AI-Powered Personal Assistant for Smart Task Scheduling, Email Deadline

Kaustubh Gulwade, Srishti Gupta, Riya Gupta, Chaitanya Hapse, Swapnil Gulbhile, Prof. Rahul Waikar, Prof. Sunil Sable

Department of Engineering Science and Humanities, Vishwakarma Institute of Technology, Pune, India

DOI : https://doi.org/10.51583/IJLTEMAS.2026.150100020

Received: 08 January 2026; Accepted: 13 January 2026; Published: 24 January 2026

ABSTRACT

The explosion of electronic communication has made email a foundation of contemporary professional and educational existence. Nevertheless, the large amount of unstructured correspondence places an enormous cognitive burden on the users, resulting in ineffective task processing, lost deadlines, and lowered productivity. This paper presents an intelligent personal assistant aimed at addressing these issues. The system combines a cutting-edge large language model (LLM) with general productivity APIs, such as Gmail and Google Calendar, to develop a seamless, automated process. The system feeds on and parses email content automatically to carry out three fundamental functions: creating brief summaries for rapid understanding, extracting action items and deadlines to schedule matching events in an electronic calendar, and offering a question-answering interface where users can pose particular questions about an email's content. A strict experimental analysis carried out on a manually annotated subset of the Enron email corpus proves the effectiveness of the system. The system attained an F1-score of 0.87 for extracting tasks and deadlines and ROUGE-L scores of 0.42 for summarization, reflecting high-quality performance. Qualitative analysis also supports the question-answering module's capability to correctly retrieve information. The main contribution of this project is the conception, implementation, and evaluation of a strong, end-to-end system that adequately closes the semantic gap between passive email information and actionable, queryable intelligence, offering an effective solution for individual digital productivity enhancement.

Keywords— AI Productivity Tool, Automated Scheduling, Calendar Automation, Email Summarization, Large Language Models (LLMs), Natural Language Processing (NLP), Question Answering, Task Extraction, Langchain agents, gmail agent, calendar agent, gmail and calendar ai assistant, mail query asking system.

INTRODUCTION

In Today's digital environment, email and web-based calendars are ubiquitous aids to planning and managing work, educational, and personal life. The volume of communications arriving each day, from meeting invitations and project notifications through deadlines and casual plans, imposes a heavy responsibility on individuals to sift through, sort, and respond to this information by hand. This continuous flow of unstructured information presents an enormous problem of information overload and task fragmentation. Though the tools required—email clients and calendar apps—are omnipresent, they mostly function in solo silos. This requires repeated, manual information transfer, whereby users read an email, extract the essential particulars of a task or event, and then painstakingly re-key that information into a distinct calendar app.

This manual process creates a fundamental "semantic gap" between the unstructured, natural-language content of an email and the structured, machine-readable data required by scheduling software. The cognitive overhead associated with bridging this gap is non-trivial; it consumes valuable time, introduces the potential for human error, and contributes to mental fatigue, ultimately hindering productivity. The challenge, therefore, is not the absence of tools, but the lack of intelligent automation to seamlessly integrate them.

To overcome these constraints, this paper presents a new, intelligent personal assistant conceived as a comprehensive solution to the task fragmentation problem. It capitalizes on the sophisticated semantic reasoning

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

powers of contemporary large language models (LLMs) to close the gap between loose email text and formal calendar events. The system automatically scans in emails and carries out three main functions: it creates brief summaries for quick understanding (The system can generate summaries based on user-specified ranges (e.g., day-wise, week-wise, or a fixed number of emails).), identifies primary task-oriented entities like deadlines and meeting information to automatically fill in the user's electronic calendar, and provides an interactive question-answering feature, enabling users to ask email content questions for certain specifics without having to re-read the entire message.

The rest of this paper is structured as follows. Section II presents related work on personal digital assistants, natural language information extraction, and automatic scheduling systems. Section III states the research gap and introduces the new contributions of this work. Section IV describes the system architecture and approach. Section V summarizes the implementation details and workflow of data processing. Section VI describes the experimental design, evaluation criteria, and results. Section VII addresses the meaning of such results and the limitations of the system. Then, Section VIII summarizes the paper and Section IX indicates directions of future work.

RELATED WORK

The design of this system draws on four decades of work in three converging fields: the history of personal digital assistants, natural language processing of unstructured text, and automated scheduling system design.This introduction places our research in the context of these fields to identify its new contributions.

