Mail Management System: A Hybrid Prioritized Summarization Approach

*

Purushotham Kalangi, Sathvik Chowdary Samineni, Rakesh Kuchipudi, Kaleemunnisa, FNU
Computer Science
Seidenberg School Of CS and IS
Pace University
(pk98671n, ss63111n, rk24434n, klnu) @pace.edu

***Abstract*—In today's digital world, thousands of emails are sent and received every day, overloading people and increasing the danger of missing important communications. Traditional email management software often fails to adapt to user-specific needs, making manual organizing time-consuming and inefficient. To solve this issue, we present an AI-powered email summary and organizing system that improves email management through automation and customization. Using natural language processing (NLP) techniques to summarize emails within predefined time periods (daily, weekly, or monthly), extract critical information such as deadlines, events, and meetings, and automatically incorporate them. In addition, machine learning algorithms leverage user preferences and previous behavior to categorize and prioritize emails, ensuring that critical communications are easily accessible. Initial testing shows that our AI-powered solution, post-classifying (TF-IDF) the emails with the combination of extractive and abstractive summarization techniques, dramatically decreases the time spent on email management while enhancing the accuracy of vital information detection. Users report better efficiency and less clutter in the inbox, resulting in greaterin greater productivity. This solution highlights the AI's potential to change digital communication by decreasing information overload and streamlining productivity. Future enhancements will center on improving NLP models for better contextual comprehension and expanding integrations with third-party productivity tool and reasoning capabilities.**

***Index Terms*—Text classification, hybrid text summarization, TF-IDF, Random forest classifier, SMOTE, LSTM, Gmail API, Chrome Extension, OAuth2, Natural Languate Tool Kit(NLTK), Productivity, ROUGE, Transformer models (T5, BART), Mail management, Flask API, Browser-based, Low-latency**

I. INTRODUCTION

In the early 1970s, email was invented, and from then email became mainstream for communication by enabling fast,global, and efficient messaging, transforming business and personal interactions.Today, email is the main method we use for long-distance communication. More than 100 billion messages are received and sent every day, and that number is only expected to increase. As billions of messages are generated daily, the chance of missing messages by individual or organizations are at high risk.Missing the mails lead to a big loss for business or organizations or individuals.

Preventing the loss of messages or mails is crucial in the current world. Existing emails have lack of features that automate the generation of the mail report for daily or weekly. The smart mail mentor feature helps prevent messages from being lost.

Smart mail mentor is a natural language processing technique, artificial intelligence, machine learning, and deep learning. This feature automates daily or weekly emails and generates the report without losing emails.

II. LITERATURE REVIEW

Email management has become an essential task in the digital age of today, where individuals and organizations receive a large volume of emails daily. Efficiently managing and prioritizing emails is crucial to improving productivity and reducing information overload. Several studies have explored various approaches to automate email classification, prioritization, and summarization.

Machine learning (ML) techniques have played a significant role in email classification. Traditional approaches such as Naïve Bayes, Support Vector Machines (SVM), and Random Forest classifiers have been widely used to categorize emails into predefined categories such as spam, promotional, or business emails [1]. However, recent advances in deep learning, particularly long-short-term memory (LSTM) networks, have demonstrated improved accuracy in text-based classification tasks, including email filtering [2].

The TF-IDF (Term Frequency-Inverse Document Frequency) and Random Forest approach has been particularly effective in feature extraction and classification. TF-IDF helps identify important words in emails, while Random Forest improves classification accuracy by leveraging an ensemble of decision trees [3]. Furthermore, LSTM-based models have shown superior performance in handling sequential text data, making them well suited for email classification [4].

One of the areas to focus on to summarize and classify emails was explored in the paper [5] which exposes abstrac-

Feature	Existing Tools (SaneBox, Superhuman, Cortana)	SMM (Our Tool)
Basic Email Customization	Yes	Advanced
Time-Based Reports/Summaries	No	Yes (Core Feature)
Email Categorization	Limited and Manual sifting through each email	Yes (6 Categories)
Reasoning Capabilities	No	Yes (Future Scope)
Lightweight & Performant	Varies	Yes

Fig. 1. Comparison Table

tive summarization using BERT-based extractive text summarization and BERT -GPT -2 hybrid models. We effectively managed to gather topic-aware summarization(T-BERTSum) that can potentially identify relevant summaries, enhancing organization and prioritization. Moreover, [5] suggests that by fine-tuning the BERT and GPT models in the paper, results could be obtained for the Q / A capabilities of our tool.

Interpreting the text and generating the human-like response by understanding the context and meaning by [6] tries to generate response by emphasizing both the abstractive and extractive approaches for more accurate and coherent summaries, this is helpful in giving the relevant information by using the technique that is seq-to-seq modelling using LSTM and encoder-decoder and however, [6] yields redundancy, contextual misinterpretation which support the previous implementations like [5] and prior methods.

