6TH SEMESTER INNOVATIVE PROJECT

Presented By : Aritra Ghosh and Subhojit Ghosh

2024

PRESENTED BY:

Aritra Ghosh
Section: A; Roll No: 74
Enrollment Number: 12021002002137
B.Tech CSE

Subhojit Ghosh
Section: B; Roll No: 97
Enrollment Number: 12021002002160
B.Tech CSE

Mentor: Dr. Anupam Mandal

AGENDA

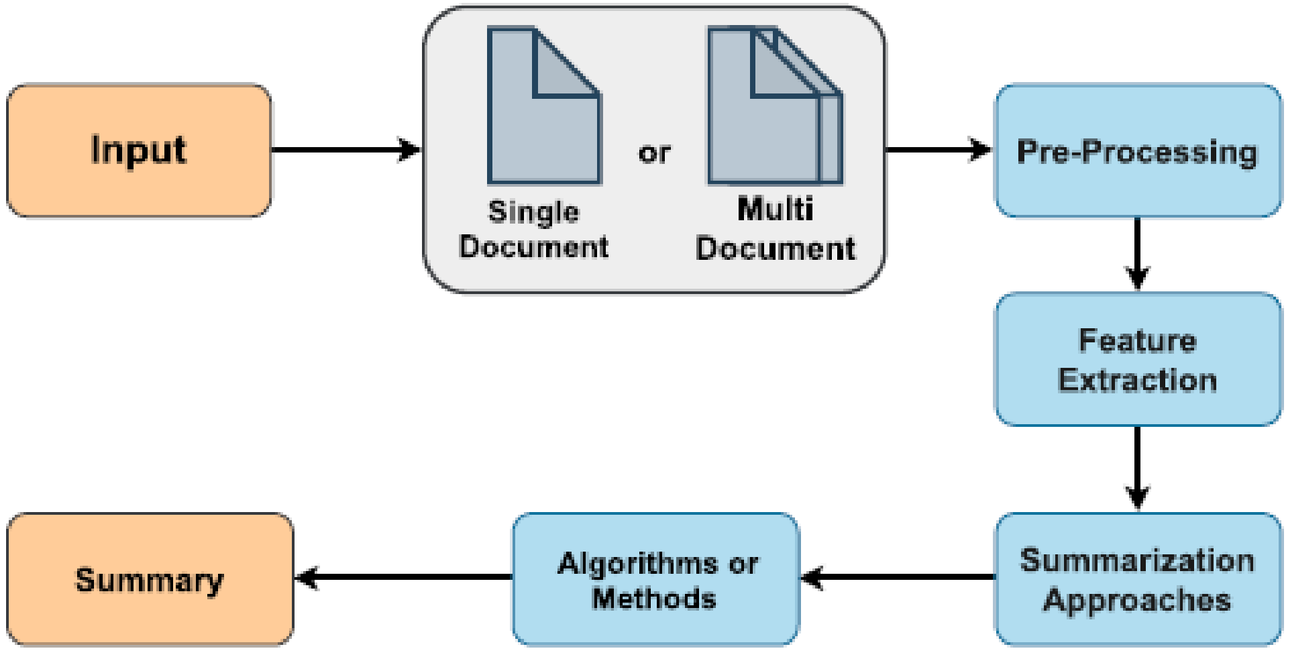
Problem Statement	4
Introduction	5
Overview	6
Work Methodology	7
Research Paper	8
Output	9
Timeline Chart	10
Ongoing Work	11
Future Scope	12
Conclusion	13

Problem Statement

To study the existing methods of text summarisation and identify the challenges in order to come up with a solution for them.

Introduction

- Summarization is the process of extracting key ideas from a larger text while retaining core concepts.
- Various approaches to summarization include fuzzy logic, concept-driven, latent semantic, machine learning, neural networks, and more.
- Existing methods have limitations like oversimplification, biases, reliance on static knowledge, and focus on extractive rather than abstractive summaries.
- The authors curated a custom dataset and are training a novel neural network model combining attention, graph neural networks, and transfer learning for improved summarization.



