capturing the context. It performs two tasks, Masked Language Modelling(MLM) and Next Sentence Prediction(NSP). The embedding representation generated from BERT can be used to select sentences for summarization based on some classifier.

A bidirectional and AutoRegressive Transformer(BART)(5) is a denoising autoencoder for pretraining sequence-to-sequence models. It uses BERT encoder for its direction agnosticism or bidirectionality and Generative PreTraining(GPT) decoder for its autoregressive feature. It is fine-tuned on summarization datasets and currently gives state-of-the-art results.

Huang et al.(4) have explored recent trends of summarization and evaluation based on various aspects. They have compared different strategies of abstractive and extractive summarization. They also analyzed the performances based on standard yardsticks.

3 Methodology

A good summary is able to represent the source or the given document in a concise and coherent manner. Hence for summarization both semantics and syntax are of the essence. As task summarization is a challenging one even when looking at it from the human perspective, one needs to comprehend the text and also be able to represent it using one's own words. More challenges arise when one is aiming to achieve the results from a mathematical model. Summarization as a task can be broken down to the following steps:
1. Read the text
2. Break it down into sections
3. Identify the key points in each section
4. Write the summary
5. Check the summary against the article

Data preprocessing was an important step in our dataset. Since the data is majorly collected by web scraping from news reporting sources, it often had noisy characters such as Unicode, and other special characters which got incorporated into the data depending on the source from which it was collected. There were machine-generated code sequences that got added to the data due to the news collection from sources containing video data. We used regex rules to filter our data and make it into the correct format.

In order to achieve our results, we first looked into **unsupervised extractive summarization**.

This would help us to set the baseline for our models and would help us identify the keywords in our document. For our extractive summarization method, we used **LexRank**, which visualizes our text as a graph and then proceeds to identify the centroid sentences in our document. It consisted of the below steps:
1) Represent the text as a graph edge weight matrix
2) Rank the sentences using the LexRank algorithm
3) Return the top five ranked sentences as the summary of the document.

We consider each sentence as node or vertex V in our graph, and our text is visualized as a completely connected graph, where there is N number of sentences and thus N^2 number of edges between them. In order to give weights to each of the edges, we calculated the cosine similarity between their embeddings of them. The sentences were embedded using the pre-trained model "MpNet". The pre-trained model has been evaluated to perform very well in sentence similarity tasks and thus was our ideal candidate. It uses the objective of masked and permutated language modeling in its pretraining. The importance of each node i.e sentence in our case is determined by the eigenvector centrality and the via "power iteration", we calculate the eigenvector centrality score of the graph, till it converges. The power iteration increases the importance of a node, as the importance of its neighbor increases. Hence once we have completely ranked our sentences, we return the top-most ranked sentence as our summary.
On qualitative evaluation, we observed even though our above model performs quite well, the extractive summary is not able to capture the entire information that is present in our document since no one sentence in our entire doc can be said to capture the entire information. Hence we looked into the abstractive methodology. Summarization can be modeled as a sequence-to-sequence task, and hence architectures for standard seq2seq tasks were evaluated. The LSTM-based encoder-decoder architecture of three stacks of encoder-decoder resulted in the model only being able to capture the frequently occurring stopwords. Considering the long-range dependencies that our model needed to capture, this relatively shallow architecture failed to learn properly.

Hence we felt the need for a pre-trained model, which would give us a good starting point from which we could fine-tune our dataset. **BART** has been evaluated to be able to capture long-range dependencies and is able to perform superior in the task of summarization across multiple domains. Choosing BART as our candidate, we fine-tuned the model firstly on the pre-processed source articles. The model started converging only after one epoch, we unfroze the 1 layer of the encoder of the model for it to fit our data properly. The model took around one hour to train on GPU.

We wanted to evaluate the performance of the model on the extractive summaries that we generated in the first step. We observed that our model was able to find the keywords and most important sentences in the first step itself. Hence we took the top five sentences as output by the extractive methodology and used it as the source when fine-tuning our BART model. We wanted to evaluate if feeding the model the crux and removing the noise yielded any improvement in performance. However, the rouge scores, as evaluated on the validation set, did not increase when following the **extractive-abstractive** method, even though its results were highly comparable when compared to fine-tuning with the entire document. The training time for this case was reduced by half. Hence this method achieved both computational efficiency and highly comparable results indicating that it is only a few sentences in the document that finally represent the document as a whole.