Implementation of chrome-based extension requires optimization and smooth processing of information/mails, so is presenting the insights with the takeaways from [7] by reducing the API calls as fewer requests results in faster loading of report which is fundamental principle behind any system. By enabling HTTP/2 , HTTP/1.1 will reduce the latency and optimize our system in generating the reports and load faster in Chrome even on slower connections from the back-end post summarizing and prioritization.

Natural Language Processing (NLP) techniques have further enhanced email analysis by enabling text summarization and sentiment analysis. Transformer-based models such as BERT significantly improved contextual understanding in summarization tasks.

In comparison with existing tools Fig 1,such as SaneBox, Superhuman, and Cortana, the Smart Mail Mentor (SMM) offers significantly enhanced capabilities. While other tools provide only basic email customization, SMM supports advanced customization options. A standout feature of SMM is its core ability to generate time-based reports and summaries—something lacking in existing solutions. It also categorizes emails into six predefined categories, unlike the limited and manual categorization found elsewhere. Although reasoning capabilities are not yet present in most current tools, SMM plans to introduce them as part of its future scope. Additionally, SMM is designed to be lightweight and highly performant, making it a robust and efficient solution.

III. Ease Of Use

The Smart Mail Mentor extension is designed to enhance user experience by offering a seamless and automated ap-

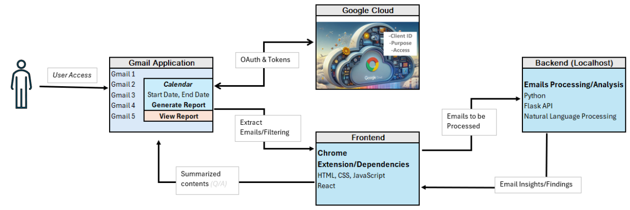

Fig. 2. System Architecture

proach to email management. The following features ensure ease of use:

Automated Email Classification: The system categorizes emails into predefined groups such as promotional, personal, business, order confirmation, and subscription renewals without user intervention. This classification is powered by ML models like LSTM and Random Forest, ensuring high accuracy and minimal manual effort.

Simple Setup and Minimal Configuration: Users only need to authenticate the extension using the Google Gmail API. Once connected, the system automatically fetches emails and processes them without requiring further input.

Prioritization of Important Emails: The extension filters emails based on urgency, sender importance, and content using NLP techniques, ensuring that users can focus on high-priority messages first.

Concise Email Summarization: Instead of manually reading multiple emails, users receive a daily or weekly report with AI-generated summaries, significantly reducing time spent on email management.

Seamless Browser Integration: As a browser extension, Smart Mail Mentor runs in the background without disrupting user workflow. It provides real-time insights without requiring additional software installations.

From the Fig 2, Users access their Gmail through a secure authentication process using OAuth and tokens provided by Google Cloud. After selecting a date range, relevant emails are extracted and filtered. These emails are then passed to a Chrome Extension frontend built with React, HTML, CSS, and JavaScript. The frontend communicates with a backend server running locally, which uses Python, Flask, and Natural Language Processing (NLP) to process and analyze the email content. The resulting insights and summarized information are returned to the frontend and presented to the user in a clear, easy-to-understand format. By integrating AI-powered automation, Smart Mail Mentor simplifies email management while enhancing productivity, ensuring a smooth and user-friendly experience.

IV. Methodology

1) *Front-end Overview*
Our tool, a.k.a Smart Mail Mentor, was developed using a methodical technique that solves the growing problem of email overload through the use of Natural Language Processing (NLP) and intelligent automation.

The procedure begins with email data extraction via connection with the Gmail API, where users login using OAuth2. After choosing a time window (daily, weekly, or monthly), the Chrome extension retrieves emails from the user's inbox using particular Gmail query parameters. These emails are then analyzed to extract key metadata, including the sender, recipient, topic, and date, as well as the email content in plain text and HTML forms.

Following parsing, the emails are transmitted as a JSON payload to a Flask-based back-end API for processing. The email summarizer module is at the heart of the back-end logic; it handles data cleaning, text preparation using NLTK, and vectorization with TF-IDF to assess term relevance. Advanced transformer-based models (such as BART or T5 from HuggingFace) are used for summarization, resulting in compact and cohesive summaries. Additional NLP approaches are used to discover critical information, such as deadlines, meeting details, and events, which are then integrated into calendar systems. To guarantee quality and relevance, each summary is reviewed using ROUGE criteria.

The backend responds with processed summaries and information, which the extension provides to the user in an easy-to-read report style. The system also includes smart features like customized summarizing time periods, email type classification (e.g., promotions, meetings, action-required), and prioritizing based on user behavior and previous interactions. In the future, the program will feature calendar integration and adaptive learning to fine-tune priority.

The entire pipeline is based on a strong tech stack that includes Flask, Python, NLTK, Scikit-learn, Transformers, and Torch, with the front end developed as a Chrome plugin that uses JavaScript and the Gmail API.

