A Comprehensive Text Summarization System Combining Abstractive and Extractive Approach

Ashu Soni
Apex Institute of Technology (CSE)
Chandigarh University
India
ashusoni.20010121@gmail.com

Pulkit Dwivedi
Apex Institute of Technology (CSE)
Chandigarh University
India
pdwivedi1990@gmail.com

***Abstract*—A text summarizer is an automated system or tool that condenses a longer piece of text into a shorter, coherent summary while retaining its key information and main points. The purpose of a text summarizer is to save time and effort by providing a concise overview of a text, allowing users to quickly grasp the main ideas without having to read the entire document. This work presents a novel system for generating summaries of both reviews and raw textual paragraphs. The system combines abstractive and extractive text summarization techniques to create coherent and informative summaries. The abstractive approach utilizes an encoder-decoder architecture to paraphrase and rephrase the content, generating concise summaries for reviews. Meanwhile, the extractive approach employs cosine similarity to select and rearrange relevant sentences from raw textual paragraphs, forming the summary. The efficacy of the system is evaluated using the widely-used ROUGE metric, which measures the quality of summaries. The results demonstrate that the system is successful in producing precise and informative summaries for both types of textual content. This research contributes to the field of text summarization by offering a comprehensive solution for summarizing diverse textual materials, thereby facilitating easier information retrieval and decision-making processes.**

***Index Terms*—Text summarizer, Deep Learning, encoder-decoder architecture, ROUGE metric, natural language processing**

I. INTRODUCTION.

Automatic Text Summarization is a system of sophisticated Neural Networks that utilize techniques to enable computers to comprehend and process vast amounts of textual information. The system then generates a summary of the information.

The importance of text summarization is growing fast as a consequence of the massive amount of text-based content available on the internet and in various archives such as news stories, scientific papers, legal documents, and so on. Creating a summary manually can be time-consuming, requires a lot of effort and cost, and can become unfeasible when dealing with enormous amounts of text. Text summarization, on the other hand, condenses a vast amount of text into a shorter version that includes all the crucial information from the original document. It focuses on extracting important sentences while retaining essential information. This task is crucial in various applications where the availability of time or resources is limited, such as news articles, legal documents, or scientific papers. Websites, blogs, articles, news, product reviews, social media, and various other digital resources are enormous sources of written content on the internet.

If we were asked to develop a machine learning model that uses textual data to get insights for solving a specific problem, it would take us a long time to reach any conclusion because most of the time will be consumed in data cleaning processes. Therefore, here comes the need for a system that automatically generates the summary of textual data. An ATS (Automatic Text Summarization) system's primary purpose is to provide a summary that condenses the key ideas from the input information while minimizing repetition while occupying as little space as possible. An automated summary can be defined as a concise and condensed version of the original information that captures the most significant aspects relevant to a specific user or task. The objective is to extract and present the essential information from one or more sources in an abridged format, allowing the reader to grasp the main points quickly and efficiently.In summary, an effective automated summary provides a concise but thorough review of the main facts in a way that is valuable and relevant to the intended audience and their objectives.

Since the 1950s, scholars have been striving to enhance ATS (Automatic Text Summarization) methodologies. Deep learning has revolutionized the area of text summarization, and numerous strategies may be utilized in order to achieve this goal. ATS (Automatic Text Summarization) approaches are divided into extractive, abstractive, and hybrid. Recurrent neural networks (RNNs), which include Long Short-Term Memory (LSTM) models, are a popular approach. These models are capable of capturing the sequential dependencies in the text and producing a summary that maintains the critical information. Text summarising systems can be broken down into two categories: single-document summarization systems and multi-document summarization systems. Single-document summarising systems produce summaries of individual documents, and multi-document summary systems generate summaries of clusters of related documents. These systems use one of three methods of summarization: extractive, abstractive, or hybrid. Because of the difficulty in recognizing the context and significance of the original text, text summarization is a complicated task in the domains of Natural Language

Processing (NLP) and Artificial Intelligence (AI).

"Extractive Summarization" is the procedure of extracting the most significant lines or expressions from a source text and arranging them in a logical manner to form a summary. "Abstractive Summarization," on the other hand, attempts to offer a summary that isn't an exact replica of the original text. Instead, it develops a fresh summary that more concisely and logically encapsulates the relevant information from the source content. The "Hybrid Summarization" strategy combines extractive and abstractive summarising techniques to provide a summary. Initially, the system uses extractive summarising techniques to extract the most relevant sentences or phrases from the original material. It then uses abstractive summarising techniques to paraphrase and organize the extracted sentences into a succinct and cohesive summary. The goal of this method is to maximise the benefits of each technique while minimizing their individual limitations.

