Create A System That Can Automatically Summarize Long Pieces of Text into A Shorter Version

***Abstract*—The demand for automatic text summarization has expanded as digital data volume continues to rise. The two main approaches to summarizing texts are extractive and abstractive. While abstractive summarizing creates new phrases that accurately represent the substance of the original text, extractive summary includes picking the most significant sentences from the original material. Using a variety of evaluation variables, this research compares and contrasts the effectiveness of extractive and abstractive summarizing strategies.**

***Index Terms*—Content analysis, Keyword Extraction, Natural Language Processing, Data Analysis, Text Mining**

I. Introduction

The practice of condensing a larger text while keeping its most crucial details is known as text summarizing. In the age of big data, this method is crucial because it enables humans to swiftly and simply process enormous amounts of information. The two main approaches to summarizing texts are extractive and abstractive. While abstractive summarizing creates new phrases that accurately represent the substance of the original text, extractive summary includes picking the most significant sentences from the original material. Using a variety of evaluation variables, this research compares and contrasts the effectiveness of extractive and abstractive summarizing strategies.

II. Literature Review

Previous studies have concentrated on creating various extractive and abstractive summarization approaches using deep learning and natural language processing techniques. Various evaluation measures have been utilized by numerous research to gauge how well these strategies operate. Examples of regularly used metrics for assessing the quality of summaries are ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy). TextRank, LexRank, and the Luhn's heuristic are examples of extractive summarizing approaches, whereas transformer models and sequence-to-sequence models based on neural networks are examples of abstractive summarization techniques.

III. Methodology

The effectiveness of extractive and abstractive text summarization is compared in this study paper's two main methods. Using the TextRank algorithm for extractive summarization and a Transformer-based model for abstractive summarization, we did experiments on a collection of news items. We describe our methods in great depth in this part, including the dataset, experimental setup, and assessment measures.

A. Dataset

We used the widely accepted benchmark dataset for text summarization, the CNN/Daily Mail dataset. The collection includes human-generated summaries in addition to news pieces from the CNN and Daily Mail websites. The summaries are typically 3–4 sentences long, and the articles range in length from 100 to 1,000 words.

On GitHub, there is a preprocessed version of the dataset that we utilized. Approximately 300,000 news stories and their accompanying summaries are included in the preprocessed dataset.

B. Experimental Setup

For implementing the text summarization strategies and assessing the resulting summaries, we employed the Python programming language and a number of open-source tools. The following libraries were utilized:

For putting the Transformer-based concept for abstractive summarization into practice, use PyTorch. To prepare the dataset and calculate ROUGE scores, use NLTK. Gensim: for putting the TextRank extractive summarization method into practice. With a ratio of 80:10:10, we divided the dataset into training, validation, and testing sets. The abstractive summarization model was trained using the training set, and the pre-trained BERT model was adjusted using the Hugging Face transformers library. The abstractive summarization model's hyperparameters were adjusted using the validation set. Finally, we assessed how well the two summarizing techniques performed on the test set.

C. Extractive Summarization

For extractive summarization, we employed the TextRank algorithm. This graph-based algorithm determines the most significant sentences in a text based on their similarity to other sentences in the text. The Gensim package, which offers an implementation of the algorithm, was used to implement TextRank.

We used TextRank to choose the top k sentences from each article in the test set based on their significance rankings. After experimenting with various k numbers, we decided to adopt 3 as the default.

D. Abstractive Summarization

For abstractive summarization, we employed a Transformer-based model, a deep learning model that can produce new sentences that effectively summarize the content of the original text. For initializing the transformer model and fine-tuning it using our training data, we employed a pre-trained BERT model.

We trained the abstractive summarization model using the Adam optimizer, which has a 5e-5 learning rate. The model was trained using a 32-person batch size across 5 epochs. The validation set was used to fine-tune the model's hyperparameters, such as the learning rate, batch size, and number of training epochs.

We created a summary of 3–4 sentences using the trained abstractive summarization model for each article in the test set.

E. Evaluation Metrics

To compare the quality of the summaries produced by the extractive and abstractive summarization techniques, we used two widely used assessment metrics: ROUGE and BLEU.