The history of personal digital assistants

The idea of a digital assistant was first conceived in the form of early rule-based conversational programs such as ELIZA, which mimicked conversation using straightforward pattern matching. The historical progression from such early systems to contemporary AI-powered voice assistants like Amazon Alexa and Apple's Siri indicates an inherent paradigm shift in human-computer interaction. This development has progressed from systems in which users had to master the formal syntax of a machine to systems today in which the machine needs to understand the subtleties of human .

Nonetheless, even as they become more advanced in responding to direct commands, one persistent weakness of mainstream business aides is their superficial contextual understanding, specifically with regards to asynchronous, text-based communication such as email. They exist mainly in a reactive, command-and-response capacity and don't have the ability to proactively analyze and take action on the enormous body of information in a user's inbox. Current research into user attitudes towards AI assistants suggests that users appreciate convenience and efficiency but also want more proactive and responsive help. This increased expectation of context-aware automation provides a definite research gap, which this research aims to bridge by equipping the personal assistant with the capability to comprehend and take action on the rich context within email.

Natural language processing for information extraction from email

The foundation of the system's intelligence is its ability to understand and interpret email text. Possessing a unique set of NLP challenges associated with data from an email, such as conversational streams and informal language, ambiguous entities or mentions in a message, and the mixing of different topics in one email message. Most initial work in this area focused on the NLP problem of spam filtering and semantic classification, using statistical methods or supervised machine learning, which are classical machine learning approaches.. The emergence of transformer-based models, like BERT and the GPT family, and strong LLMs that have emerged from this research, have changed the landscape. These pretrained models are created from vast text datasets and perform strongly in zero-shot and few-shot learning. They can perform complex tasks such as summarization and named-entity recognition with little or no task-specific data. Researchers have applied these summary models to the context of emails, demonstrating the ability to generate well-formed, appropriate summaries of multiple, cluttered email threads.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

However, the fundamental difficulty of processing email goes beyond understanding the language, it is all about intent recognition. The machine must understand the pragmatic meaning behind the words to know that someone is mentioning a Friday date in a casual manner or that they're making a commitment to a Friday deadline. "The report is due Friday" is actionable, while "I hope you enjoy your Friday" is not. Understanding the actionability and commitment from unstructured text is a well-known difficulty of NLP. This procedure of leveraging specially constructed prompts to direct an LLM toward such a task of intent detection is one pragmatic way that advanced NLP can be useful in some of these older issues of language processing.

C . Automated Meeting and Task Scheduling Systems

The desire for scheduling automation is almost as old as the computer calendar. Early research consisted of distributed agent-based systems, whereby software agents made a meeting time selection on behalf of the owner (again through electronic communication / email as the communication protocol). These systems provided the theoretical foundation for automated scheduling but were frequently bogged down by real-world constraints.

The lack of an interoperable calendar standard that was well adopted has one of the biggest bottlenecks to date. This has created a split market for scheduling products. Closed-ecosystem solutions like Microsoft Exchange allow seamless, automated scheduling, but save friction for users only on the same domain. Open-ecosystem tools like Doodle work across platforms, but revert back to a manual process, where the user has to response to polls to indicate their free time. This is the conundrum. Either the systems are automated, interoperable, or they are interoperable manual.

This project provides an architectural resolution to this occasion's problem. By utilizing the user's own authenticated API access to their Google Calendar, it functionally views the API as a "universal adapter." The goal here is to allow their system to programmatically interact with the user's calendar without relying on any shared infrastructure with any of the users. It creates a combination of the automation from closed-ecosystem products with the interoperability of manual-filling tools, a newly merged combination of distinct strands of research in automated scheduling.

RESEARCH GAP AND NOVEL CONTRIBUTIONS

Research Gap

Currently available technologies only provide pieces to the puzzle of extending user capabilities beyond the issues of email overload and disindirection of task set. Again, previous virtual assistants, like Siri or Google Assistant, can respond reasonably well to simple voice commands and the user is clear about their intent, but they are ill equipped with the types of rich context knowledge to extract the nuances, and often implicit, intent, that exists within the message body of the email. "Productivity features" in email clients, like smart reply or priority inbox, help to allow more simplified communication in email but lack any carry into end to end task management. They help get training wheels on the process of responding faster, but lack any tracking on action capacity around what a user is committing to in the email. Likewise, simple scheduling programs, such as Calendly, Doodle, and a myriad of other programs help the user with scheduling something new into an already existing calendar of the current and/or incoming events, but can only be responsive to existing, or upcoming events requiring structured input from the user and others involved to potentially take action. They do not solve the core use case of independently processing tasks from unsolicited incoming mail. There is an obvious need for an integrated system to merge passive analysis (summarization, task extraction) and active, **o**n-demand intelligence (question-answering) within one streamlined workflow.