OVERVIEW

- Exhaustive analysis of existing research to identify promising summarization models and methodologies that require improvement.
- In-depth evaluation and comparative analysis of the top 10 models against key metrics like accuracy, coherence, and efficiency.
- Curation and creation of a custom dataset tailored to the specific requirements of the project for training and evaluation.
- Training and optimization of the top 3 identified models on the custom dataset through rigorous iterations, followed by consolidation and iterative refinement to develop the optimized summarization model.

WORK METHODOLOGY

- Literature Review and Model Analysis
- Comparative Analysis of Top Models
- Dataset Curation and Development
- Model Training and Evaluation
- Fine-Tuning and Optimization

TIMELINE CHART

Step No.	Description	Status	Result
1.	**Literature Review and Model Analysis** **a.** Generation of Report of Survey, Research Paper Review **b.** Develop a basic Model Using inbuilt library functions in Python to gain a better understanding of the problem and ask in hand	Completed	**a.** https://docs.google.com/document/d/1sNxGtEG7QVPPM6zPaM07_Tsb2P3EmxzEy48yHFyzX80/edit?usp=sharing **b.** https://github.com/TheCleverIdiott/summarizer
2.	**Comparative Analysis of Top Models** **a.** Comparative Analysis of Top Models **b.** Research Paper Publication - I	Completed	a. https://iemtronics.org/wp-content/uploads/2024/04/IEMTRONICS-2024-Conference-Proceedings-1.pdf b. https://iemtronics.org/wp-content/uploads/2024/04/IEMTRONICS-2024-Conference-Proceedings-1.pdf
3.	**Dataset Curation and Development** a. Creating a Custom Dataset from several Genres that target humane text summary to train Further Models	Completed	a. https://mega.nz/file/sdBwHQRI#yykHpuwcHldDz-K_EG-qhXGcgvvJyO4eNydVMEkKWhs
4.	**Model Training and Evaluation**	Ongoing	
5.	**Fine-Tuning and Optimization**	Scheduled	

RESEARCH PAPER

Informed by an in-depth exploration of existing scholarly works, we contributed to the field by publishing our research paper titled "An Analytical Survey of Text Summarization Techniques." This comprehensive study was presented at the prestigious IEMTronics, International IOT, Electronics and Mechatronics Conference, hosted from April 3rd to 5th, 2024, at Imperial College London, UK. (https://iemtronics.org/wp-content/uploads/2024/04/IEMTRONICS-2024-Conference-Proceedings-1.pdf)

International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2024

An Analytical Study of Text Summarization Techniques

Bavrabi Ghosh[1], Aritra Ghosh[2], Subhojit Ghosh[3], and Anupam Mondal[4]

Institute of Engineering & Management, Kolkata, India,
University of Engineering & Management, Kolkata, India
bavrabi.ghosh@iem.edu.in, aritra.ghosh2021@iem.edu.in,
subhojit.ghosh2021@iem.edu.in, anupam.mondal@iem.edu.in

Abstract. Text summarization is the process of condensing lengthy texts into concise and coherent summaries, capturing the main points of the document. It presents a significant challenge in machine learning and natural language processing (NLP) due to the vast volume of digital data available. There is a growing demand for algorithms capable of automatically condensing extensive texts into accurate and understandable summaries to effectively convey the intended message. Machine learning models are typically trained to comprehend documents, extracting essential information to produce the desired summarized output. The application of text summarization offers several advantages, including reduced reading time, accelerated information retrieval, and efficient storage of more information. In NLP, two primary methods for text summarization exist: extractive and abstractive. The extractive approach identifies key phrases within the source document and assembles them to form a summary without altering the text's content. The abstractive technique paraphrases and condenses sections of the source document. In deep learning applications, abstractive summarization can overcome grammar inconsistency issues often encountered in the extractive method. Our analysis has examined various existing methods for text summarization, including unsupervised, supervised, semantic, and structure-based approaches, and has critically assessed their potential and limitations. Specific challenges highlighted include anaphora and cataphora problems, interpretability issues, and readability concerns for long texts.To address these challenges, we propose solutions aimed at improving the quality of the dataset by addressing outliers through the integration of corrected values obtained from human-generated inputs. As research in this domain progresses, we anticipate the emergence of innovative breakthroughs that will contribute to the seamless and accurate summarization of lengthy textual documents.