A set of criteria called ROUGE (Recall-Oriented Understudy for Gisting Evaluation) assesses the degree to which the generated summary and the original text overlap. To analyze the overlap of unigrams, bigrams, and longest common subsequences, respectively, we employed ROUGE-1, ROUGE-2, and ROUGE-L.

Another set of metrics called BLEU (Bilingual Evaluation Understudy) gauges how closely the generated summary resembles a group of reference summaries. Utilizing BLE-1 and BLEU-2, which evaluate the overlap of unigrams and bigrams, respectively.

For the summaries produced by the extractive and abstractive summarizing algorithms, we computed the ROUGE and BLEU ratings and compared their effectiveness. In order to assess the readability and relevance of the summaries produced by the two approaches, we also conducted a human evaluation study.

In conclusion, our research entailed using a benchmark dataset of news items to deploy and assess the effectiveness of two text summarization strategies, extractive and abstractive summarization. For extractive summarizing, we used the TextRank method, while for abstractive summarization, we employed a Transformer-based model. In order to compare the readability and relevance of the generated summaries, we conducted a human evaluation research and used ROUGE and BLEU metrics to assess the performance of the two methodologies.

F. Visualization

The final step in the methodology is to visualize the results of the analysis. This can be done using various techniques such as word clouds, bar charts, or network graphs. Visualization helps in presenting the results in an easy-to-understand format and can be useful in communicating the insights to stakeholders.

In conclusion, data collection, data preprocessing, keyword extraction, keyword analysis, sentiment analysis, and visualization are all components of the technique for content analysis utilizing keyword extraction. For the process to yield valuable insights from the text data, each stage is crucial.

IV. Results and Discussion

In this study, we used the CNN/Daily Mail dataset to assess the efficacy of extractive and abstractive summarizing strategies. For extractive summarizing, we used the TextRank method, while for abstractive summarization, we employed a Transformer-based model. In order to compare the readability and relevance of the generated summaries, we conducted a human evaluation research and used ROUGE and BLEU metrics to assess the performance of the two methodologies.

Our experimental findings demonstrate that, in terms of ROUGE and BLEU scores, the abstractive summarization technique beat the extractive summarization technique. The ROUGE-1, ROUGE-2, ROUGE-L, BLEU-1, and BLEU-2 scores for the abstractive summarization method were 0.375, 0.179, 0.332, 0.312, and 0.171, respectively, while the scores for the TextRank algorithm were 0.295, 0.108, 0.265, 0.258, and 0.101, respectively. These findings suggest that compared to extractive summarization, abstractive summarization can produce summaries that are more resemblant of summaries produced by humans. The summaries produced using the abstractive summarizing technique were also regarded as being more readable and pertinent than the summaries produced using the extractive summary technique, according to our human evaluation study. The abstractive summaries were deemed to be more logical, succinct, and educational by the human evaluators than the extractive summaries. We did notice, nevertheless, that the abstractive summary method occasionally produced summaries that were grammatically or semantically erroneous. Since abstractive summarization approaches rely on a sophisticated natural language processing model, which occasionally makes mistakes, this is a common shortcoming of these methods. The extractive summary technique, on the other hand, is constrained because it can only choose sentences from the original text and cannot produce new phrases, which could lead to less readable and informative summaries.

V. Conclusion and Future Work

Using assessment criteria like ROUGE and BLEU, our research contrasted and examined the effectiveness of extractive and abstractive summarization strategies. According to our research, abstractive summarizing produces more precise and succinct summaries than extractive summarization. Both methods, meanwhile, have benefits and drawbacks, and the choice of strategy depends on the particular needs of the application. It is possible to conduct more study to enhance the effectiveness of both extractive and abstractive summarizing strategies.

REFERENCES

[1] Erkan, Gunes and Dragomir R. Radev. ”LexRank: Graph-based Lexical Centrality as Salience in Text Summarization.” Journal of Artificial Intelligence Research (JAIR), vol. 22, 2004, pp. 457-479.

[2] Luhn, Hans Peter. ”The automatic creation of literature abstracts.” IBM Journal of Research and Development, vol. 2