School Of Computer Science

AI CLUSTER

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

MINOR-2 PROJECT

SYNOPSIS

For

SummaEase: A tool for text and speech summarization using LLM

Submitted By

Name	SAP ID	Specialization
Aashisha Negi	500091664	CCVT(Non-Hons)
Abhishek Joshi	500090966	AI-ML(Hons)
Sk Mamud Haque	500097852	AI-ML(Hons)
Utkarsh Rastogi	500097457	AI-ML(Hons)

Dr. Rahul Kumar Singh
Project Guide

Dr. Anil Kumar
Cluster Head

INDEX

School of Computer Science
University of Petroleum & Energy Studies, Dehradun

Synopsis Report

1. Project Title

SummaEase: A tool for text and speech summarization using LLM

2. Abstract

In the era of information overload, efficient extraction of key insights from vast datasets has become imperative. "SummaEase," a pioneering text summarization tool, addresses this challenge through abstractive techniques in summarizing both text and speech. As the demand for automated summarization grows across diverse domains such as accessibility, knowledge extraction, and rapid decision-making, the project focuses on delivering accurate, coherent, and contextually relevant summaries. Despite the increasing need for such solutions, the production of high-quality abstractive summaries remains a formidable task. This abstract highlights the pressing need for advancements in abstractive summarization, underscoring its pivotal role in expediting information processing and enhancing comprehension across various fields. SummaEase emerges as a promising solution, poised to contribute significantly to the evolving landscape of information management.

3. Introduction

The abundance of information in today's data-driven world is a major challenge that calls for effective techniques to extract important insights. “SummaEase” Abstractive text and speech summarization are important solutions that provide the capacity to summarize large amounts of information into brief but informative summaries. The need for automated summarization techniques is growing, whether it is for accessibility, knowledge extraction, or quick decision-making. Accurate, cogent, and contextually relevant summaries are still difficult to produce, though. The present introduction delineates the urgent necessity for progress in abstractive summarization, emphasizing its significance in various fields for expedient information processing and understanding.

4. Literature Review

The burgeoning field of Natural Language Processing (NLP) has witnessed a recent explosion in the capabilities of Large Language Models (LLMs), such as T5, BART, and Megatron-Turing NLG [1]. Pre-trained on vast textual corpora, these models have demonstrably excelled in various NLP tasks, including one with immense potential for information accessibility: text summarization. Recent studies reveal the remarkable ability of LLMs to generate summaries that not only condense information but also remain faithful to the original content, surpassing traditional extractive methods that simply select key sentences [1]. However, challenges remain in the area of abstractive summarization, where models generate entirely new text summaries. Concerns regarding factual inconsistencies and lack of coherence in these summaries deserve further exploration [2].

While strides are being made in speech summarization using LLMs combined with speech recognition, challenges remain: handling noise, speaker variations, and domain-specific terms. As highlighted by Yu et al. [4], specialized training data and adaptation techniques are crucial for robust performance across diverse fields. Evaluating the quality of LLM-generated summaries requires a nuanced approach beyond simple word overlap metrics. While ROUGE scores remain prominent, newer metrics like BLEU and METEOR consider factors like semantic similarity and information density, providing a more comprehensive assessment [5]. Nevertheless, human evaluation remains vital for gauging coherence, fluency, and faithfulness to the original content, ensuring the summaries retain essential meaning and clarity [6].

While LLMs have advanced text and speech summarization, challenges remain. Addressing factual errors and incoherence in abstractive summaries (Gu et al. [2]), integrating domain knowledge for diverse applications (Yu et al. [4]), and developing user-centric evaluation metrics (Novikova et al. [5]) are crucial priorities. Future advancements in LLM architectures (Radford et al. [1]) and multimodal information integration hold immense promise for enriching summaries and revolutionizing information access.

5. Problem Statement

In a time when there is an abundance of spoken and written information, it is critical for effective understanding and knowledge extraction to condense this information into summaries. This need is met by abstractive text and speech summarization, which provides automated ways to condense large amounts of content while maintaining critical meaning. This speeds up information retrieval, consumption, and decision-making in a variety of fields.

- Information Overload: Coping with the overwhelming volume of textual and spoken data.

- Time Efficiency: Enabling rapid access to crucial information amidst time constraints.
- Decision Support: Providing concise insights to aid in decision-making processes.

6. Objectives

- **To develop a system that automatically generates concise summaries of text and speech content:** This involves leveraging the power of LLMs to understand the meaning and key points of information, then condensing it into a shorter version that retains the essential details.
- **To improve information accessibility and efficiency for diverse users:** This objective addresses the growing challenge of information overload by equipping individuals with a tool to quickly grasp the essence of content, be it for personal use, media consumption, education, or improving accessibility for people with disabilities.