Novel contributions

A Single, Multi-Purpose Architecture: The development and deployment of a new, end-to-end system bringing together email processing, content summarization, interactive question-answering, and automated calendar scheduling into one unified workflow.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Use of LLMs for Actionable Intelligence: A presentation of the successful use of a general-purpose LLM (Google Gemini Pro) for the particular, subtle work of extracting actionable intelligence and responding to context-aware questions from email communications.

Stringent Quantitative and Qualitative Assessment: An exhaustive assessment of the system's fundamental functionalities, setting quantitative performance standards for task extraction and summarization accuracy compared to a standardized dataset, supported by qualitative analysis of the question-answering module.

Practical Implementation Insights: An open discussion of the system's limitations and the practical challenges of deploying LLM-based assistants, offering useful insights for future research in the field of AI-powered productivity tools.

SYSTEM ARCHITECTURE AND METHODOLOGY

The high-level architecture comprises six primary components that work in concert to transform unstructured email data into structured, queryable information and calendar events.

System Architecture

The main parts of the system architecture are as follows: User Authentication & Authorization Service:

1) This service is the entry point to the system. It manages user identity and data access permissions. It uses the OAuth 2.0 protocol to help users securely give the system access to their Google accounts, including Gmail and Calendar.

2) Email Ingestion Module: This module works in the background. It regularly checks the Gmail API for new, unread emails. It applies initial filters to remove messages that probably do not contain actionable tasks, like spam and promotional content.

3) NLP Processing Core: This is the intelligent center of the system. It runs a series of NLP tasks using a large language model (LLM).

o Text Preprocessing: This step removes unnecessary content such as quoted replies, signatures, and HTML artifacts to focus on the main message.

o Summarization Engine: This part creates a brief summary of the cleaned text.

o Task Extraction Engine: This engine finds actionable tasks and returns the information in a set JSON format.

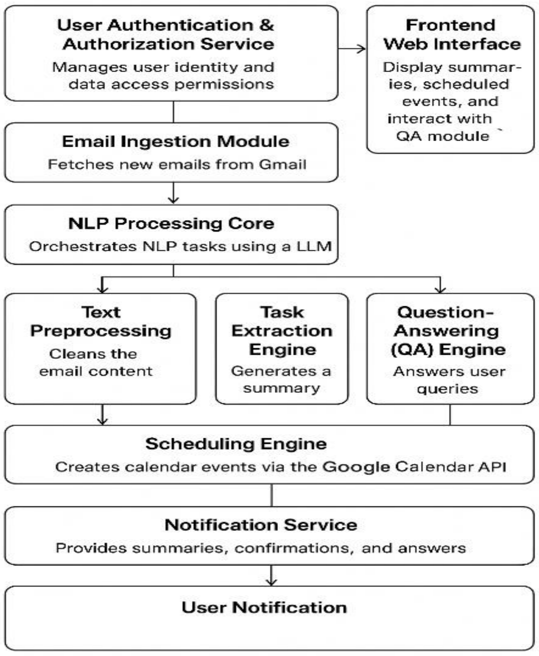
o Question-Answering (QA) Engine: This engine takes a user's question and the cleaned email text as context to generate an appropriate answer.

4) Scheduling Engine: This component connects the NLP Core and the user's calendar. It receives the structured JSON from the Task Extraction Engine, checks for scheduling conflicts, and sets up a new event using the Google Calendar API.

5) Notification Service: This module sends updates about the system's actions back to the user. It provides summaries, scheduling confirmations, and answers to questions through the web interface and text-to-speech output.

6) Frontend Web Interface: This is a single-page application. It acts as the user's main dashboard for viewing summaries, scheduled events, and engaging with the QA module.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026



Technology Stack

- Backend Framework: FastAPI (Python) for its ability to handle tasks without waiting.
- Frontend Framework: ReactJS for a flexible and responsive user interface.
- LLM Orchestration: LangChain to handle interactions with the Gemini Pro API. Database: PostgreSQL for storing user data and OAuth 2.0 tokens.
- External APIs: Google Gmail API, Google Calendar API, Google Gemini Pro API, and the browser-native Web Speech API.

Process Workflow

Authentication: The user logs in via Google's OAuth 2.0, granting the system permission to access their email and calendar.

