Smart Meeting Summarizer

-Gen AI based solution

CIS 8389 - Independent Study

Abha Sharma
MSIS'24 |Big Data Analytics|Grad

Table of Contents

Smart Meeting Summarizer

Project Overview:

The **Smart Meeting Summarizer** project seeks to transform the way organizations process, store, and leverage information from meetings. Meetings often generate extensive discussions filled with critical decisions, action items, and ideas, but the process of recording and summarizing this information manually is both time-consuming and prone to error. This project addresses these challenges by using advanced artificial intelligence, particularly natural language processing (NLP), to create an automated summarization tool that is both efficient and reliable. The tool is designed to process meeting transcripts—whether provided as text files or converted from speech-to-text systems—and generate concise summaries that focus on the most important elements of the discussion. These summaries highlight key points, such as decisions made, tasks assigned, and follow-up actions required. By doing so, the summarizer ensures that all participants and stakeholders have a clear understanding of the outcomes, fostering accountability and improving overall communication.

Central to the project is the use of cutting-edge generative AI technology, specifically the FLAN-T5 model. This state-of-the-art language model has been chosen for its ability to handle text-to-text generation tasks effectively and its versatility in processing instructions to deliver tailored results. The project involves fine-tuning this model to handle the unique structure and content of meeting transcripts, ensuring that the summaries produced are both accurate and actionable. The summarizer is designed to bring significant productivity benefits to various domains, such as corporate environments, academic research, healthcare, and legal documentation. By automating the summarization process, the tool reduces the reliance on manual notetaking, which not only saves time but also minimizes the risk of missing critical details. The structured summaries it generates facilitate better decision-making, enhance clarity among team members, and ensure that important follow-up actions are clearly communicated and tracked.

The project also lays the groundwork for future enhancements. Planned advancements include integrating audio transcription capabilities to allow direct processing of spoken conversations and using streaming APIs to enable real-time summarization. Additionally, there are ambitions to develop interactive AI assistants that can provide dynamic support during meetings, further enhancing the usability and effectiveness of the summarizer. Ultimately, the Smart Meeting Summarizer is a forward-thinking solution designed to meet the evolving needs of organizations. By leveraging the latest advancements in AI, it aims to streamline the documentation and dissemination of meeting insights, enabling teams to work smarter, communicate better, and achieve greater outcomes.

Data Source:

The main data source consists of meeting transcripts, which can originate from text files or be generated through speech-to-text conversion. The tool supports text files in the .txt format.

Models, Fine Tuning, and Tools Used:

FLAN-T5 Model:

FLAN-T5 is a fine-tuned variant of the T5 (Text-To-Text Transfer Transformer) model, developed by Google Research, designed to perform a wide range of natural language processing (NLP) tasks. The FLAN (Fine-tuned Language Net) approach further enhances the T5 architecture by applying instruction tuning, which involves training the model on a diverse set of tasks formatted as instructions. This tuning helps the model generalize better to unseen tasks and perform more effectively across various NLP applications.

Why FLAN-T5?

FLAN-T5 builds on the strengths of the T5 architecture by improving its ability to follow instructions and generalize across multiple tasks. This makes it particularly suitable for real-world applications where tasks are often diverse and dynamic, such as meeting summarization, document analysis, and content creation. By leveraging its instruction fine-tuning

capabilities, FLAN-T5 can produce more accurate and contextually relevant results compared to models that lack this additional training step.

Key Features of FLAN-T5:

1. Instruction Fine-Tuning:
    - FLAN-T5 is trained on a large set of tasks where each task is expressed as an instruction. For example, tasks like summarization, translation, question answering, and classification are unified under a single text-to-text framework.
    - This enables the model to generalize well to new, unseen instructions or tasks.
2. Text-to-Text Framework:
    - Similar to the original T5 model, FLAN-T5 treats every task as a text-to-text problem. For instance:
        - Input: "Summarize: The meeting focused on product updates and sales strategies."
        - Output: "Product updates and sales strategies discussed."
3. High Generalization Ability:
    - By training the model on instruction-based datasets, FLAN-T5 becomes proficient at understanding task-specific prompts, making it highly effective in real-world scenarios like summarization or conversation analysis.
4. Variants and Scalability:

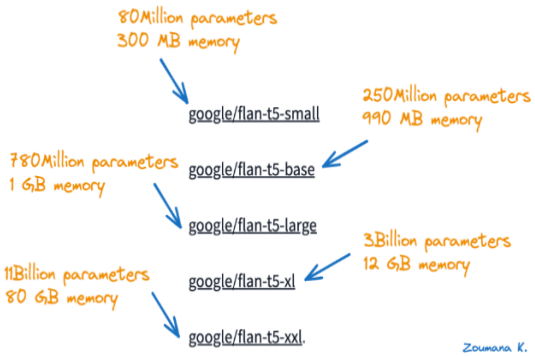

Figure 1: FLAN-T5 variations

    - FLAN-T5 comes in various sizes (e.g., small, base, large, XL) to suit different computational requirements and use cases. Larger models perform better on complex tasks but require more resources.