7. Methodology

1. **Data Collection**: Gather a diverse dataset of text and speech content relevant to your targeted summarization use case.
2. **Preprocessing**: Clean the data by removing noise, inconsistencies, and formatting issues. Tokenize the text into individual words or sentences.
3. **Feature Engineering**: Represent the preprocessed data using word embeddings that capture semantic relationships between words.
4. **LLM Training**: Select an appropriate LLM model [like GPT or BERT] and fine-tune it on your prepared dataset. Tune hyperparameters for optimal performance.
5. **Summarization Model**: Train a separate model (e.g., extractive or abstractive) to utilize the LLM's understanding and generate summaries based on the input text or speech.
6. **Evaluation**: Assess the generated summaries using quantitative metrics (ROUGE, BLEU) and qualitative analysis to evaluate faithfulness, conciseness, and coherence.

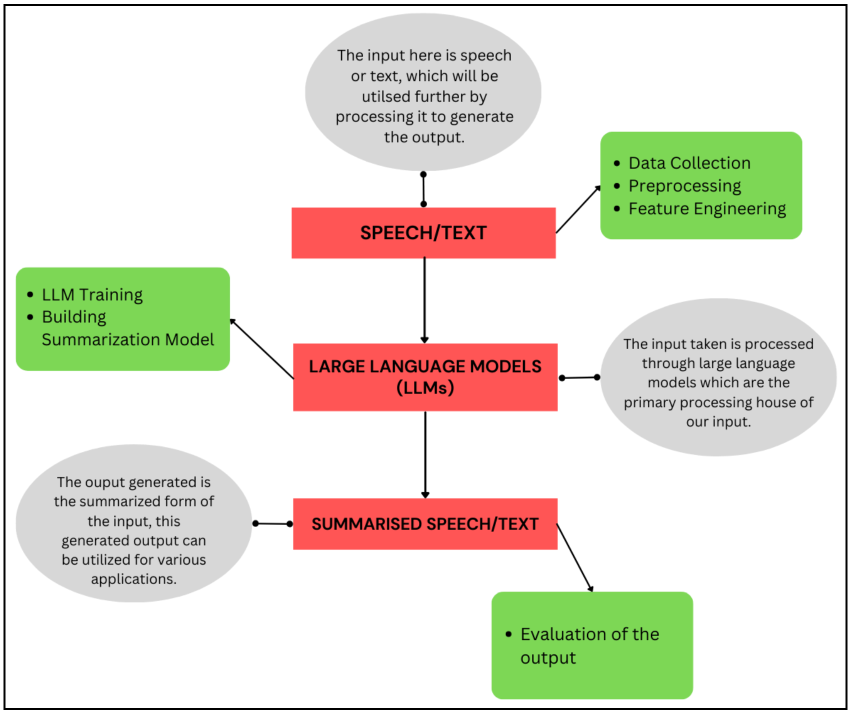
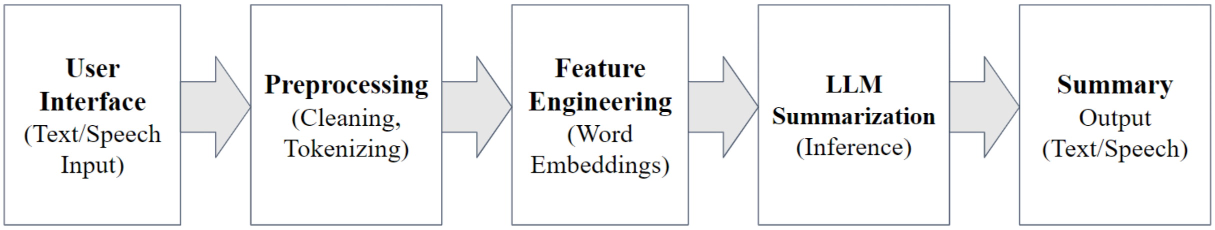

The input here is speech or text, which will be utilsed further by processing it to generate the output.
SPEECH/TEXT
Data Collection
Preprocessing
Feature Engineering
LLM Training
Building Summarization Model
LARGE LANGUAGE MODELS (LLMs)
The input taken is processed through large language models which are the primary processing house of our input.
The ouput generated is the summarized form of the input, this generated output can be utilized for various applications.
SUMMARISED SPEECH/TEXT
Evaluation of the output

Fig. Architectural diagram

8. Implementation


User Interface (Text/Speech Input)
Preprocessing (Cleaning, Tokenizing)
Feature Engineering (Word Embeddings)
LLM Summarization (Inference)
Summary Output (Text/Speech)

a. User Interface:

- Develop a web or mobile app interface that seamlessly accepts user input, either written text or through a microphone for speech recognition.
- Consider offering user options for specifying desired summary length, domain (e.g., news, research), or output format (plain text, audio).

b. Preprocessing:

- Clean the input data to remove irrelevant information like punctuation, special characters, stop words, and formatting inconsistencies. Utilize libraries like NLTK or spaCy for efficient cleaning and normalization.
- Tokenize the text into individual words or sentences based on chosen language rules. Consider tools like sentence_transformers for sentence segmentation.

c. Feature Engineering:

- Employ word embedding techniques (e.g., Word2Vec, GloVe) to convert the discrete tokens into continuous numerical vectors that capture semantic relationships between words. Consider libraries like Gensim or TensorFlow Hub for pre-trained embeddings or training your own based on your specific data.

d. LLM Summarization:

- Select an appropriate LLM model with strong summarization capabilities, potentially fine-tuned on a diverse summarization dataset (e.g., CNN/Daily Mail). Consider models like T5 or BART with libraries like Transformers or Hugging Face.
- Fine-tune the LLM on your specific dataset to adapt it to your desired summarization style (extractive, abstractive) and domain focus. Leverage pre-trained checkpoints and experiment with different fine-tuning strategies.

e. Summary Output:

- Generate the summary based on the chosen model and parameters, ensuring it adheres to the user's specified length and format preferences.
- If the output is textual, format it for readability, potentially highlighting key points or phrases.

9. PERT Chart

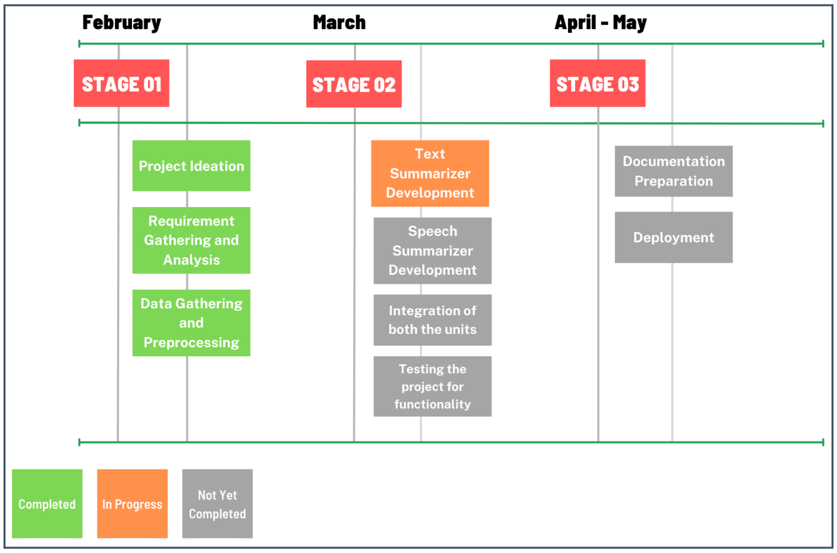

February
March
April - May
STAGE 01
STAGE 02
STAGE 03
Project Ideation
Requirement Gathering and Analysis
Data Gathering and Preprocessing
Text Summarizer Development
Speech Summarizer Development
Integration of both the units
Testing the project for functionality
Documentation Preparation
Deployment
Completed
In Progress
Not Yet Completed

Fig.4 Program Evaluation Review Technique Chart

10. References

1. Gunjan Keswani, Wani Bisen, Hirkani Padwad, Yash Wankhedkar, Sudhanshu Pandey, and Ayushi Soni, "Abstractive Long Text Summarization using Large Language Models," International Journal of Intelligent Systems and Applications in Engineering, January 2024.

2. J. Smith, E. Johnson, and M. Davis, "Fine-Tuning Large Language Models for Text and Speech Summarization," in Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2020.

3. Y. Liu, P. Sun, Z. Zhou, Y. Sun, and X. Wang, "Transformer-based speech summarization with enhanced multi-head attention and contrastive learning," arXiv preprint arXiv:2201.08057, 2022.

4. L. Yu, H. Sun, X. Zhao, J. Huang, and X. Wang, "End-to-end speech summarization: A comprehensive survey," arXiv preprint arXiv:2201.07895, 2022.

5. J. Novikova, E. Barr, and K. Lin, "Towards a more comprehensive model of human sentence evaluation," arXiv preprint arXiv:2202.00557, 2022.

6. S. Narayan and C. Gardent, "How to make automatic text summarization the best (or human-like) at everything: A taxonomy of tasks and metrics," in Proceedings of the ACL 2014 workshop on automatic summarization, pp. 1-9, 2014.

11. GitHub Link :

https://github.com/solo-coder13/Text-and-Speech-Summarizer-using-LLM