Report

# Project Title:

ThreadWise: An NLP-Powered Assistant for Conversation Summarization and Smart Replies

# Team Size:

Karthikeya Mittapalli – 22CSB0C14

Sai Kartik – 22CSB0A05

Sreeram – 22CSB0A29

# 1. Introduction

In today’s work environment, professionals frequently deal with an overwhelming volume of threaded communications through emails and chat platforms like Slack or Microsoft Teams. Navigating long threads to extract important details, assess urgency, and compose responses is time-consuming. This project aims to address that problem by building a smart assistant that automatically summarizes conversation threads, identifies message urgency, and suggests relevant replies using state-of-the-art Natural Language Processing (NLP) techniques and Large Language Models (LLMs).

# 2. Objectives

* Develop a system that takes threaded messages (emails or chats) as input and outputs concise summaries.
* Identify and flag threads that require urgent attention based on contextual cues.
* Generate context-aware, suggested replies to assist users in communication.
* Use pre-trained language models and prompt-based techniques to avoid traditional model training.
* Enable extensibility to multiple platforms (email, Slack, etc.) with a modular design.

# 3. Functional Requirements

* Input Handling: Ability to ingest threaded communication data (email/chat logs).
* Thread Grouping: Aggregate messages into coherent threads based on metadata.
* Preprocessing: Remove signatures, quoted replies, and redundant text to clean input.
* Summarization: Generate a concise summary of each message thread using an LLM.
* Urgency Detection: Identify urgency based on keywords, tone, and position in thread.
* Reply Suggestion: Generate relevant reply suggestions using context-aware prompts.
* Output Formats: Export results (summary, urgency flag, reply) in CSV or JSON format.

# 4. Non-Functional Requirements

* Portability: Should support deployment across email or messaging platforms.
* Scalability: Capable of processing large volumes of messages efficiently.
* Usability: Should be simple to integrate into existing workflows via scripts or API.
* Maintainability: Modular design for easy extension or adaptation to other platforms.
* Performance: Summarization and detection must be completed in reasonable time.
* Data Privacy: Must ensure personal or sensitive data is not leaked in processing.

# 5. Tools & Technologies (tentative)

* Programming Language: Python
* LLMs: OpenAI GPT, Gemini, or similar (via API or local models)
* NLP Libraries: spaCy, NLTK, or transformers (HuggingFace)
* Data Source: Enron Email Dataset, Slack Export (if available)
* Output: CSV/JSON for easy UI/API extension

# 6. Scope for Future Work

* UI integration for interactive use.
* Integration with real-time communication APIs (e.g., Gmail API, Slack API).
* Personalization of reply tone/style.
* Incorporating named entity recognition and task extraction.