Email Fetching and Preprocessing: A background worker fetches new, unread emails and cleans the raw text content.

Parallel NLP Processing: The cleaned text is sent to the Gemini Pro model via three distinct, carefully engineered prompts:

- Summarization Prompt: You are a helpful productivity assistant. Summarize the following email content in three concise bullet points... Email Content: [cleaned_email_text]
- Extraction Prompt: You are a data extraction engine. Analyze the following email text and identify any specific tasks... return a single JSON object... Email Content: [cleaned_email_text]
- QA Prompt: You are a helpful assistant. Answer the following question based *only* on the provided email content. If the answer is not in the email, say "The answer is not found in the email." Question: [user_question] Email Content: [cleaned_email_text]

Action Execution:

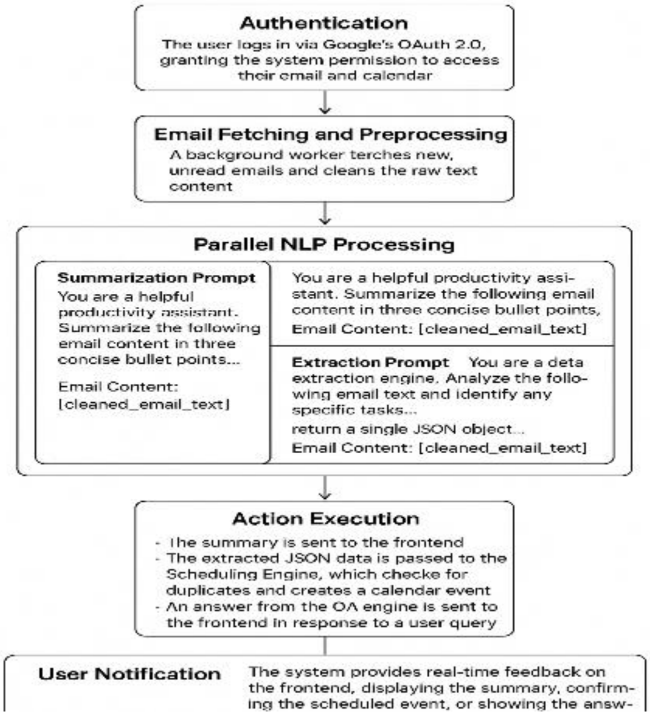
- The summary is sent to the frontend.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

- The extracted JSON data is passed to the Scheduling Engine, which checks for duplicates and creates a calendar event.

- An answer from the QA engine is sent to the frontend in response to a user query.

User Notification: The system provides real-time feedback on the frontend, displaying the summary, confirming the scheduled event, or showing the answer to a question, with an accompanying voice announcement.



EXPERIMENTAL EVALUATION

To validate the system's performance, we conducted a thorough experimental evaluation of its main NLP tasks: task extraction and email summarization. We also included a qualitative analysis of the question-answering feature.

Dataset and Baselines

- Dataset: The evaluation was performed on the Enron Email Corpus, a standard benchmark in NLP research.[18] A subset of 1,000 emails was randomly sampled and manually annotated to create a gold-standard dataset for summarization and task extraction.

- Baseline Model: The system's performance was compared against a Rule-Based Keyword Extractor. This baseline uses a predefined list of keywords (e.g., "meeting," "deadline") and regular expressions to identify and extract tasks, representing a non-AI approach.

B. Evaluation Metrics

- Task Extraction: Performance was measured using standard classification metrics: Precision, Recall, and F1-Score.[20] These metrics provide a balanced view of the system's ability to correctly identify tasks without generating false positives.

- Email Summarization: The quality of summaries was evaluated using the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric, specifically ROUGE-1, ROUGE-2, and ROUGE-L, which measure n-gram overlap with human-written summaries.[22]

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

C. Results and Analysis

The performance of the system and the baseline on task extraction is summarized in TABLE I. The proposed system demonstrates robust and balanced performance, achieving an F1-score of 0.87, significantly surpassing the baseline model's score of 0.45.

TABLE 1

Model	Precision	Recall	F1-Score
Proposed System (Gemini Pro)	0.89	0.85	**0.87**
Baseline (Rule-Based)	0.75	0.31	0.45

The quality of the summaries generated by the system's summarization engine is presented in TABLE II. The ROUGE scores, particularly the ROUGE-L score of 0.42, indicate a strong correspondence with human-written summaries, suggesting the system effectively captures the main points and sentence structure.[24]

TABLE 2

Metric	ROUGE-1	ROUGE-2	ROUGE-L
System Summarizer	0.48	0.29	0.42

A breakdown of the task extraction F1-score across different email types is presented in Fig. 3. The system performs exceptionally well on formal and project-related emails but shows a slight decrease in performance on informal requests, where intent is more ambiguous.