ATS has various domain-specific use cases. Text summarising in the news sector, for example, entails extracting the most significant information from a news piece and presenting it in a succinct summary. In the finance domain, ATS can be used to summarize financial reports, market trends, and investment opportunities. This can help investors make informed decisions by providing them with a quick overview of important financial information. In the legal domain, ATS can be used to summarize legal documents such as court opinions, contracts, and legislation. This can help lawyers and legal professionals quickly identify the key information in a document without having to read through all of its contents.

II. Related Work

In the past, the majority of work in text summarization was done using the extractive approach. However, unlike machines, humans often rephrase a story using their own words, resulting in more abstract summaries that rarely include exact sentences from the original document. This created a need for machines to do abstractive summarization to reduce human effort.

With the advent of deep learning in the field of Natural Language Processing (NLP), academics have started seeing it as a possibility to generate abstractive summaries [17] [18] [19]. Abstractive summarization (ABS) is generally considered to be more advanced and effective than extractive summarization because ABS can compress the text into a shorter summary by understanding the context of the text and generating new sentences that capture the main ideas. Abstractive summarization is better suited to handle new or unfamiliar information that may not be present in the original text. It has been shown to outperform extractive summarization in terms of summary quality, as measured by various metrics such as ROUGE which refers to "Recall-Oriented Understudy for Gisting Evaluation."

In a study conducted by Zhang et al. [1] on Abstractive Summarization using deep learning methods, they demonstrated how to employ deep neural networks (DNN), recurrent neural networks (RNN), and convolutional neural networks (CNN) for conducting ABS.

Alhojely et al. [2] explained all the possible ways to generate a summary from raw textual data and the different methodologies for performing extractive, abstractive, and hybrid summarization.Several approaches, including machine learning, deep learning, natural language processing, and statistical methods, are applied.

Nallapati et al. [3] explains an interesting methodology for ATS using Sequence-to-sequence RNNs they used the Sequence-to-sequence technique for generating the summarization results, the use of encoders and decoders with attention layers, and a large vocabulary of words have been used. The baseline model employed by the researchers for abstractive summarization is based on a neural machine translation model.This model consists of an encoder that utilizes a bidirectional GRU-RNN and a decoder that uses a unidirectional GRU-RNN having the same hidden-state size as the encoder. To generate words for the summary, the decoder utilizes an attention mechanism over the source-hidden states and a soft-max layer over the target vocabulary. Deep learning has emerged as a potent approach for various NLP tasks, including abstractive summarization. Different deep learning architectures, such as convolutional models and RNNs, have been tried by researchers to encode and decode text for summarization purposes. Widyassari et al. [4] proposed a total of 9 review questions in their research paper in the context of text summarization, with the goal to create a review process more focused and coordinated. These are the question that must be answered while performing any ATS task. Rafea. et al [5] in the paper has explained the types of summarization techniques and they introduced the general architecture of an ATS system i.e. pre-processing, processing, and post-processing. Mathi et al. [6] proposed a technique for summarising product reviews so that customers don't have to devote a lot of time reading about the products. Their approach was simple yet efficient, and they utilized 9 text summarization algorithms (LexRank, LUHN, LSA, TextRank, Edmundson, SumBasic, KL, Reduction, Pagerank) followed by used cosine similarity, ROUGE, and BLEU scores as evaluation metrics to compare the final outcomes [15] [16]. In another research paper proposed by Bani-Almarjeh et al. [7] they used an RNN (recurrent neural network) and transformer-based architecture to generate summaries for the Arabic language, and they also compared the performance of different pre-trained models. They used the ROUGE evaluation matrix for their project.

In another paper proposed by Ertam et al. [8] they have trained their model for the Turkish language using text-to-text transformers.

As Text summarization can also be done for multiple documents here is another research paper by Veningston et al. [9] representing the same idea, they have proposed The study titled "Personalised Multi-document Text Summarization Using Deep Learning Techniques" used Long Short Term Memory (LSTM) models and Recurrent Neural Network (RNN) models for extractive summarization.

Text Summarizations can also be done using hybrid systems for example the methodology "HNTSumm: Hybrid text summarization of transliterated news articles" published by Sabarmathi et al. [10] was about hybrid systems using the TextRank algorithm for extractive summary and a hybrid seq2seq framework for abstractive summaries, a bidirectional LSTM (Long Short-Term Memory) to feed encoders.

