ABSTRACTIVE TEXT SUMMARIZATION USING LSTM BASED DEEP LEARNING

A Report Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Piyush Yadav, 20214091
Rishu Raj, 20214043
Naresh Kumar, 20214321
Rishikesh Kumar, 20214246

to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD PRAYAGRAJ
May, 2024

UNDERTAKING

I declare that the work presented in this report titled "*ABSTRACTIVE TEXT SUMMARIZATION USING LSTM BASED DEEP LEARNING*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

May, 2024
Allahabad

(Piyush Yadav, 20214091
Rishu Raj, 20214043
Naresh Kumar, 20214321
Rishikesh Kumar,
20214246)

CERTIFICATE

Certified that the work contained in the report titled "*ABSTRACTIVE TEXT SUMMARIZATION USING LSTM BASED DEEP LEARNING*", by *Piyush Yadav, 20214091*
Rishu Raj, 20214043
Naresh Kumar, 20214321
Rishikesh Kumar, 20214246, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Prof. R.S. Yadav)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

May, 2024

Preface

A good B.Tech. thesis is one that helps you in furthering your interest in a specific field of study. Whether you plan to work in an industry or wish to take up academics as a way of life, your thesis plays an important role. Your thesis should judiciously combine theory with practice. It should result in a realization of reasonably complex system (software and/or hardware). Given various limitations, it is always better to extend your predecessor's work. If you plan it properly, you can really build on the experience of your seniors.

Acknowledgement

It is with profound gratitude that we acknowledge the giants on whose shoulders we stand. We extend our sincere appreciation to Professor R.S. Yadav for entrusting us with the opportunity to embark on this project and for his unwavering guidance and encouragement at every juncture. His sage recommendations, reassuring counsel, and meticulous scrutiny have served as wellsprings of creative inspiration and have been instrumental in the genesis of this report.

Sir's keen insight and unwavering enthusiasm for scholarly pursuits have propelled us to delve deeper into the complexities of the subject matter, both through paper consultations and engaging discussions. His enduring commitment to fostering a culture of excellence in research and his steadfast support for optimal learning outcomes have been a source of immense inspiration throughout this journey. We are also indebted to various individuals and groups who generously shared their insights and expertise, enriching our understanding of practical applications in real-world scenarios. Their willingness to impart knowledge has been invaluable in shaping the scope and depth of this endeavor.

Contents

Chapter 1

Introduction

In the era of information overload, the ability to distill vast amounts of text into concise, meaningful summaries is paramount. Abstractive text summarization, a branch of natural language processing (NLP), aims to generate condensed representations of documents while preserving their essential meaning. This report delves into the realm of abstractive text summarization, focusing specifically on a cutting-edge approach leveraging Long Short-Term Memory (LSTM) networks.

Traditional methods of text summarization, such as extractive techniques, often rely on selecting and rearranging existing sentences from the original text. While effective to some extent, these methods may lack coherence and fail to capture the underlying semantic structure of the document. In contrast, abstractive summarization models, inspired by human cognition, generate summaries by synthesizing new phrases and rephrasing content, akin to how a person would summarize a passage in their own words.

The rise of deep learning has revolutionized the field of NLP, offering powerful tools for abstractive summarization. LSTM networks, known for their ability to capture long-range dependencies in sequential data, provide a robust framework for understanding and generating natural language. When combined with CNNs, which excel at capturing local patterns and features, the resulting hybrid architecture can effectively handle both the semantic understanding and syntactic structure of text.

Throughout this report, we will explore the architecture, training process, and

performance of LSTM based models for abstractive text summarization. Additionally, we will discuss challenges and future directions in this rapidly evolving field, highlighting opportunities for further innovation and research. By delving into the intricacies of this advanced approach, we aim to provide a comprehensive understanding of the capabilities and potential applications of LSTM based deep learning in text summarization.

1.1 Motivation

The motivation behind undertaking this project stems from the recognition of the ever-growing need for efficient and effective methods of information synthesis in an age characterized by an abundance of textual data. In a world inundated with vast quantities of information across diverse domains, the ability to distill this wealth of data into concise and meaningful summaries is indispensable.