2) *Model Simulated Outcome*

   a) *Dataset Preparation and Filtering* A curated dataset of real-time 2000+ emails from inbox is sampled and preprocessed to exclude low-information content (i.e., emails with less than 10 words). This ensures that only meaningful textual content is used for analysis.
   b) *Extractive Summarization* We apply TF-IDF-based sentence ranking to identify the most informative sentence from each email. This lightweight and unsupervised approach guarantees real-time performance while maintaining contextual relevance, especially valuable in enterprise environments.
   c) *Abstractive Summarization (with Novel Batch Optimization)* Leveraging the T5-small transformer model, we summarize each email in an abstractive manner. A key innovation here is our batch-wise processing technique, which groups emails to reduce inference time and memory usage. The content is truncated to a maximum of 200 words to align with transformer input constraints, making the summarizer viable for large-scale deployments.

🔍 ROUGE Scores (0-1 scale, higher is better):

	rouge1	rouge2	rougeL
Extractive	0.490347	0.487448	0.490347
Abstractive	0.114064	0.104974	0.111525

Fig. 3. ROUGE METRICS

   d) *Dual-Pipeline Integration* Both summarization pipelines are executed in parallel using multi-threading with Python's ThreadPoolExecutor, significantly enhancing processing efficiency without compromising accuracy.
   e) *Quantitative Evaluation using ROUGE* To assess the quality of generated summaries, we utilize ROUGE-1, ROUGE-2, and ROUGE-L metrics. Our analysis highlights that extractive summaries score significantly higher than abstractive ones (*Fig 3*, ROUGE-1 F1: 0.49 vs. 0.11), emphasizing the precision of sentence-level extraction and the improvement potential for transformer-based models in domain-specific text
   f) *Visual and Interactive Insights* In addition to statistical evaluation, we introduce a rich HTML-based interface for visualizing summaries alongside the original emails. This makes human validation easier and supports iterative refinement for future deployments.

This technique, when combined, offers a seamless, AI-powered email assistant that assists users in effectively managing their inboxes and staying focused on what is most important.

Conclusions

The mail management system introduces a effective hybrid summarization and prioritization approach to streamline email management, addressing the increasing challenge of information/data overload in digital communication on a daily basis. By making use of Natural Language Processing (NLP), machine learning (Random Forest, LSTM), and TF-IDF-based feature extraction, the system effectively classifies, summarizes, and prioritizes emails into key categories (Promotional, Personal, Business, Order Confirmations, and Subscription Renewals). With respect to existing solutions, SMM provides automated time-based reports (daily, weekly, monthly) and extracts critical details (deadlines, meetings, events), significantly reducing manual effort to skim each email which is time consuming process.

Our Chrome extension integrates seamlessly with Gmail API, delivering top 5 prioritized emails per category directly to the user's frontend—a feature absent in competing tools appending the insights to the extension. The dual summarization pipeline (extractive + abstractive) ensures concise yet meaningful summaries in a short amount of time, validated through ROUGE metrics, with extractive methods outperforming abstractive ones in precision. Future enhancements and add-ons will focus on improving contextual reasoning capabilities, integrating adaptive learning for personalized prioritization based on the email data, and expanding third-party productivity tool integrations.

By blending AI-driven automation with user-friendly design, SMM boosts user productivity, minimizes inbox clutter, and sets a foundation for next-generation intelligent email assistants. The system's lightweight architecture, real-time processing, and scalability make it a robust solution for modern email management challenges with low-code development.

REFERENCES

[1] Mansoor RAZA, Nathail Dilshani Jayasinghe and Muhana Magbouli Ali Muslam ”A Comprehensive Review On Email Spam Classification using Machine Learning Algorithms” Publisher IEEE, 2021.

[2] Palwasha Bhatti,Zunera Jalil,Asma Majeed ”Email Classification using LSTM: A Deep Learning Techniques”Publisher IEEE,2021

[3] Lorelyn F.Adrales,Ariel M Sison ”Deep Learning Approaches to a School-Email Classification With TF-IDF-based vectorization” Publisher IEEE,2024.

[4] A. S. U. R. K. S. Harish and P. S. Bhat, ”Email classification using deep learning with LSTM,” Procedia Computer Science

[5] G. U. Kiran, G. B. Desale, R. Gandi, C. U. Rao, M. Lavanya, and B. V. Reddy, “Deep Learning Based Abstractive Text Summarization: A Survey,” 2024 Parul International Conference on Engineering and Technology (PICET), pp. 1–8, 2024.

[6] K. S. Kavyashree, T. S. V, S. R, and S. R, ”Survey on Automatic Text Summarization using NLP and Deep Learning,” 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS), pp. 523-527, 2023.

[7] L. Cegan, ”Improving the performance of loading web pages in the environment with high latency connections,” Dept. Inf. Technol., Fac. Elect. Eng. Informatics, Univ. Pardubice, Pardubice, Czech Republic, 2015.

[8] DeepSeek, ”DeepSeek Open Source Language Models,” DeepSeek, 2024. [Online].

[9] OpenAI, ”ChatGPT: Optimizing Language Models for Dialogue,” OpenAI, [Online].

[10] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, A., and Levy, O., ”BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” Proceedings of ACL 2020, pp. 7871–7880, 2020

[11] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P., ”Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5),” Journal of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020

[12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Duchesnay, E., ”Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011