Bilingual word embedding (BWE) is the method of transferring word representations from one language to another, which must be performed for cross-linguistic transfer learning. Wijayanti et al. citeWijayanti utilized a specific technique in their study that started with pre-processing the corpus by deleting non-alphanumeric characters and transforming them to lowercase. They then used SkipGram citeSkipGram to construct word embeddings for each language as input for the VecMap technique. BiVec, on the other hand, may train parallel corpora immediately, with or without an alignment procedure.

NLP has its uses in various filed one such field is behavioral biology. Behavioral biology is a field that studies the behavior of animals, including their interactions with each other and their environment. Text summarization in this field involves condensing lengthy articles, reports, or studies into shorter, more concise versions that capture the essential information and key findings. In their research, Regunathan et al. [14]A hybrid technique combining a seq2seq encoder-decoder model with a stacked LSTMciteLSTM layer was proposed by the researchers. A system for attention with a T5 transformer model pre-processor are also included in their model. T5LSTM-RNN is the hybrid model that was used to generate the summarised data. The ability to summarize ancient texts written in native languages such as Hindi and Sanskrit can be of immense value. These languages are repositories of rich cultural heritage and accumulated wisdom, spanning several millennia. By summarizing these texts, we can distill the essence of their knowledge and insights, and make them accessible to a wider audience. Pimpalshende et al. [21] had built a summarizer for the hindi language. They have used Binary Particle Swarm Optimization (BPSO).

Overall, deep learning-based approaches have shown promise for abstractive summarization tasks, and researchers continue to explore new architectures and techniques to improve their performance

III. Proposed Methodology

The proposed model uses an extractive as well as an abstractive approach for text summarization.

The method of extractive text summarising is picking a selection of phrases from a larger source that contains the major concepts and important information. The goal is to create a summary that is shorter in length than the original document but still conveys the main points. This methodology outlines the steps involved in extractive text summarization using a cosine similarity matrix and raw textual data for training and testing purposes, as well as algorithms for sentence ranking. Following are the NLP steps required for text summarization in general:

- **Data Collection and Preprocessing:** The first step in any text summarization process is to collect the relevant data. This can be done by scraping web pages, extracting content from articles, or collecting data from a database. Once the data is collected, it needs to be preprocessed. The preprocessing step involves removing stop words, punctuations, and other irrelevant characters.
- **Sentence Tokenization:** The preprocessed text is then split into individual sentences. This can be done using sentence tokenization techniques.
- **Cosine Similarity Matrix:** A cosine similarity matrix is then created to compare each sentence in the document to every other sentence. The cosine similarity measure is a popular technique used in text mining to measure the similarity between two sentences. It ranges from 0 to 1, with 1 indicating complete similarity.
- **Sentence Scoring:** The next step is to score each sentence in the document based on its resemblance to the other sentences. This library uses the Levenshtein distance algorithm to measure the similarity between two sentences.
- **Summary Creation:** After ranking the sentences, the top ones are chosen to generate a summary. The summary's length can either be predetermined or decided based on a similarity score threshold.
- **Summary Evaluation:** The final step is to evaluate the summary created using various metrics such as ROUGE, BLEU, or F-score. These metrics provide a measure of how well the summary captures the main ideas and key information of the original document.
- **Model Tuning:** Hyperparameters such as the similarity criterion and the number of sentences to include in the summary can be altered to enhance the performance of the summarization model. This can be done by experimenting with different values of these parameters and evaluating the resulting summaries using the evaluation metrics.

In text mining and information retrieval, cosine similarity is a popular way to gauge similarity between two vectors. It may also be used to compare the similarity of pairs of phrases or documents in the context of text summarization. The cosine similarity metric runs from 0 to 1, with 1 indicating full similarity.

To calculate the cosine similarity among two vectors, A and B, divide the dot product of the two vectors by the product of their magnitudes. This formula helps to quantify the degree of similarity between the two vectors, where a higher value implies greater similarity.

$$cosinesimilarity(A, B) = (A \cdot B)/(||A|| \times ||B||) \tag{1}$$

where:

$A \cdot B$ is the dot product of A and B

$||A||$ is the Euclidean norm (magnitude) of A

$||B||$ is the Euclidean norm (magnitude) of B

By using cosine similarity, we can compare the similarity between pairs of sentences or documents without worrying about the length of the sentences or documents. This is because cosine similarity is normalized by the magnitudes of the vectors, which takes into account the differences in length between the vectors.