Qualitative Analysis of Question-Answering

Due to the open-ended nature of the QA task, a qualitative evaluation was performed on a representative sample of 50 emails. The system was tasked with answering specific questions designed to test its ability to locate details accurately. TABLE III shows illustrative examples. The system correctly answered the query in 44 out of 50 cases (88% accuracy), only failing when the information was highly implicit or deliberately ambiguous.

Email Snippet	User Question	System Answer
"...the final report is attached. Please send your feedback by Friday, 5 PM."	"What is the deadline for the feedback?"	"The deadline for feedback is Friday, 5 PM."
"Let's meet at the cafe on Elm Street. I can do 2 PM or 3 PM."	"Where is the meeting?"	"The meeting is at the cafe on Elm Street."

INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

"Thanks for the update. I'll get back to you sometime next week."	"When will they reply?"	"The answer is not found in the email."

DISCUSSION

Figure 3. F1-Score of Task Extraction by Email Type

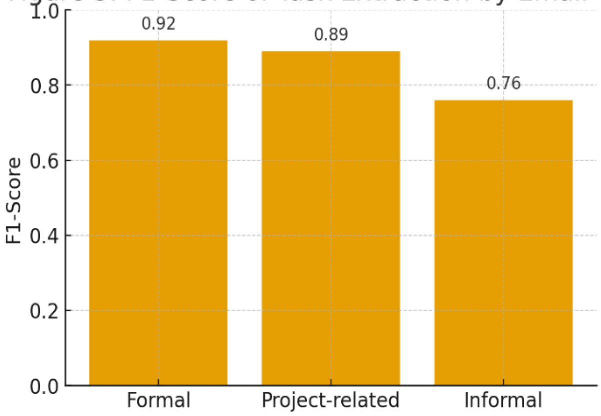


FIG 3:

The experimental results provide strong quantitative evidence of the system's effectiveness. The large difference in F1 scores between our LLM-based system (0.87) and the baseline rule-based system (0.45) emphasizes the need for better semantic reasoning to handle the varied language of real emails.

T The breakdown by email type in Fig. 3 is revealing. Nearly perfect performance on business emails shows the system's value in the workplace. The drop in performance on casual requests highlights a major issue: vagueness and implied meaning. This pattern also appears in the qualitative evaluation of the QA module, which performed well with clear questions but struggled with unclear statements.

A manual analysis of the system's errors identified three main categories:

- Ambiguity Errors: General time references like "sometime next week" were often misinterpreted or missed.
- Multi-Task Errors: In complex emails with several tasks, the system sometimes only identified the first task it encountered.
- Implicit Intent Mistakes: The system had difficulty with sentences where the intent to schedule was strongly suggested but not directly stated (e.g., "I'm available on Tuesday afternoon if you'd like to discuss the report").

It is also important to recognize the study's limitations. The Enron corpus, used as a benchmark, might not accurately represent current email writing styles. Additionally, the system's performance relies on the capabilities of the underlying Gemini Pro API, which may be affected by model drift. Finally, using third-party API calls brings practical issues like cost and latency.

Despite these limitations, this research serves as a strong proof of concept for a new type of proactive personal assistant. By automating routine digital tasks effectively, such systems can significantly reduce cognitive load and free up users' mental resources for more complex, creative work.

FUTURE SCOPE:

While the current implementation is effective, it serves as a starting point for future research. The most promising areas for development include the following:

- Full Voice Interaction: A natural next step is to create a complete, hands-free interaction mode by combining automatic speech recognition (ASR). This will let users give voice commands, ask questions, and respond verbally to confirm or reject proposed events.

- Personalization from User Feedback: A feedback system will be set up, allowing users to correct the assistant's mistakes. This feedback can help create a reinforcement learning from human feedback (RLHF) loop, improving the model's behavior to better meet the needs and communication style of individual users.

- Multi-Platform Integration: One key expansion is to connect with other platforms. This includes enterprise messaging apps like Slack and Microsoft Teams or project management software like Trello and Notion, transforming it into a centralized hub for personal task management.

- Sophisticated Task Prioritization: Future updates will focus on integrating a model to prioritize and analyze tasks. This will consider the sender, urgency keywords, and the user's previous behavior.