4 Experimental Results

Please find the experimental results in 1. The rouge scores are averaged over the entire validation set. The validation set was achieved over the train set by doing an 80:20 split.

Model	Rouge-1	Rouge-2	Rouge-1
MPNet sen scoring + LexRank	'r': 0.36357489067363125, 'p': 0.3650794271125573, 'f': 0.3519916067098555	'r': 0.2264376679528056, 'p': 0.22039065052152135, 'f': 0.21575764710876053	'r': 0.32208910265328494,\ 'p': 0.3244935965421752, 'f': 0.31251766195903224
BART Abstractive↑	'r': 0.530222271531095, 'p': 0.49849008912796644, 'f': 0.5053129757796015	'r': 0.4308130044096971, 'p': 0.39229329392008166, 'f': 0.4030298755760389	'r': 0.5031064130723554, 'p': 0.47018622252150766, 'f': 0.4781776484045712
LexRank + BART Abstractive	'r': 0.5224733380235832, 'p': 0.4502607454399228, 'f': 0.4742957474076542	'r': 0.4077329619789426, 'p': 0.3424317750234633, 'f': 0.36427123995353083	'r': 0.488887019506377, 'p': 0.42047470624803623, 'f': 0.4435227590198014

Table 1: Results

5 Analysis

On dataset analysis, we realized that the summarization task was one of high compression, i.e., given the source document of large length, we had to produce a very short summary.

As seen in 1, most of our source document lies below the token length of 2000 while most of the generated summary lies below the token length of 75. This indicates the challenge that our proposed model must be able to capture long-range dependencies and result in a comprehensive summary.

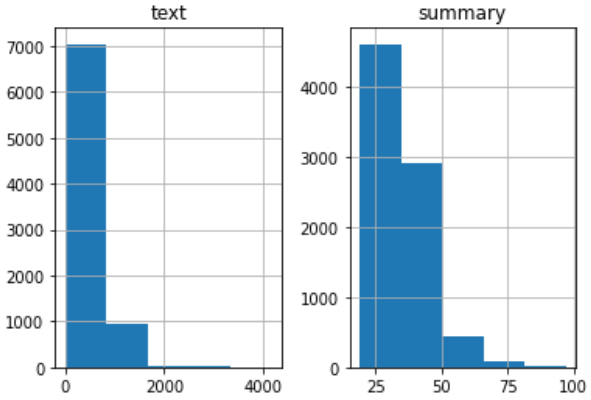

Figure 1: Source and target length

Below we have analyzed some of the results of our model:

The results from our extractive summary showed that ranking sentences as per their lexical similarity scores it able to capture the crux of the document. As seen in 2 the top two sentences returned by the lexrank are able to capture the information as present in the gold summary. However, we also realize the limitation of the extractive summary, which it includes unnecessary details, while the gold summary is concise.

Gold Summary :

A six-week-old baby died after getting infected with coronavirus in Connecticut, the United States on Wednesday. The information was shared by Connecticut Governor Ned Lamont who confirmed the death of the child on Twitter saying the 6-week-old newborn from the Hartford area became another victim of the deadly virus.

Top 5 sentences from LexRank :

6-week-old baby dies of covid-19 in Connecticut, united states. lamont said it is with heart-breaking sadness today that we can confirm the first pediatric fatality in connecticut linked to covid-19. we believe this is one of the youngest lives lost anywhere due to complications relating to covid-19, connecticut governor ned lamont said. the information was shared by connecticut governor ned lamont who confirmed the death of the child on twitter saying the 6-week-old newborn from the hartford area became another victim of the deadly virus. a six-week-old baby died after getting infected with coronavirus in connecticut, the united states on wednesday

Figure 2: LexRank results

Our abstractive model yields meaningful results only after training for one epoch, as shown in 3. The model is able to capture the factual information from the text very well and the summary generated is highly abstractive. The model is able to represent the information as per its own understanding and langugage.