Keywords: Summarization, Text Summary, Automatic Text Summarization, Natural Language Processing, Abstractive Method, Extractive Method, Deep learning Approach, Unsupervised Method, Supervised Method, Anaphora Problem, Cataphora Problem, Semantic Approach, Structure Based Approach, Machine Learning, Readability Concern

1 Introduction

Summarization is the process of separating the key bits from a larger piece of material while retaining the core ideas and concepts. It is required in a variety of settings. In the world of journalism, news pieces are frequently abbreviated to highlight the most important aspects of an event, allowing readers to stay informed despite their hectic schedules. It allows researchers to quickly comprehend the important findings and techniques of relevant studies, allowing for a more efficient examination of the current literature. Long business reports are synthesized into simple summaries in the corporate environment for executives who need a quick overview before making critical choices. Summarization algorithms are used by social media platforms to give users condensed updates that capture the substance of discussions.

Text condensation techniques rely on diverse methodologies to distill key ideas and concepts from larger textual content. The fuzzy logic approach leverages fuzzy set theory and fuzzy logic principles to handle imprecise or ambiguous information present in texts. Concept-driven methods aim to pinpoint and extract pivotal notions from the text, generating summaries based on the identified concepts and their interrelationships. Latent-semantic techniques, such as latent semantic analysis (LSA) or latent Dirichlet allocation (LDA), seek to unveil the underlying semantic patterns inherent in the text. Machine learning algorithms, including Naive Bayes, Decision Trees, or Support Vector Machines, are trained on labeled datasets to discern patterns conducive to summarization. Neural network architectures, encompassing deep learning models like Recurrent Neural Networks (RNNs) or Transformers, are harnessed to learn representations and generate summaries. Conditional Random Fields treat summarization as a sequence labeling task, employing CRFs to identify and extract salient sentences or phrases. Tree-based approaches, such as Tree Summarization or Rhetorical Structure Theory, scrutinize the discourse structure of the text to pinpoint pertinent information. Template-driven methods rely on predefined templates or schemas to extract relevant details from the text and generate summaries based on the populated templates. Rule-based techniques

COMPLETED WORK

- Customizing a pre-trained BERT model for text summarization task.
- Using a curated dataset of 200,000 [article, summary] pairs from diverse sources.
- Fine-tuning phase to adapt BERT's language understanding capabilities for summarization.
- Employing techniques like hyperparameter tuning, gradient clipping, and warm-up.
- Utilizing attention visualization methods for interpretability.
- Integrating graph neural networks to capture semantic relationships in text.
- Incorporating attention mechanisms to better understand contextual nuances.
- Aim is a robust, versatile model for high-quality summaries across domains, advancing summarization technology.

ONGOING WORK

- Customizing a pre-trained BERT model for text summarization task.
- Using a curated dataset of 200,000 [article, summary] pairs from diverse sources.
- Fine-tuning phase to adapt BERT's language understanding capabilities for summarization.
- Employing techniques like hyperparameter tuning, gradient clipping, and warm-up.
- Utilizing attention visualization methods for interpretability.
- Integrating graph neural networks to capture semantic relationships in text.
- Incorporating attention mechanisms to better understand contextual nuances.
- Aim is a robust, versatile model for high-quality summaries across domains, advancing summarization technology.

Breaking the 200,000 dataset into segments of 4 x 50,000 for easier training

```
In [2]: import pandas as pd

In [3]: df = pd.read_csv('train.csv', nrows=50000)

In [4]: df = df.rename(columns={'highlights': 'summary'})
        df = df.drop(columns=['id'])

In [5]: df.columns

Out[5]: Index(['article', 'summary'], dtype='object')

In [6]: num_rows = df.shape[0]

In [7]: df.to_csv("f50k.csv", index=False)

In [ ]:
```

Training and performing BERT on the first 50k sets of data

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('train50k.csv')

In [3]: df.columns
Out[3]: Index(['article', 'summary'], dtype='object')

In [4]: null_values = df.isnull().sum()
        null_values
Out[4]: article    0
        summary    0
        dtype: int64

In [5]: from sklearn.model_selection import train_test_split

In [6]: # Split the DataFrame into features (X) and target (y)
        X = df['article']
        y = df['summary']

        # Split the data into training (80%) and the rest (20%)
        X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.2, random_state=42)

        # Further split the rest into validation (50%) and testing (50%)
        X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)

        # Now we have X_train, y_train for training
        # X_val, y_val for validation
        # X_test, y_test for final testing

In [7]: from transformers import BartTokenizer, BartForConditionalGeneration, AdamW
        from tqdm import tqdm
        import torch

In [8]: tokenizer = BartTokenizer.from_pretrained("facebook/bart-large-cnn")

In [9]: train_encodings = tokenizer(list(X_train), truncation=True, padding=True, max_length=512, return_tensors="pt"

In [10]: model = BartForConditionalGeneration.from_pretrained("facebook/bart-large-cnn")
```

```
In [9]: rain_encodings = tokenizer(list(X_train), truncation=True, padding=True, max_length=512, return_tensors="pt", return

In [10]: model = BartForConditionalGeneration.from_pretrained("facebook/bart-large-cnn")

In [11]: model.train()

Out[11]: BartForConditionalGeneration(
           (model): BartModel(
             (shared): Embedding(50264, 1024, padding_idx=1)
             (encoder): BartEncoder(
               (embed_tokens): Embedding(50264, 1024, padding_idx=1)
               (embed_positions): BartLearnedPositionalEmbedding(1026, 1024)
               (layers): ModuleList(
                 (0): BartEncoderLayer(
                   (self_attn): BartAttention(
                     (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
                     (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
                     (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
                     (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
                   )
                   (self_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
                   (activation_fn): GELUActivation()
                   (fc1): Linear(in_features=1024, out_features=4096, bias=True)
                   (fc2): Linear(in_features=4096, out_features=1024, bias=True)
                   (final_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)

In [12]: optimizer = AdamW(model.parameters(), lr=5e-5)

         /Users/aritra/anaconda3/lib/python3.10/site-packages/transformers/optimization.py:306: FutureWarning: This implemen
         tation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.A
         damW instead, or set `no_deprecation_warning=True` to disable this warning
           warnings.warn(

In [13]: y_train_tensors = tokenizer(list(y_train), truncation=True, padding=True, max_length=128, return_tensors="pt", retur

In [*]: # Training loop
        batch_size = 4
        num_epochs = 3
        for epoch in range(num_epochs):
            for i in tqdm(range(0, len(train_encodings["input_ids"]), batch_size)):
                # Clear gradients
                optimizer.zero_grad()
```

```
y_train_tensors = tokenizer(list(y_train), truncation=True, padding=True, max_length=128, return_tensors="pt", retur
```

Since BART doesn't have an in-built train function we push the model into a training loop of epoch = 3 and batch size = 4

```python
# Training loop
batch_size = 4
num_epochs = 3
for epoch in range(num_epochs):
    for i in tqdm(range(0, len(train_encodings["input_ids"]), batch_size)):
        # Clear gradients
        optimizer.zero_grad()

        # Get batch inputs
        batch_inputs = {key: val[i:i+batch_size] for key, val in train_encodings.items()}
        batch_labels = {key: val[i:i+batch_size] for key, val in y_train_tensors.items()}

        # Shift labels to the right
        decoder_input_ids = batch_labels["input_ids"].clone()
        decoder_input_ids[:, :-1] = decoder_input_ids[:, 1:]
        decoder_input_ids[:, -1] = tokenizer.pad_token_id

        # Forward pass
        outputs = model(**batch_inputs, decoder_input_ids=decoder_input_ids)

        # Compute loss
        loss = outputs.loss

        # Check if loss is None
        if loss is None:
            print("Warning: Loss is None!")
            continue

        # Backward pass
        loss.backward()

        # Update parameters
        optimizer.step()
```

Output from the trained Model

In [1]:
```
from transformers import pipeline