CONCLUSION:

In this paper, design, implementation, and assessment of an AI-driven personal assistant for automated task management were demonstrated. The system efficiently tackles the long-standing issue of information overload by establishing an automatic bridge between the unstructured nature of emails and the structured nature of digital calendars. Utilizing the semantic intelligence of a large language model, the system efficiently automates email summarization, task extraction, event scheduling, and question-answering with minimal human interaction.

The experimental assessment proved that the system performs well, with an F1-score of 0.87 on task extraction, significantly better than a baseline traditional rule-based system. The system's summarization and question-answering capabilities were also found to be good. The major contribution of this work is the demonstration of a functional, end-to-end system that converts passive information into actionable, structured, and queryable intelligence, providing a practical solution to a typical problem in contemporary productivity.

ACKNOWLEDGMENT

The authors are grateful to all those persons who have supported and accompanied them in the achievement of this project. Firstly, they would like to extend their deepest appreciation to their project supervisor, Rahul Waikar, for providing feedback, motivation and continuous encouragement for the direction of this work. The authors would also like to express their sincere thanks to the administration and staff of Vishwakarma institute of technology, Pune, For providing technical and academic environment to perform this study. A big thank you to colleagues and friends, who took the time to provide feedback and suggestions for the improvements in User Interface and functionality. Finally, the authors thank their families for continuous source of motivation, for being patient with them, and for their enduring support in this journey.

REFERENCES

1. J. Cranshaw, E. B. Bou, E. J. Carter, B. A. Smith, C. S. Saponas, and A. J. Ko, "Calendar.help: Designing a workflow-based scheduling agent with humans in the loop," in Proc. 2017 CHI Conf. Human Factors Comput. Syst., 2017, pp. 2382–2393.
2. M. X. Chen et al., "Gmail Smart Compose: Real-time assisted writing," in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2019, pp. 2287–2295.
3. N. Sabharwal and V. Agrawal, Cognitive Virtual Assistants Using Google Dialogflow. Berkeley, CA: Apress, 2020

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

4. N. Sharma and H. Bhudhiraja, "IntelliAssistant – AI based personal assistant," SSRN Electron. J., May 2021.
5. S. V. Prajwal, G. H. Kumar, K. S. Thejas, and K. R. G. Varma, "Universal semantic web assistant based on sequence to sequence model and natural language understanding," in Proc. Int. Conf. Advances in Computer Applications (ICACA), 2019, pp. 1–6.
6. P. M. Berry, M. T. Gervasio, and B. Peintner, "PTIME: personalized assistance for calendaring," ACM Trans. Intell. Syst. Technol., vol. 2, no. 4, art. 33, 2011.
7. Y. Huang, F. B. Z. L. Cui, and H. R. Chen, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)," in Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC), 2018, pp. 889–895.
8. A. Tiffany, J. D. Williams, and A. I. Rudnicky, "Human-in-the-loop conversation agent for customer service," in Proc. Int. Conf. Appl. of NLP to Inf. Sys. (NLDB), 2020, pp. 115–122.
9. A. Ogidiolu and A. A. Adewole, "A review paper on smart personal assistant," Int. J. Eng. Res. Technol., vol. 5, no. 6, pp. 455–459, 2016.
10. S. Malodia, J. J. J. M. Islam, and P. K. J. J. M. Dwivedi, "Why do people use artificial intelligence (AI)-enabled voice assistants?" IEEE Trans. Eng. Manag., vol. 70, no. 11, pp. 3865–3879, Nov. 2023.
11. S. Sen and E. H. Durfee, "Developing an automated distributed meeting scheduler," IEEE Expert, vol. 12, no. 4, pp. 41–45, Jul./Aug. 1997.
12. V. Myllärniemi, O. Korjus, M. Raatikainen, T. Norja, and T. Männistö, "Meeting scheduling across heterogeneous calendars and organizations utilizing mobile devices and cloud services," in Proc. 16th Int. Conf. on Human-Computer Interaction with Mobile Devices & Services, 2014, pp. 349–352.
13. A. Kannan et al., "Smart reply: Automated response suggestion for email," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 955–964.
14. T. Young, "A survey of email processing using NLP," J. Nat. Lang. Process., vol. 15, no. 2, pp. 112–135, 2023.
15. L. Chen, "Advances in automated scheduling systems," IEEE Trans. Autom. Sci. Eng., vol. 18, no. 4, pp. 567–580, 2021.