Gold Summary :

Pakistan termed the Indian action as "unilateral and illegal", and said it will take the matter to the UN Security Council. "The Secretary-General has been following the situation in Jammu and Kashmir with concern and makes an appeal for maximum restraint,"

Predicted Summary :

'UN chief Antonio Guterres on Thursday urged India and Pakistan to exercise "maximum restraint" and refrain from taking steps that could affect the status of Jammu and Kashmir, as he highlighted the Simla Agreement which rejects any third'

Figure 3: Abstractive results

Our extractive-abstractive model shows that the BART is able to capture the factual information well, however, there is some loss of information as seen in 4. We can also observe that our abstractive model is able to gather information from multiple sentence and able to represent it in a single sentence in a coherent manner.

Gold Summary :

Dengue outbreak has claimed 66 lives and affected over 44,000 people in Pakistan this year. According to a document available with Dawn, this year 44,415 cases of dengue have been confirmed. Of this, 12,433 cases were reported from Islamabad, 10,142 from Sindh, 9,260 from Punjab, 7,346 from Khyber Pakhtunkhwa and 3,051 from Balochistan.

Predicted Summary :

The dengue has claimed 66 lives and affected over 44,000 people in Pakistan this year. According to a document available with Dawn, this year 44,415 cases of dengu have been confirmed.

Figure 4: Extractive Abstractive results

6 Conclusion and Future Scope

The task of automatic text summarization has been challenging especially given that abstractive summarization is in demand. The roadblocks faced by such challenges could be overcome by combining both the extractive and abstractive approaches. In this work, we merged the task of unsupervised learning-based extractive summarization and supervised learning-based abstractive summarization. The results from the extractive-abstractive method gave us par performance with the abstractive-only approach. The former also reduced the training and inference time to much extent.

Summarization is tricky because it has to capture the semantics at a broader as well as at a fine-grained level simultaneously. Sometimes the

semantics are not inherent in the literal meaning, such as in the case of a figure of speech(metaphor, irony, etc.) and allegorical references. In those cases, reading between the lines becomes difficult. Hence summarization can also be combined with an implicature recovery task, where the hidden meaning behind a statement is recovered. Summarization, as it can understand the hidden implications, can also be used to detect the presence of irony, humor, sarcasm, and intent.

Contributions

- Asmita: Literature survey, suggesting combining extractive and abstractive summarization, data preprocessing, and implementing extractive and abstractive models based on LexRank and BART, results analysis, de-bugging code.
- Dhiraj: Literature Survey, Data Analysis, transformer and LSTM-based seq-to-seq model, results analysis, debugging code, evaluating the models, hyperparameter-tuning.
- Shashwat: Literature survey, data preprocessing, incorporating Multi-arm bandit to improve vanilla LSTM model, implementing abstractive summarization, de-bugging code.

References

[1] BAHDANAU, D., CHO, K., AND BENGIO, Y. Neural Machine Translation by Jointly Learning to Align and Translate, May 2016. Number: arXiv:1409.0473 arXiv:1409.0473 [cs, stat].

[2] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. Number: arXiv:1810.04805 arXiv:1810.04805 [cs].

[3] ERKAN, G., AND RADEV, D. R. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research 22* (Dec. 2004), 457–479.

[4] HUANG, D., CUI, L., YANG, S., BAO, G., WANG, K., XIE, J., AND ZHANG, Y. What Have We Achieved on Text Summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, 2020), Association for Computational Linguistics, pp. 446–469.

[5] LEWIS, M., LIU, Y., GOYAL, N., GHAZVININEJAD, M., MOHAMED, A., LEVY, O., STOYANOV, V., AND ZETTLEMOYER, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, Oct. 2019. Number: arXiv:1910.13461 arXiv:1910.13461 [cs, stat].

[6] LIU, Y., AND LAPATA, M. Text Summarization with Pretrained Encoders, Sept. 2019. Number: arXiv:1908.08345 arXiv:1908.08345 [cs].

[7] RADFORD, A., NARASIMHAN, K., SALIMANS, T., AND SUTSKEVER, I. Improving Language Understanding by Generative Pre-Training. 12.

[8] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention Is All You Need, Dec. 2017. Number: arXiv:1706.03762 arXiv:1706.03762 [cs].