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")
```

```
Downloading: 100% 1.36M/1.36M [00:01<00:00, 1.39MB/s]
```

In [2]:
```
ARTICLE = """ New York (CNN)When Liana Barrientos was 23 years old, she got married in Westchester County, New York.
A year later, she got married again in Westchester County, but to a different man and without divorcing her first hu
Only 18 days after that marriage, she got hitched yet again. Then, Barrientos declared "I do" five more times, somet
In 2010, she married once more, this time in the Bronx. In an application for a marriage license, she stated it was
Barrientos, now 39, is facing two criminal counts of "offering a false instrument for filing in the first degree," r
2010 marriage license application, according to court documents.
Prosecutors said the marriages were part of an immigration scam.
On Friday, she pleaded not guilty at State Supreme Court in the Bronx, according to her attorney, Christopher Wright
After leaving court, Barrientos was arrested and charged with theft of service and criminal trespass for allegedly s
Annette Markowski, a police spokeswoman. In total, Barrientos has been married 10 times, with nine of her marriages
All occurred either in Westchester County, Long Island, New Jersey or the Bronx. She is believed to still be married
Prosecutors said the immigration scam involved some of her husbands, who filed for permanent residence status shortl
Any divorces happened only after such filings were approved. It was unclear whether any of the men will be prosecute
The case was referred to the Bronx District Attorney\'s Office by Immigration and Customs Enforcement and the Depart
Investigation Division. Seven of the men are from so-called "red-flagged" countries, including Egypt, Turkey, Georgi
Her eighth husband, Rashid Rajput, was deported in 2006 to his native Pakistan after an investigation by the Joint T
If convicted, Barrientos faces up to four years in prison.  Her next court appearance is scheduled for May 18.
"""
```

In [3]:
```
print(summarizer(ARTICLE, max_length=130, min_length=30, do_sample=False))
```

```
[{'summary_text': 'Liana Barrientos, 39, is charged with two counts of "offering a false instrument for filing in t
he first degree" In total, she has been married 10 times, with nine of her marriages occurring between 1999 and 200
2. She is believed to still be married to four men.'}]
```

Rogue Metrics

```
1  rouge_names = ["rouge1", "rouge2", "rougeL", "rougeLsum"]
2  rouge_dict = dict((rn, score[rn].mid.fmeasure ) for rn in rouge_names )
3
4  pd.DataFrame(rouge_dict, index = ['pegasus'])
```

	rouge1	rouge2	rougeL	rougeLsum
pegasus	0.015596	0.000296	0.015525	0.015562

Result = https://huggingface.co/spaces/TheCleverIdiot/Text_Summarization/

Dataset = https://mega.nz/file/sdBwHQRI#yykHpuwcHldDz-K_EG-qhXGcgvvJyO4eNydVMEkKWhs

GitHub Repo = https://github.com/TheCleverIdiott/Innovative_Project

FUTURE SCOPE

- Exploring the use of pre-trained LSTM and Pegasus models for text summarization.
- Training custom LSTM and Pegasus models using the curated dataset.
- Investigating an ensemble approach combining BERT, Pegasus, and LSTM architectures.
- Fine-tuning and integrating the distinct models into a unified ensemble model.
- Aiming to leverage the strengths of each architecture for superior performance.
- Ensemble model can capture semantic relationships, context, and long-range dependencies.
- Potential to generate more coherent and informative summaries.
- Multi-model strategy with tailored dataset to advance robust and generalizable summarization techniques.

CONSLUSION

This project represents a major stride towards advancing the cutting-edge field of text summarization through the meticulous development and optimization of a state-of-the-art model.

The systematic approach employed in this endeavor, encompassing thorough analysis, rigorous comparative evaluation, careful curation of a high-quality dataset, and iterative refinement of the model architecture, establishes a robust foundation for future advancements.

The invaluable insights and learnings gained throughout this comprehensive research initiative pave the way for the conception and realization of innovative summarization techniques, pushing the boundaries of what is achievable in this domain.

By addressing the evolving needs of information processing and knowledge extraction in the rapidly accelerating digital age, the outcomes of this project hold the potential to enable more efficient and effective summarization solutions, thereby revolutionizing the way we interact with and derive value from vast repositories of textual data.

THANK YOU!!!