Traditional methods of text summarization, particularly extractive techniques, while useful to some extent, often fall short in capturing the nuanced relationships and underlying context within the original text. These methods tend to merely extract and rearrange existing sentences, lacking the ability to generate novel, coherent summaries that encapsulate the essence of the source material.

Abstractive text summarization, on the other hand, holds the promise of addressing this limitation by enabling the synthesis of new phrases and rephrasing that capture the essence of the original text. Leveraging the power of deep learning, particularly LSTM based architectures, presents an opportunity to advance the state-of-the-art in abstractive summarization, offering the potential for more accurate, coherent, and contextually rich summaries.

The potential applications of such advancements are manifold. From aiding readers in quickly grasping the key points of news articles to assisting professionals in navigating complex legal documents, the impact of robust abstractive summarization techniques extends across various domains. Furthermore, in the realm of education, healthcare, and social media analysis, the ability to distill vast amounts of information into succinct summaries can streamline decision-making processes, enhance knowledge dissemination, and facilitate deeper insights into complex phenomena.

Chapter 2

Related Work

- Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges
  View at : https://www.hindawi.com/journals/mpe/2020/9365340/
- Abstractive Text Summarization Based on Deep Learning and Semantic Content Generalization
  View at : https://aclanthology.org/P19-1501/
- Bidirectional LSTM Networks for Abstractive Text Summarization
  View at : https://link.springer.com/chapter/10.1007/978-3-030-90055-$7_2$1/

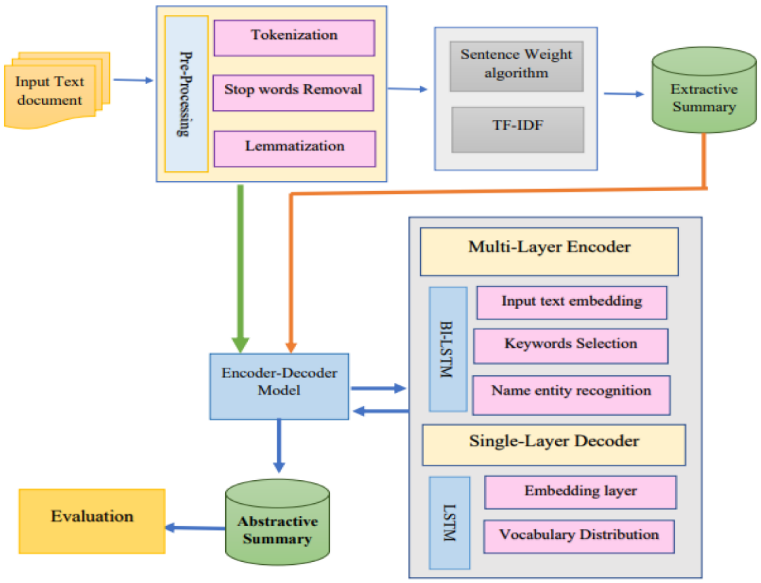
Prior research in the field of abstractive text summarization using LSTM based deep learning has laid the groundwork for our understanding of the challenges,methodologies, and advancements in this area. The following studies represent a selection of seminal works that have significantly contributed to the body of knowledge surrounding this topic:
1. "Abstractive Sentence Summarization with Attentive Recurrent Neural Networks" by Rush, Alexander M., Sumit Chopra, and Jason Weston (2015). • This influential study introduced the concept of abstractive sentence summarization using recurrent neural networks (RNNs) with attention mechanisms. By applying attention mechanisms to the summarization task, the authors demonstrated significant

improvements in the quality and fluency of generated summaries. convolutional layers, the model achieved competitive performance while reducing training time and computational complexity.

Chapter 3

Methodology

3.1 Workflow



3.2 Data Collection :

- Dataset Description:
  Title: Kaggle News Summary Dataset
  Source: Kaggle (provide link to dataset)
  Rows: 98,402

Columns: (List all columns and their descriptions)

- Data Sources:
  Provide a brief overview of where the dataset was sourced from on Kaggle, including any relevant details about the dataset's origin and collection methodology.
- Data Retrieval:
  Download the dataset from Kaggle or access it through Kaggle's API.