Abstractive text summarization is the process of condensing a lengthy text into a concise statement that conveys the main ideas and meaning of the original text. Abstractive text summarising creates new sentences that might not be found in the original text, as opposed to extractive text summarization, which chooses and rephrases significant sentences from the original text. Because they can recognise intricate correlations and patterns in data, deep learning techniques like neural networks are frequently utilised for abstractive text summarization.

The steps involved in the abstractive approach are as follows:

1) **Sequence-to-Sequence Modeling:** A deep learning architecture called sequence-to-sequence (Seq2Seq) modelling is used to change an input sequence of one kind into an output sequence of another type. A recurrent neural network (RNN), which can be an LSTM (long short-term memory) or a gated recurrent unit (GRU), is typically used to encode the input sequence. A different RNN, instructed to predict the next element in the output sequence according to the previous elements, is then used to generate the output sequence. Dealing with input and output sequences of varying lengths is one of the primary difficulties in Seq2Seq modelling. Seq2Seq models commonly include a mechanism for attention (Attention Layer), which enables the model to concentrate on various elements of the input sequence while it creates the output sequence, to get around this problem.
The objective of developing text summarization using the many-to-many Seq2Seq approach is to provide a brief summary as the output when a lengthy text is provided as the input. The major components of a Seq2Seq model are encoders and decoders.

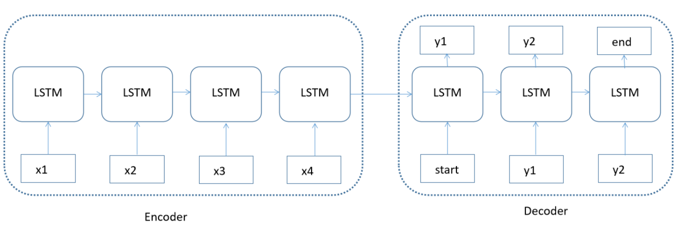

Fig. 1. Encoder And Decoder architecture

2) **Encoder and Decoder Architecture:** The Encoder and the Decoder, as shown in Figs. 1 and 2, are the two fundamental parts of the Encoder-Decoder architecture, a key framework in deep learning. Recurrent neural networks (RNNs), such as Gated Recurrent Neural Networks (GRU) or Long-Short-Term Memory (LSTM), are typically used to implement these components. The main benefit of using RNNs is that they can capture long-term relationships in the input sequence due to their ability to circumvent the vanishing gradient problem (Vanishing gradient).

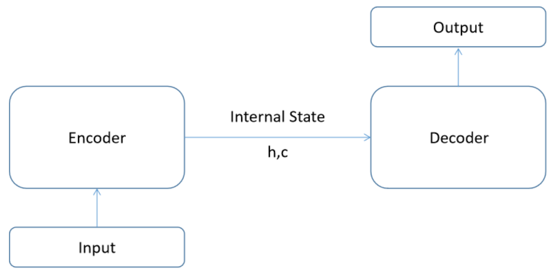

Fig. 2. Encoders and decoders

We can set up the encoder-decoder in two phases namely the "Training phase" and the "Inference phase".

3) **Training Phase**: We'll establish the encoder and decoder components of the model and then proceed to train it to forecast the target sequence, shifted by a one-time step.
   - **Encoder:** In Fig.3, A recurrent neural network, such as Long Short-Term Memory (LSTM), serves as the encoder in most cases. It goes over each word in the input sequence one at a time and generates a hidden state vector for each one. A fixed-length representation of the input sequence is used as the last hidden state of the encoder. To put it another way, the encoder analyses the data one word at a time at each time step, collecting the contextual data included in the input sequence.
   The encoder in Fig.3 transfers its final hidden state (h) with cell state (c) to the decoder in Fig.4. The reason for this is that both the decoder and encoder are two distinct LSTM-based architectures that must have their own beginning states in order to operate properly.
   - **Decoder:** As shown in Fig.4, the decoder is also usually an RNN, which takes the fixed-length vector from the encoder as an input and generates the output sequence word by word. The decoder component of an encoder-decoder architecture is in charge of producing the output sequence depending on the input sequence that has been encoded. To do this,