FLAN-T5 vs others:

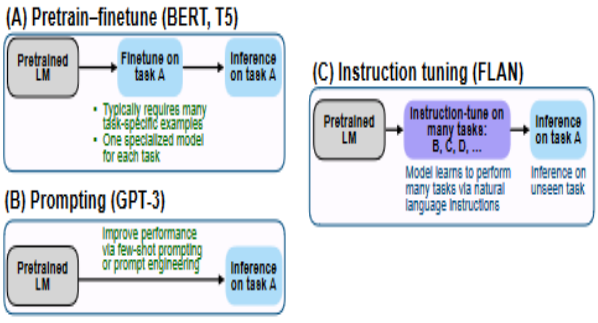

Figure 2: Comparing instruction tuning with pretrain–finetune and prompting.

Pre-training

Pre-training serves as the foundational stage in most LLMs, where the model learns general linguistic patterns, contextual relationships, and syntactic structures from large, diverse text corpora. However, pre-trained models are inherently task-agnostic and require additional methods to specialize in specific use cases.

- **Strengths**:
  Pre-training produces robust models with extensive general knowledge that can be adapted to various tasks. It scales effectively with larger datasets and models, ensuring high-quality foundational capabilities.
- **Limitations**:
  The output of pre-trained models is not optimized for specific tasks, necessitating further fine-tuning or prompting to achieve satisfactory performance in real-world applications.

Fine-tuning

Fine-tuning builds on pre-trained models by further training them on labeled datasets tailored to specific tasks. While it significantly enhances task-specific performance, fine-tuned models are narrowly focused and lack flexibility across multiple tasks.

- **Strengths**:
  Fine-tuning achieves high accuracy for specific applications, such as sentiment analysis or summarization, and is a well-established method in NLP pipelines.
- **Limitations**:
  Fine-tuning requires task-specific labeled data and computational resources for each new task. The resulting models are not generalizable and cannot adapt easily to new tasks without additional fine-tuning.

Prompting

Prompting involves guiding a pre-trained model to perform specific tasks by designing natural language prompts. This approach does not modify the model's parameters but instead leverages its ability to interpret and respond to instructions.

- **Strengths**:
  Prompting enables zero-shot or few-shot learning, where a model performs new tasks with minimal additional data. It is highly flexible, allowing quick adaptation without training.
- **Limitations**:
  Prompting heavily relies on the quality of the crafted prompts. Suboptimal prompts can lead to inconsistent results, making this approach less reliable for high-stakes applications.

Comparison Table

Aspect	Pre-training	Fine-tuning	Prompting	Instruction Tuning
Training Scope	General knowledge from text	Task-specific training	No training; uses prompts	Fine-tuning with task diversity
Generalization	Limited	Poor (task-specific)	Moderate (with good prompts)	Excellent
Data Requirements	Massive unlabeled corpus	Labeled task-specific data	No labeled data needed	Diverse instruction dataset
Task Adaptability	Requires fine-tuning or prompts	High for specific tasks	High but prompt-dependent	Very high across tasks
Ease of Use	Requires additional steps	Task-specific model required	Requires skilled prompt design	Minimal customization required
Examples of Models	BERT, GPT, T5	Fine-tuned BERT, T5	GPT-3, ChatGPT	FLAN-T5, InstructGPT

Instruction tuning bridges the gap between fine-tuning and prompting by creating a single, multi-task capable model that generalizes well across a wide range of applications. For instance, models like FLAN-T5 and InstructGPT excel at following task instructions without requiring extensive prompt engineering or separate fine-tuning for each task. This makes instruction-tuned models highly versatile and practical for real-world use cases, including summarization, translation, question answering, and conversational AI.