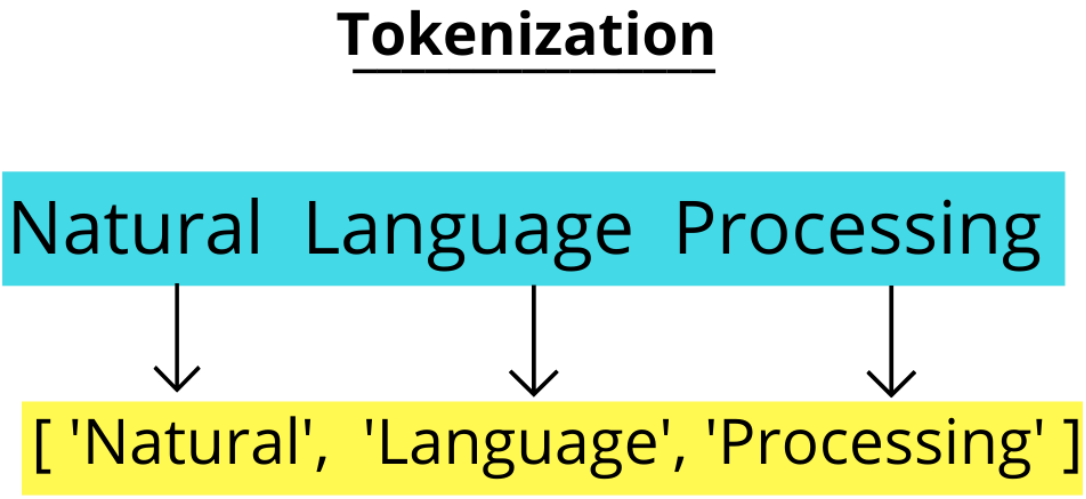
3.3 Data Preprocessing :

For preparing data suitable for the requirement, we have used specific common approach, such as:

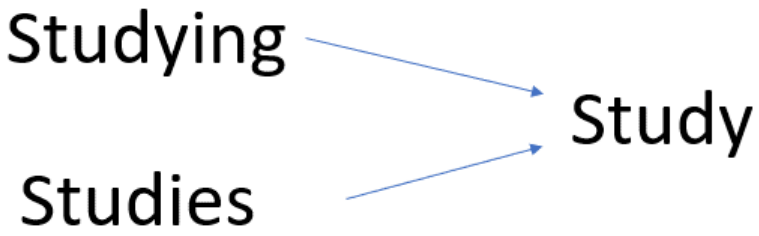
- Stop Word Removal
- Tokenization
- Lemmatization
- Expanding Contractions Using Regex
- **Stop Word Removal**- This process helps reduce the dimensionality of the text data and can improve the performance of natural language processing tasks such as text classification or sentiment analysis.

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

- **Tokenization**-In natural language processing, tokenization typically involves splitting text into words or sub words based on whitespace or punctuation.



- **Lemmatization**- Lemmetization helps normalize text data by reducing inflected or variant forms of words to a common base form, which can improve the performance of text analysis tasks by reducing vocabulary size and capturing the underlying meaning of words.



- **Expanding Contractions Using Regex**- Contractions are shortened versions of words or phrases that combine two words into one by replacing one or more letters with an apostrophe (e.g., ”don’t” for ”do not”, ”can’t” for ”cannot”). For example, using regex, you can convert ”I’m” to ”I am”, ”can’t” to ”cannot”, ”won’t” to ”will not”, and so on.