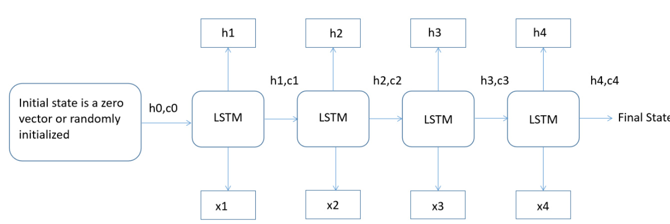

Fig. 3. Inside an Encoder

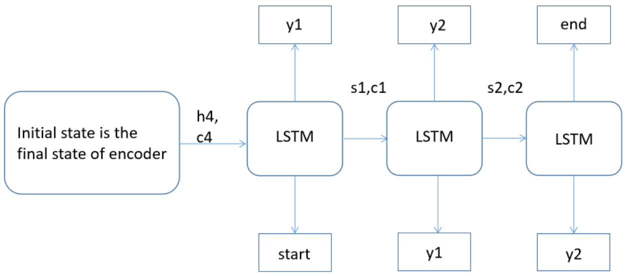

Fig. 4. Inside a Decoder

the decoder initializes its hidden state using a fixed-length vector produced by the encoder. Then, based on the preceding word and the current concealed state, it constructs each word in the output sequence. Until the whole output sequence is produced, the decoder is trained to anticipate the following word in the sequence given the preceding word. The **start** and **end** tokens are unique symbols that are inserted into the desired target sequence during a deep learning model's training phase. The beginning and conclusion of the sequence are indicated by these tokens. The **start** token is always the initial word predicted by the model during the testing phase. This serves as a signal to the model that it should start generating the output sequence. The "end" token is used to indicate to the model when it has finished generating the sequence. Once the model predicts the **end** token, it stops generating new words.

4) **Inference Phase:** An inference architecture is required to decode a test sequence in order to assess the model's performance on fresh source sequences where the target sequence is unknown. To decode a test sequence, the following steps are taken:
   a) Encode the full input sequence first, then initialize the decoder with the resultant internal states.
   b) As the decoder's first input, provide it the **start** token.
   c) Utilising the encoder's internal states, run the decoder for a single step.
   d) A distribution of probabilities for the following word in the sequence will be produced by the decoder. The anticipated next word should be the one with the highest likelihood.
   e) Update the decoder's internal states with the most recent time step while using the chosen word as the input for the following time step.
   f) Keep repeating steps **c-e** until the decoder predicts the 'end' token or until the maximum target sequence length is reached.

The neural network's requirement to condense all pertinent information from a source phrase into a vector of a certain length presents a potential issue with the encoder-decoder technique. This could create difficulty for the neural network when processing lengthy sentences. As the input sentence's length increases, the performance of a basic encoder-decoder quickly declines.

Because of this, we want an attention mechanism that allows us to anticipate a word by focusing only on a select few discrete sequence elements rather than the complete sequence.

Attention Mechanism: In the context of natural language processing, the attention mechanism determines the amount of emphasis required on each word in the input sequence while generating a word at a particular step. The idea behind this concept is to identify the specific sections of the input sequence that have higher relevance in generating the target sequence by computing the attention weights for each input element. Rather than giving equal importance to all the input elements, the attention mechanism allows us to give more weight to the relevant parts. The decoder then calculates a weighted sum of the input elements based on the attention weights, which act as weights for the sum.

Global and local attention are two different categories of attention mechanisms. The suggested approach makes advantage of the global Attention Layer.

- **Global Attention:** In this method, attention is given to every position in the source sequence, meaning that all the encoder's hidden states are taken into account to generate the attended context vector. To compute the attention weights, the decoder state is multiplied by the encoder states for each element of the input sequence using a dot product. The resulting dot products are normalized to obtain the attention weights.

IV. Experimental Results

In my experiments, I found that our approach performed better than the existing state-of-the-art models. The model produced impressive ROUGE scores, showing significant improvements in both ROUGE-1 and ROUGE-2 metrics. These results suggest that our approach successfully captures the important information from the original text and generates clear and coherent summaries.

A. *Dataset used:*

In this research work, we have used Amazon Fine Food Reviews dataset [23].This dataset, sourced from the Kaggle website [24], consists of a vast collection of reviews pertaining to fine foods available on Amazon. The dataset covers a substantial time frame, ranging from October 1999 to October 2012. It encompasses an impressive total of 568,454 reviews, offering a wealth of information for analysis and exploration. The dataset encompasses a rich variety of 74,258 distinct fine food products, ensuring a broad coverage of the culinary landscape. This diversity allows for in-depth exploration of specific products, brands, or categories, enabling researchers to glean valuable insights into consumer preferences and market dynamics. In addition to the extensive review corpus, the dataset comprises details on 256,059 unique users who have contributed their insights and opinions. These users form a diverse set, bringing a wide range of perspectives to the reviews. While primarily centered around fine foods, it's worth noting that this dataset also encompasses reviews from other diverse categories available on Amazon. The dataset includes two main components:

Reviews.csv: This file is extracted from the corresponding SQLite table called "Reviews" within the "database.SQLite" file. It likely contains the structured data in a tabular format, with columns representing different attributes of the reviews such as product information, user details, ratings, and the text of the review itself. The CSV format allows for easy handling and analysis of the data using various software tools and programming languages.

database.SQLite: This file is an SQLite database file containing a table named "Reviews." SQLite is a popular lightweight relational database management system. The "database.SQlite" file likely stores the entire dataset in a structured manner, with the "Reviews" table containing the detailed information of each review, including its associated attributes and data fields. This dataset is the collective work of McAuley et. al [25]

B. Evaluation Techniques:

The effectiveness of the intended models for producing textual summaries of reviews and unstructured text data has been evaluated, and the findings have been contrasted with cutting-edge learning techniques like the BLEU [26] score and ROUGE [27] score. A frequently used statistic called the BLEU score evaluates the quality of machine-generated text by contrasting it with one or more reference summaries. Based on n-grams (contiguous word sequences), it determines how much the generated summary and reference summaries overlap. In addition to shortness, which penalizes overly brief summaries, BLEU takes into account precision, which gauges how many n-grams from the produced summary match those from the reference summaries.BLEU score works by calculating a modified n-gram precision score for various n-gram orders (typically 1 to 4). The individual precision scores are then combined using a weighted geometric mean to obtain the final BLEU score. A higher BLEU score indicates a closer match between the generated summary and the reference summaries. The modified accuracy values produced on the text data are geometrically averaged out and then compounded by an exponential penalty ratio.

An assessment measure set created especially for automated summarization activities is called the ROUGE score. An indicator used to assess how closely the produced summary and the reference summaries resemble each other is the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric. The amount of recall and accuracy, which measure what percentage of the reference summary is included in the produced summary and how much of the generated summary is pertinent to the reference summaries, respectively, are used to determine the degree of overlap. Several techniques, including ROUGE-N (which measures n-gram overlap), ROUGE-L (which determines the longest common subsequence), and ROUGE-S (which concentrates on skip-bigram statistics), are used by the ROUGE score to evaluate the quality of summaries. Similar to BLEU, ROUGE scores are computed for several n-gram orders and then added together to get an overall score. Better summarising performance is indicated by higher ROUGE scores. Better summarising abilities are indicated by higher ROUGE scores.

TABLE I
ROUGE SCORE VALUES FROM PROPOSED ENCODER-DECODER ARCHITECTURE WITH ATTENTION LAYER AND LSTM SEQ2SEQ MODEL

Evaluation Matrix	Proposed model	Existing state-of-the-art model.
ROUGE-1 [27]	0.456	0.33
ROUGE-2 [27]	0.0011	0.0
ROUGE-L [27]	0.0.399	0.33

TABLE II
ROUGE SCORE GENERATED BY EXTRACTIVE SUMMARIZATION MODEL

SCORES	ROUGE-1	ROUGE-2	ROUGE-L
Recall (r)	0.290	0.2403	0.290
Precision (p)	1.0	0.9636	1.0
F-1 (f)	0.4508	0.384	0.450

C. ROUGE scores and cutting-edge techniques are compared.

Table I shows the comparison between the proposed LSTM model with an attention layer and a simple LSTM model without any attention layer. This comparison of the proposed and state-of-the-art model is based upon the ROUGE score for a Review of a particular product.

Summary:"Great product great service"

Predicted summary:"great food"

D. Results:

Table II represents the ROUGE scores that were generated by the extractive summarization model that uses a cosine similarity matrix. The results were achieved upon a raw paragraph that has about 500 words.

The precision (p) values for both ROUGE-1 and ROUGE-L are high (1.0), indicating a strong agreement between the generated summaries and the reference summaries, the recall (r) values are relatively lower. The F1 scores, which represent the harmonic mean of precision and recall, also show room for improvement.

The recall value suggests that not all relevant information from the input text is captured in the generated summary. Similarly, for

ROUGE-2, the recall value is indicating a limited ability to generate accurate two-word sequences. The ROUGE-L recall value aligns with ROUGE-1, indicating room for improvement in terms of capturing overall content.

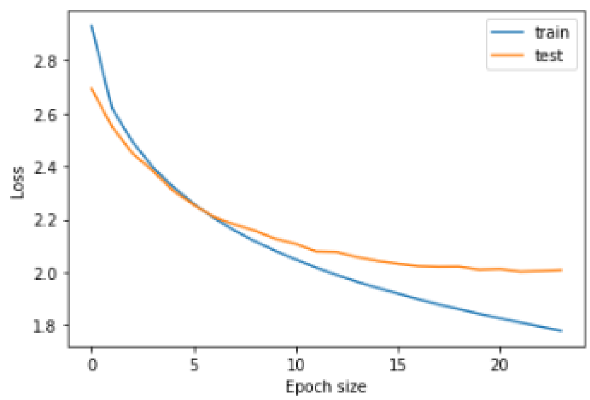

Fig. 5. diagnostic plot for training loss and validation loss

While the cosine similarity approach shows potential, further enhancements may be required to increase recall and overall F1 scores.

Here are the ROUGE scores of the abstractive model for the Amazon Fine Food Reviews dataset. [23]

Review: "got treasure hunt daughter came quickly packed right thrilled end back"

Original summary: "Great product great price."

Predicted summary: "great gift"

TABLE III
ROUGE SCORE GENERATED BY ABSTRACTIVE SUMMARIZATION MODEL

SCORES	ROUGE-1	ROUGE-2	ROUGE-L
Recall (r)	0.456	0.01	0.396
Precision (p)	0.5	0.0	0.5
F-1 (f)	0.399	0.0	0.399

When evaluated using ROUGE scores, the results indicate the following performance:

The ROUGE-1 scores indicate that approximately one-third of the content in the predicted summary overlaps with the content in the reference summary. However, the scores for ROUGE-2 show no overlapping two-word sequences, resulting in a lack of precision and recall. The ROUGE-L scores align with the ROUGE-1 scores, indicating similar performance in terms of content overlap.

Fig.5 represents the diagnostic plot for the training loss and validation loss during model preparation.

V. Conclusion and Future Scope

This research paper introduced a text summarization model that employed a hybrid approach, combining both extractive and abstractive techniques, with a specific focus on the abstractive approach utilizing the attention mechanism. The model's performance was evaluated using the widely accepted metric which is called ROUGE (Recall-Oriented Understudy for Gisting Evaluation).

By integrating the attention mechanism into the abstractive approach, the model aimed to enhance its ability to generate coherent and contextually relevant summaries. The attention mechanism allowed the model to assign varying levels of importance to different parts of the source text, enabling it to focus on salient information while generating the summary.

The evaluation using ROUGE scores provided insights into the model's performance. The abstractive approach, enriched by the attention mechanism, exhibited promising results in terms of content overlap and summary quality.

The primary goal of the study was to construct a model that can produce meaningful summaries of reviews written about fine foods sold on Amazon, as well as generate summaries of raw text data. The analysis of the code revealed that the model has the ability to produce concise sentences or word pairs as output when given large input sizes. This model could also be useful in other domains apart from the specific use case presented in the study, such as financial research, legal contract analysis, and social media marketing.

The future scope of this approach is promising, and several potential directions can be explored to address the current limitations and improve the overall performance of the model:

- **Increased Training Epochs:** Conducting additional training epochs can significantly enhance the model's performance. Increasing the number of training iterations allows the model to learn more complex patterns and capture a more comprehensive understanding of the data. This extended training duration can potentially result in more accurate and reliable summaries.
- **Hyperparameter Tuning:** Fine-tuning the model's hyperparameters is crucial for optimizing its performance. By systematically adjusting parameters such as learning rate, batch size, and regularization techniques, the model can be optimized to achieve better generalization and improve its ability to generate high-quality summaries
- **Ensemble Models:** Exploring ensemble techniques by combining multiple summarization models can lead to improved results. Ensemble models can leverage the diverse strengths and capabilities of different models, mitigating individual model limitations and enhancing the overall summarization quality.

References

[1] Mengli Zhang, Gang Zhou, Wanting Yu, Ningbo Huang, and Wenfen Liu Proposed the "A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning" in the "National Library of Medicine, Journal- Comput Intell Neurosci", vol.2022, 2022.

[2] Suad Alhojely and Jugal Kalita published their article "Recent Progress on Text Summarization" at the 2020 "International Conference on Computational Science and Computational Intelligence (CSCI)", 2022

[3] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çaglar G ˘ ulçehre, and Bing Xiang Published an article on "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond" at "Cornell University Computation and Language Journal", vol.1,19 Feb 2016.

[4] Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, and De Rosal Ignatius Moses Setiadi, published "Review of automatic text summarization techniques & methods" paper in the Journal "Computer and Information Sciences", vol.34, issue 4, pp. 1029-1046, on April 2022.

[5] Ahmed. Rafea, Hoda. Mohamed, Wafaa. El-Kassas, and Cherif. Salama published "Automatic text summarization: A comprehensive survey "in Journal "Expert Systems with Applications" vol.165, 1 March 2021, 113679

[6] Senthilkumar Mathi, Manojkumar, and Xiao-Zhi Gao proposed ”An Experimental Investigation on Unsupervised Text Summarization for Customer Reviews” in Journal ”Procedia Computer Science”. vol. 218, 31 January 2023, pp. 1692-1701.
[7] Mohammad Bani-Almarjeh and Mohamad-Bassam Kurdy, proposed ”Arabic abstractive text summarization using RNN-based and transformer-based architectures” in Journal ”Information Processing and Management”, vol. 60, March 2023
[8] Betul Ay, Fatih Ertam, Guven Fidan, and Galip Aydin published ”Turkish abstractive text document summarization using text to text transfer transformer” in the Journal of ”Alexandria Engineering”, vol. 68, pp. 1-13, 2023.
[9] K Veningston, P V Venkateswara Rao, and M Ronalda published the article on ”Procedia Computer Science” vol.218, 2023, Pages 1220-1228.”
[10] K.R. Sabarmathi, Padhma Muniraj, R. Leelavathi and Saravana Balaji. Published their article ”HNTSumm: Hybrid text summarization of transliterated news articles” in the ”International Journal of Intelligent Networks vol.4, 2023, pp. 53-61.”
[11] Rini Wijayanti, Masayu L. Khodra, Kridanto Surendro, and Dwi H. Widyantoro published an article ”Learning bilingual word embedding for automatic text summarization in low resource language” in the Journal of”King Saud University - Computer and Information Sciences”, vol.35, April 2023, pp.224-235
[12] Richard Xiao purposed an article on ”Corpus Creation” in the Journal of ”Edge Hill University”
[13] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. In the article ”Efficient Estimation of Word Representations in Vector Space” Published on 7 September 2013.
[14] Rajeshkhannan Regunathan, Shivangi Chaurasia, and Debalay Dasgupta published their work ”T5LSTM-RNN based Text Summarization Model for Behavioral Biology Literature”, Vol.218, 2023, pp. 585-593”, in the journal of ”Procedia Computer Science” vol.218, 2023, pp.585-593”
[15] T. P. Kancharlapalli and P. Dwivedi, ”A Novel Approach for Age and Gender Detection using Deep Convolution Neural Network,” 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023, pp. 873-878.
[16] P. Dwivedi and B. Sharan, ”Deep Inception Based Convolutional Neural Network Model for Facial Key-Points Detection,” 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 792-799, doi: 10.1109/ICCCIS56430.2022.10037639.
[17] P. Dwivedi and A. Upadhyaya, ”A Novel Deep Learning Model for Accurate Prediction of Image Captions in Fashion Industry,” 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2022, pp. 207-212, doi: 10.1109/Confluence52989.2022.9734171.
[18] P. Dwivedi and B. Islam, ”An Item-based Collaborative Filtering Approach for Movie Recommendation System,” 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023, pp. 153-158.
[19] C. Lala and P. Dwivedi, ”Hate Speech Detection Network Using LSTM,” 2023 International Conference for Advancement in Technology (ICONAT), Goa, India, 2023, pp. 1-6, doi: 10.1109/ICONAT57137.2023.10080786.
[20] Andrew Pulver, and Siwei Lyu published their research work on ”LSTM with working memory” in the IEEE in the Journal of ”International Joint Conference on Neural Networks” in 2017
[21] Anjusha Pimpalshende, Shailendra S. Aote, Archana Potnurwar, and Shantanu Lohi published ”Binary Particle Swarm Optimization with an improved genetic algorithm to solve multi-document text summarization problem of Hindi documents” in the journal ”Engineering Applications of Artificial Intelligence” vol.117 Part A, January 2023.
[22] Sepp Hochreiter in ”The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions” at the journal of ”Institut fur Informatik, Technische Universit at Munchen”
[23] Amazon Fine Foods Reviews: https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews
[24] a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. For further reference: https://www.kaggle.com/
[25] J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.
[26] Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J., “BLEU: a method for automatic evaluation of machine translation,” ACL- 2002: 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318, CiteSeerX 10.1.1.19.9416, 2002
[27] Chin-Yew Lin, ”ROUGE: A Package for Automatic Evaluation of Summaries”, in the journal of ”Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey”, pp.74-81, 2004