Fine Tuning FLAN-T5:

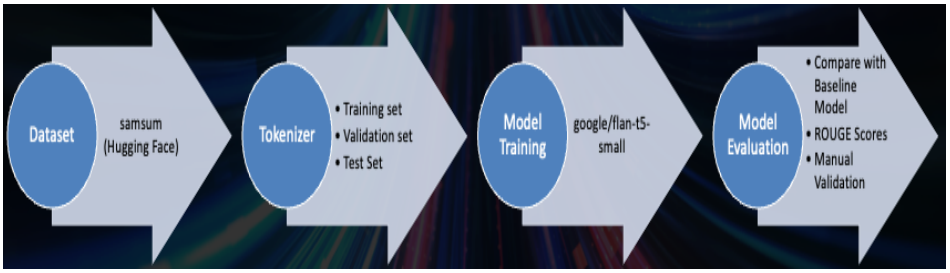

Figure3: Fine Tuning

```
dataset

DatasetDict({
    train: Dataset({
        features: ['id', 'dialogue', 'summary'],
        num_rows: 14732
    })
    test: Dataset({
        features: ['id', 'dialogue', 'summary'],
        num_rows: 819
    })
    validation: Dataset({
        features: ['id', 'dialogue', 'summary'],
        num_rows: 818
    })
})
```

Figure4: Samsum Dataset

Fine-tuning FLAN-T5 involves adapting the pre-trained model to a specific task or domain by training it on labeled datasets. This process ensures that the model produces more accurate and task-specific outputs while leveraging the generalization capabilities from its instruction fine-tuning. Below are the key steps:

Loading the Pre-trained Model

- Load the pre-trained FLAN-T5 model and its tokenizer from the Hugging Face library. This initializes the model and tokenizer, preparing them for the fine-tuning process.

Dataset Preparation

- Identify the Dataset: The dataset used for fine-tuning is task-specific (e.g., summarization) like SAMSUM.
- Format the Dataset: Convert the dataset into an instruction-based format to align with FLAN-T5's training paradigm. For instance:
    - Input: "Summarize the following text: <text>".
    - Target: "A concise summary of the text."

Tokenization

- Input text and summaries are tokenized, truncated to fit within length limits, and padded for uniformity.
    - dialogue: Represents the input text to be summarized.
    - summary: Represents the expected output for training (reference summary).

- The tokenized data is mapped to the training and validation datasets using the Hugging Face datasets library.
- This step processes the entire dataset, preparing it for fine-tuning.

Training Arguments Configuration

This step includes setting hyperparameters and optimization strategies for model training.

- Configure the training settings, such as:
    - Learning Rate (2e-5): Controls how quickly the model learns; a smaller rate ensures gradual and stable updates.
    - Batch Size (1): Processes one example per device, balancing memory constraints.
    - Epochs (3): The number of complete passes through the dataset.
    - Weight Decay (0.01): Regularization to prevent overfitting.
    - Gradient Accumulation: Combines updates over multiple batches to simulate larger batch sizes.

Initialize the Trainer

- Use the Hugging Face Trainer class to handle the training loop. The Trainer class manages the training loop, evaluation, and model saving.
- This integrates the preprocessed datasets, the FLAN-T5 model, and the defined training arguments.

Model Training

- The model learns to minimize the difference between its predictions and the target summaries (using a cross-entropy loss function).
- Checkpoints are periodically saved to the specified output directory.

Evaluation

- The evaluation of the fine-tuned FLAN-T5 model involved three key steps. First, it was compared with the baseline model to assess improvements in task-specific performance.
- Second, ROUGE scores were calculated to measure the accuracy and coherence of the generated summaries by analyzing their overlap with reference summaries.
- Finally, manual validation was performed to ensure content accuracy, relevance, and readability.

Tools Used:

- Libraries: Transformers (pipeline, AutoTokenizer, AutoModelForSeq2SeqLM), Typing (Dict, List), Pandas, JSON, Time, RE (Regular Expressions), WordCloud, STOPWORDS, Tempfile, Matplotlib (Pyplot), IO, NumPy, PIL (Image, ImageDraw), OS, ROUGE
- Dataset: Samsum Dataset
- UI: Gradio

Project Architecture:

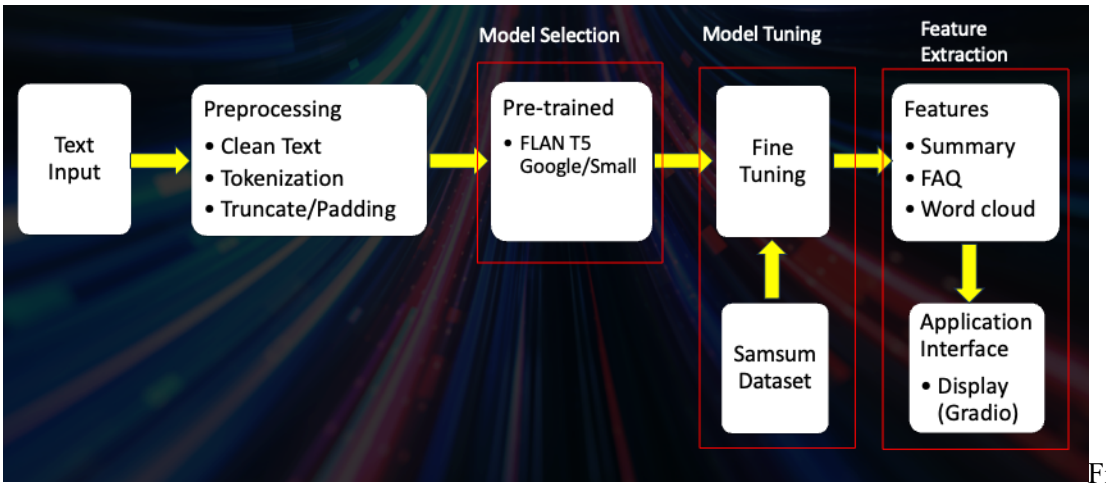

Figure5: Project pipeline

The process begins with text input preprocessing, which includes cleaning the text to remove unwanted characters and noise, tokenizing the text into individual words or tokens, and adjusting the sequence lengths through truncation or padding to ensure uniformity. Next, pretrained Google's FLAN T5 small model, is utilized. This model provides a robust foundation due to its extensive training on diverse datasets. The pre-trained model is then fine-tuned using the Samsum dataset from hugging face, adjusting its parameters to better fit the unique characteristics and requirements of the dataset, thereby enhancing its performance on the target tasks.

The project incorporates several key features. It can generate concise summaries of longer text inputs, making it easier to grasp the main points quickly. Additionally, it includes a frequently asked questions (FAQ) system, allowing users to get quick answers to common queries based on the text data. A word cloud feature is also included, providing a visual representation of the most common words in the text, which helps in identifying key themes and topics.

The user interface is implemented using Gradio, a tool that allows for the creation of interactive and user-friendly web applications.