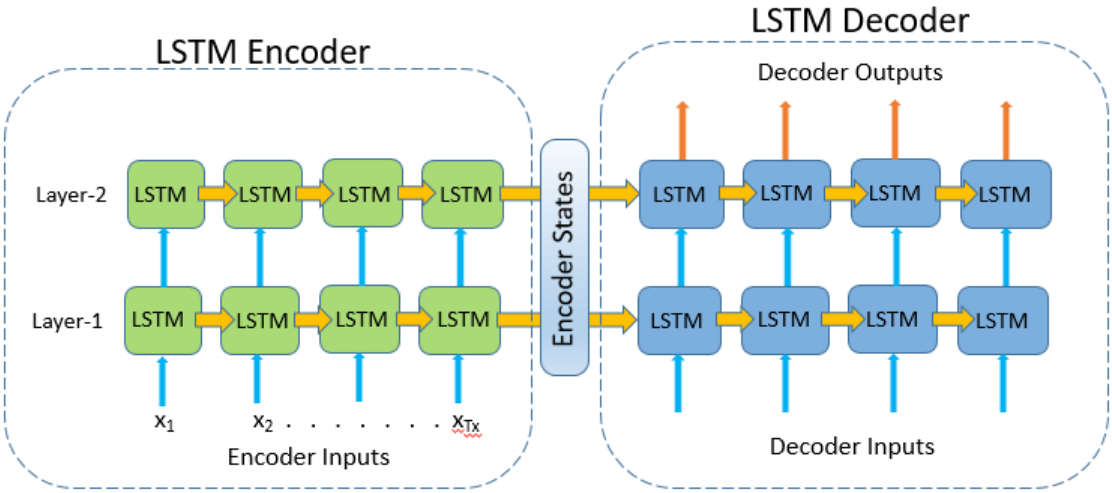
3.4 Encoder Decoder Architecture

3.4.1 Encoder Model

- The encoder processes the input text (e.g., news articles) and generates a context vector that captures the semantic meaning of the input.
- It typically consists of one or more LSTM (Long Short-Term Memory) layers.
- Each LSTM layer processes the input sequence token by token, updating its hidden state at each time step. The final hidden state of the encoder LSTM(s) represents a condensed representation of the input text, often referred to as the "thought vector" or "context vector."
- This context vector contains information about the entire input sequence and is passed to the decoder to guide the generation of the summary.

3.4.2 Decoder Model

- The decoder takes the context vector from the encoder and generates the abstractive summary token by token.
- . It also typically consists of one or more LSTM layers
- At each time step, the decoder LSTM processes the previous token in the summary and the current hidden state to predict the next token.
- The decoder LSTM(s) are initialized with the context vector from the encoder, which provides the initial state and context for generating the summary.
- During training, the decoder is trained to generate the summary tokens by maximizing the likelihood of the target summary given the input text
- During inference (i.e., generating summaries for new input text), the decoder utilizes a mechanism such as beam search to iteratively generate the most likely sequence of tokens for the summary.



3.5 Bi-LSTM Architecture

Bidirectional LSTM or BiLSTM is a term used for a sequence model which contains two LSTM layers, one for processing input in the forward direction and the other for processing in the backward direction. It is usually used in NLP-related tasks. The intuition behind this approach is that by processing data in both directions, the model is able to better understand the relationship between sequences (e.g. knowing the following and preceding words in a sentence).

Architecture:

The architecture of bidirectional LSTM comprises of two unidirectional LSTMs which process the sequence in both forward and backward directions. This architecture can be interpreted as having two separate LSTM networks, one gets the sequence of tokens as it is while the other gets in the reverse order. Both of these LSTM network returns a probability vector as output and the final output is the combination of both of these probabilities. It can be represented as:

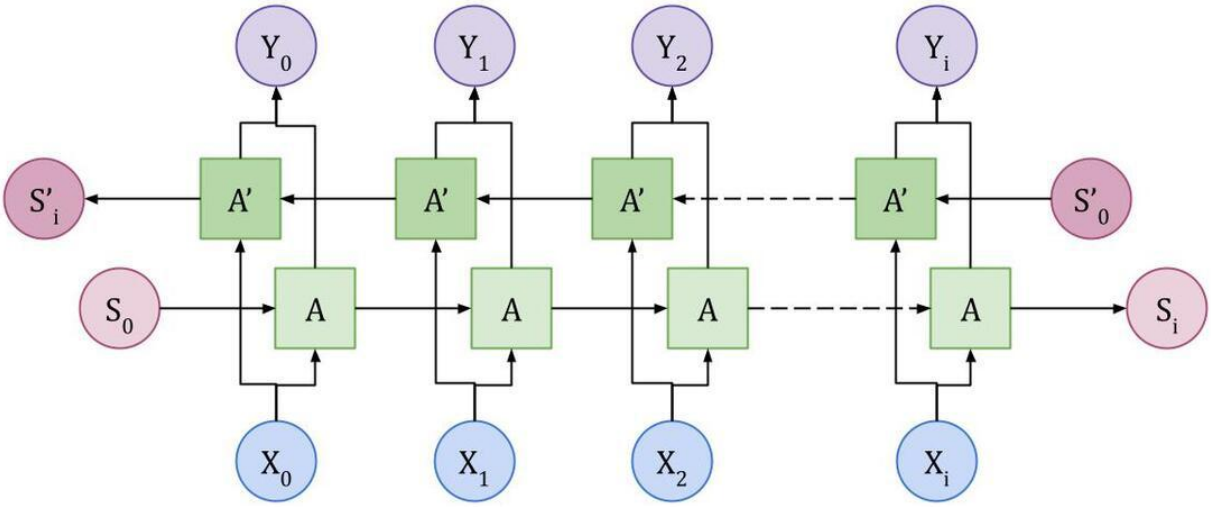


Fig.: Bidirectional LSTM layer Architecture

Chapter 4

Proposed Work

We have implement the certain neural network performance improvement techniques which consist of:

1. Early Stopping: The training process of machine learning projects can become overwhelming, especially when it's unclear how many epochs are needed for the model. To avoid such situations, the technique of early stopping is employed. It entails terminating the training process early if no progress is observed in the validation accuracy. Early stopping can help reduce unnecessary processing and enhance the overall efficiency of the training process.

```
earlystopping = EarlyStopping(monitor = 'val_auc',
                              mode = 'max' ,
                              patience = 5,
                              verbose = 1)

checkpoint    = ModelCheckpoint(filepath,
                                monitor = 'val_auc',
                                mode='max',
                                save_best_only=True,
                                verbose = 1)


callback_list = [earlystopping, checkpoint]
```

Fig. 6: Early Stopping and Checkpoints

2. Optimizer: Optimizer in machine learning refers to an algorithm that modifies the parameters of a model to minimize the differences between predicted and actual output. It is a key component of the training process and is used to update the model's parameters to minimize the loss function. There are several types of optimizers, such as Stochastic Gradient Descent, Adagrad, Adam, and RMSprop.

   Adam optimizer is a popular stochastic gradient descent optimization algorithm utilized in machine learning due to its efficient, low memory requirements, and ability to handle large datasets. It is an amalgamation of two other optimization algorithms, Adagrad and RMSprop, that offer better performance and quicker convergence.

3. No. of Epochs: 12

4. Training and validation split is 80%, 20%

5. Evaluation Metrics and Benchmarking: We will employ a comprehensive set of evaluation metrics to assess the performance of our proposed model objectively. These metrics will include traditional measures such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores as well as more nuanced metrics that capture aspects of summary quality such as readability and informativeness. Additionally, we will benchmark our model against existing state-of-the-art summarization systems to gauge its comparative performance.

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

6. **Libraries Used :**

- **Numpy**: NumPy is a library for numerical computing in Python. It provides support for multi-dimensional arrays and a collection of mathematical functions for operating on these arrays efficiently.
- **Pandas :** It provides data structures like DataFrames for working with structured data and a wide range of functions for data manipulation, cleaning, and exploration. In this project, Pandas might be used for tasks like loading datasets, preprocessing data, and analyzing the distribution of data.
- **Seaborn:** Seaborn is a data visualization library based on Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. In this project, Seaborn might be used for visualizing the distribution of data, exploring relationships between variables.
- **Matplotlib.pyplot:** Matplotlib is a comprehensive plotting library for creating static, interactive, and animated visualizations in Python. The pyplot module provides a MATLAB-like interface for creating plots and customizing their appearance.
- **sklearn:** Scikit-learn is a popular machine learning library in Python. The model selection module provides functions for splitting datasets into training and testing sets, which is essential for evaluating the performance of machine learning models.
- **tensorflow :** TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools and libraries for building and deploying machine learning models efficiently. In this project.
- **tensorflow.keras:** Keras is an open-source neural network library written in Python that provides a high-level API for building and training deep learning models. In this project, tensorflow.keras will be used for defining and training LSTM and Bi-LSTM models for text summarization.

Chapter 5

Experimental Setup & Result Analysis

After training the model the maximum frequency in 12 epochs turned out to be 71.35%. Following are the detailed results out of each epoch.

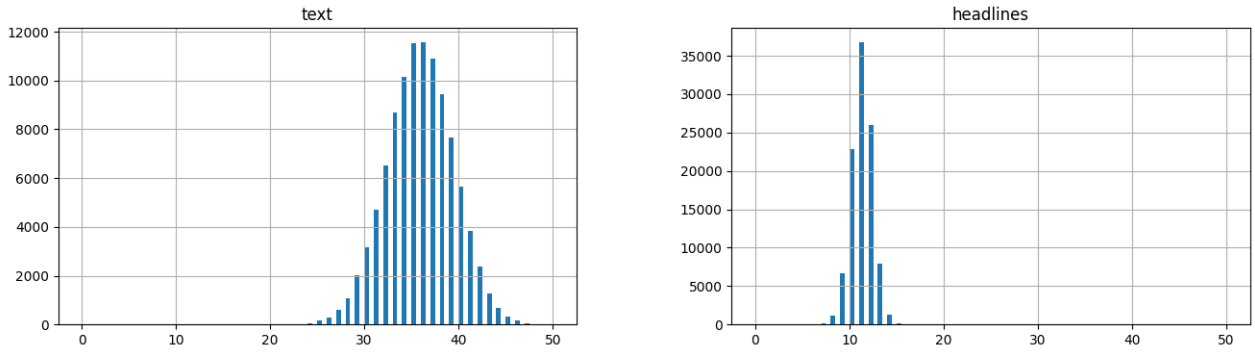


Fig.: No. of sentences VS length of sentences

5.0.1 Result Analysis

- *Accuracy:*

Accuracy measures how well a machine learning model can correctly predict or classify a given set of data. It is calculated as the ratio of the number of correct

predictions to the total number of predictions made by the model.

$$Accuracy = \frac{Correctlypredictedsamples}{Totalsamples}$$

- *Precision:*

  Precision is the percentage of true positive predictions.

$$Precision = \frac{TP}{TP + FP}$$

- *Recall:*

  Recall calculates the percentage of true positive predictions out of all actual positive cases in the data.

$$Recall = \frac{TP}{TP + FN}$$

- *Accuracy & Loss:*

  Following are the loss and accuracy curves per epoch:

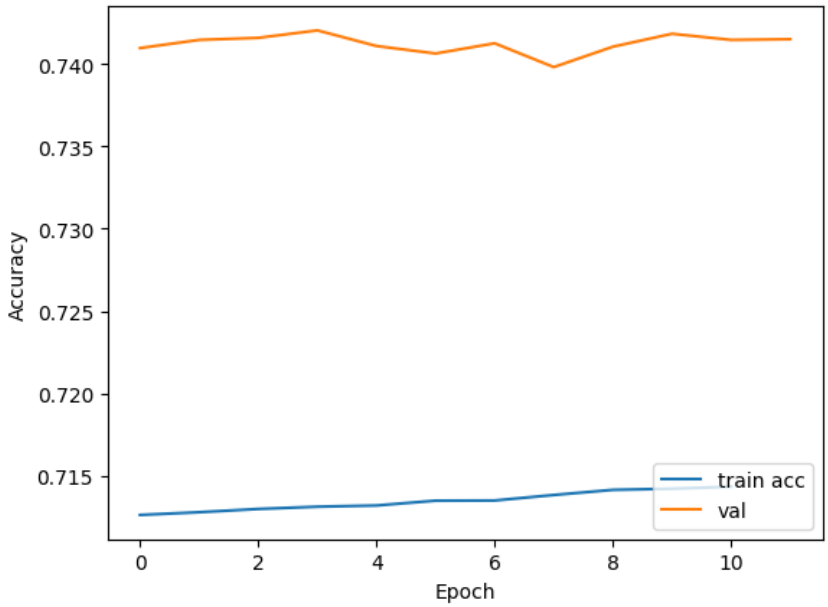


Fig.: Accuracy Curve

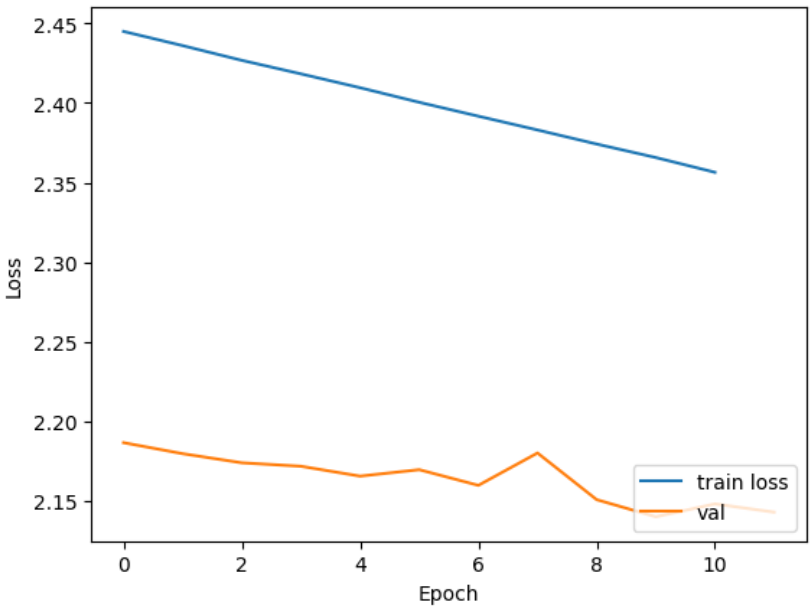


Fig. Loss Curve

Chapter 6

Conclusion & Future Work

6.1 Conclusion

In conclusion, the abstractive text summarization project utilizing LSTM and Bi-LSTM models has demonstrated promising results in generating concise and accurate summaries from longer texts.Both LSTM and Bi-LSTM models have shown effectiveness in capturing long-range dependencies and understanding the context of the text, which is crucial for generating coherent summaries. The bidirectional aspect of the Bi-LSTM model further enhances its ability to comprehend the text by considering both past and future information simultaneously.

However, like any machine learning model, there is always room for improvement. Future work could focus on exploring more advanced neural network architectures, incorporating attention mechanisms to better focus on important parts of the input text, and experimenting with larger datasets to further enhance the models' performance.

Overall, the abstractive text summarization project utilizing LSTM and Bi-LSTM models represents a significant step forward in the field of natural language processing, offering a promising approach to automatically generate meaningful and concise summaries from large volumes of text.

6.2 Future Work

As our project nears completion, it is crucial to look into its potential future developments and advancements. The project has several aspects that could be further improved in the future.

- **Addressing Overfitting:** The primary focus will be on mitigating overfitting issues within the neural network. Overfitting occurs when the model learns to perform well on the training data but fails to generalize effectively to new, unseen data. Techniques such as regularization, dropout, and early stopping will be explored to ensure the model's performance is robust and reliable.

- **Enhancing Bidirectional Encoding with BERT:** Implementing advanced transformer models like BERT (Bidirectional Encoder Representations from Transformers) will be a key strategy to improve bidirectional encoding. This enhancement will significantly enhance the understanding of semantics and the structural framing of abstract summaries. By leveraging BERT's pre-trained contextual embeddings, the model will grasp nuanced relationships within the text, leading to more accurate summarization.

- **Leveraging Transfer Learning:** Given the challenge of limited data availability, transfer learning will play a pivotal role in our approach. Transfer learning involves transferring knowledge from a pre-trained model to a new task with a smaller dataset. By fine-tuning pre-trained models on our specific dataset, we can capitalize on the generalizable knowledge learned from vast amounts of data, thereby boosting the performance of our model even with limited training samples.

- **Evaluate Pegasus Model Accuracy:** An integral part of our strategy involves assessing the accuracy of the Pegasus model. Pegasus, a transformer-based model specifically designed for text summarization tasks, offers promising capabilities in generating coherent and informative summaries. Through rigorous evaluation and comparison against existing benchmarks, we aim to ascertain the effectiveness of Pegasus in meeting our summarization objectives.

This evaluation will inform our decision-making process and guide further refinements in our approach.

By taking these future aspects into account, the project can continue to progress and remain pertinent in the dynamic field of machine learning. Moreover, it can provide opportunities for collaborations and partnerships with other researchers and organizations.

Chapter 7

References

1. Song, S., Huang, H. Ruan, T. Abstractive text summarization using LSTM-CNN based deep learning. Multimed Tools Appl 78, 857–875 (2019). https://doi.org/10.1007/s11042-018-5749-3

2. Bidirectional LSTM Networks for Abstractive Text Summarization
   View at : https://link.springer.com/chapter/10.1007/978-3-030-90055-721

3. Abstractive Text Summarization Based on Deep Learning and Semantic Content Generalization
   View at : https://aclanthology.org/P19-1501/