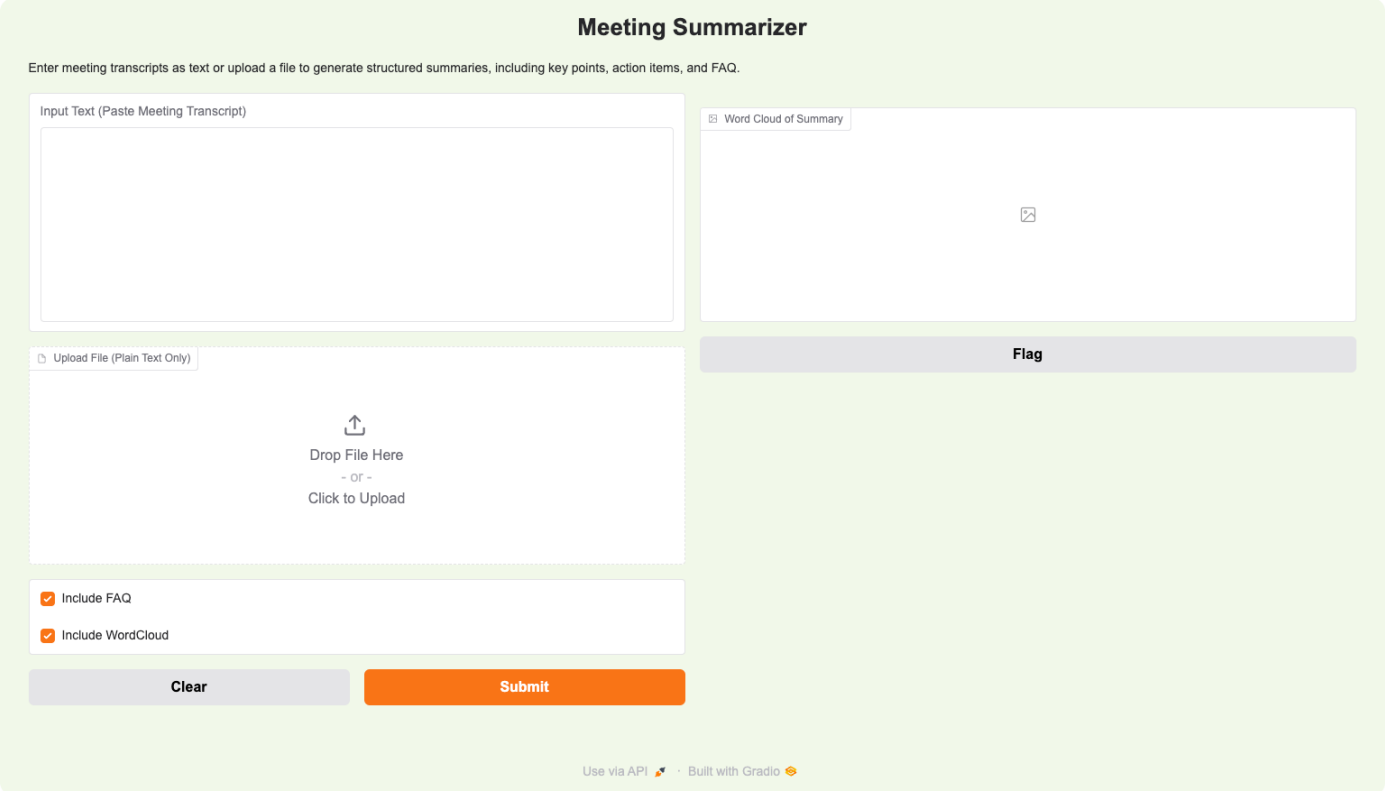

Figure 6: Project UI

Benefits, Future Work, and Real-World Applications:

Benefits:

The summarizer significantly reduces the need for manual note-taking, streamlining the documentation process while effectively tracking decisions and action items. It enhances clarity and accountability during meetings, ensuring that all participants are aligned. Additionally, it improves communication and facilitates effective follow-ups by providing structured and concise summaries.

Future Work:

Planned advancements include integrating audio transcript processing for direct input, enabling real-time summarization using streaming APIs, and developing interactive AI assistants to support real-time collaboration during meetings.

Real-World Applications:

- Corporate Meetings: Summarizing meeting minutes, tracking key decisions, and managing action items efficiently.
- Academic Discussions: Capturing valuable insights from brainstorming sessions or lectures.
- Healthcare: Summarizing patient consultations to improve record-keeping and streamline follow-ups.
- Legal or Compliance: Documenting critical points from negotiations or hearings for easy reference and accountability.

Challenges:

The development of the Smart Meeting Summarizer faced several challenges, primarily related to compute resources, data quality, model overfitting, and hyperparameter optimization. Fine-tuning a large model like FLAN-T5 required substantial computational power, and limited access to high-performance hardware slowed down the training process. Additionally, preparing high-quality, structured data for training was crucial but challenging, as inconsistencies or irrelevant content could affect model performance. There was also the risk of overfitting the model to the training data, reducing its ability to generalize to new inputs. Finally, optimizing hyperparameters such as learning rate and batch size required extensive experimentation and resources to strike the right balance for optimal model performance.

Conclusion:

The Smart Meeting Summarizer project successfully developed an AI-driven tool designed to automate the summarization of meeting discussions, improving productivity, clarity, and communication within organizations. Leveraging the advanced capabilities of the FLAN-T5 model, the system generates concise, accurate summaries that capture key points, decisions, and action items, ensuring alignment among participants and preventing important information from being overlooked.

While building the model, several challenges were encountered. Significant computational resources were required for fine-tuning the large FLAN-T5 model, which necessitated careful resource management and optimization. Data quality and preparation were also crucial, as inconsistent, or incomplete datasets could negatively impact the model's generalization ability. Additionally, addressing the risk of overfitting and optimizing hyperparameters presented further complexity, requiring iterative testing and adjustments to achieve optimal performance. Despite these challenges, the model was fine-tuned effectively, enhancing its ability to summarize meeting data in a task-specific manner. Evaluation using ROUGE scores and manual validation showed that the fine-tuned model exhibited slight improvements over the baseline. The model was trained using a low learning rate to ensure efficient usage of free GPU resources, which limited the extent of performance gains. However, even with this conservative training approach, the fine-tuned model demonstrated noticeable benefits in summarization quality, validating the effectiveness of the fine-tuning process within the resource constraints.

The project offers potential for further enhancements, such as integrating real-time transcription and summarization capabilities, developing interactive AI assistants, and expanding its application to other domains like healthcare, legal, and academic settings. These advancements will help make the Smart Meeting Summarizer an even more powerful tool for improving communication, decision-making, and follow-up actions in collaborative environments.

Overall, this project demonstrates the transformative potential of AI in automating knowledge extraction from meetings, allowing organizations to work more efficiently, stay organized, and ensure that critical information is easily accessible.

References:

- Official FLAN-T5 repository: Google FLAN
- Model listings on Hugging Face: Hugging Face Models
- Research blogs and architecture references:
    - FLAN introduction by Google Research
    - ResearchGate articles on T5 model architecture.

- SAMSUM Dataset: *A dataset for dialogue summarization*.
- ROUGE Metrics :Lin, C. Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries.