A Unified Browser-Embedded Ai Assistant for Summarization, Task Mining, And Calendar Automation

Saksham Ovalekar[1], Omkrish Chauhan[2], Meet Pabari[3], Pratik Pachorkar[4]
[1,2,3,4]Computer Engineering Vishwakarma institute of technology, Pune, Maharashtra

***Abstract*—Emails and webpages are often processed, tasks are identified and the calendars are updated whilst browsing, which often is supported by different applications. Such division involves repetitive manual work to condense information, to action, and to plan undertakings. This paper proposes unified browser-based AI assistant to summarize, task mining, and calendar automation, which will be directly embedded into the Chrome platform with a floating interface. The system allows real-time support without the users having to quit the current webpage, the architecture integrates lightweight transformer-based models of low latency summarization with large language models of semantic understanding, structured task extraction and event inference. It analyses raw webpage and email content to generate concise summary at a glance, categorize messages, saving to-do items with an appointed date and to automatically generate calendar events. The design is now multilingual with noisy text that is often used in real-life communication. The system is able to bring all these capabilities into one workflow that can be based on a browser to minimize the effort of manual coordination and show how intelligent assistants can transition outside of information support, towards practical task automation in the daily digital environments.**

***Index Terms*—Browser-Embedded AI, Text Summarization, Task Mining, Email Analysis, Calendar Automation, Intelligent Personal Agents, Human–AI Interaction, Natural Language Processing, Productivity Systems**

I. INTRODUCTION

Users often work with emails, read web pages and list tasks and schedules in the course of the same browsing session, however, package the activities tend to be distributed across different applications which do not share the context. Due to this, users are left to summarize content manually, interpolate requests, and derive actionable items, as well as, get commitments into their calendars. The process brings in friction and creates a high chance of lack of information or action delays. The recent progress in natural language processing and large language models now allow systems to comprehend an unstructured text, summarise it, and select an actionable content, thus the possibilities to provide a more integrated and automated assistance in the digital contexts.

The present paper presents a single browser-based AI assistant used to summarize, mine tasks, and automate the calendar which works via a floating interface within the Chrome browser. It combines lightweight transformer based low-latency summarization with large language models based on semantic understanding, structured task extraction, and event inference. It allows the real time web page and email summary, email category recognition, deadlines-based tasks extraction, and automatic calendar event creation. The system has the capability to integrate these functions in one in-browser workflow, thereby offering a highly viable and scalable solution to manual coordination costs and productivity enhancement in the daily digital routine.

II. LITERATURE REVIEW

The intelligent meeting scheduling technology examined by Brzozowski et al. [1] is the group Time system, which instead of using free/busy calendar information to solve the scheduling problem, the scheduling problem can be considered as a preference-learning problem. Their system enables people to define subtle availability and machine learning predicts an appropriate time to have a meeting. It also has interface designs that maintain the privacy of the users and minimize friction in coordination. The study shows that learning user preferences can make a considerable decrease in the cognitive effort to make the required choices and enhance decision support

when managing the calendar. Brachman et al. [2] examined the exploration of the knowledge worker adoption and use of large language models in professional contexts by means of a large-scale user study. They discovered that LLMs are mostly applied to writing, summarization, brainstorming and automatization of routine work. They find that AI systems that are connected to actual data sources and everyday tools are appreciated by workers. The research notes the need to integrate AI into the current operations instead of providing it as a separate chatbot. The OPINE framework of open intent discovery was introduced by Vedula et al. [3], and it can identify action-object pairs of text by extracting them, without using a predefined intent schema. They employ neural sequence tagging and attention as part of the model in identifying actionable intents across domains. The work shows that interesting tasks may be automatically mined out of open-domain text. This justifies the possibility of retrieving TODOs and commitments via emails and webpages. In Calendar. help presented by Cranshaw et al. [4] a hybrid human-AI scheduling assistant, meetings are scheduled by the corresponding natural email interaction. The system breaks down scheduling into quite organized micro tasks and microtasks, that are managed by automation and human operant. These assistants have demonstrated the ability to deal with thousands of meetings consistently over a period of one year. The researcher offers findings that the integration of smarts into email processes can significantly decrease user work load. Brachman et al. [5] examined how generative AI tools can be used to increase the productivity of knowledge workers in controlled experiments. They discovered that access to LLCM-based assistance increased speed and quality of task completion, particularly to tasks that had high level of information. According to their findings, AI can serve as a cognitive support aid tool to professionals. The paper supports the possibility of the more efficient ways of decision-making and productivity in daily work and the use of integrated AI systems.

Lewis et al. [6] proposed BART, a sequence-to-sequence denoising pretraining model, which is a combination of bidirectional encoding with autoregressive decoding. Their model is corrupted with text and trained to rebuild it to provide well-developed language comprehension and generation abilities. BART is able to produce state of the art performance in the summarization and dialogue tasks. The analysis reveals that abstractive summaries that are produced by pretrained transformer models have quality, thus they can be applied in actual text summarization systems. Huot et al. [7] described the idea of uPLAN which is a cross-lingual summarization method which employs entity-based content planning to direct summary generation. The mode works with salient entities among languages to minimize hallucinations and enhance faithfulness. The system displays excellent transfer and multilingual capabilities. This article brings out measures towards the production of credible overviews within the multilingual environment. In the paper by Mowar et al. [8], the authors reviewed how AI coding assistants can be used in accessible web development using the CodeA11y system. As they discovered, developers neglect accessibility until it is suggested and AI tools can positively influence improved behavior. With their extension, they offer accessibility-based recommendations and warnings. The study demonstrates that inbuilt AI assistants may influence the behavior of users and enhance the results once they are included in the working processes. Chester et al. [9] compared the idea of summarization related to low-resource languages in terms of zero-shot LLMs, multilingual fine-tuning, and data augmentation. Their results indicate that multilingual fine-tuning usually works better in comparison with zero-shot methods. They also emphasize the issue of evaluation in low-resource settings. The paper puts significant emphasis on adjustment of model to multilingual summarization. Intrator et al. [10] tested the need of pre-translation to English during the use of multilingual LLM. They conducted experiments in 108 languages where they discovered direct inference to be better in the source language. Some of the evaluation techniques they introduced include language-ratio to prevent misleading averages. Their finding indicates that the most current version of LLM is capable of working on multilingual input without the need of translation pipelines.

In their research on information extraction, Freitag et al. [11] focused on the informal and noisy domains, including emails and online posts, in which grammatical format is not always sure. The book compares several machine learning methods such as, rote learning, Naive Bayes, grammatical induction and

learning relational rules. It focuses on non-linguistic elements of communicating such as formatting, typography, and layout. The paper demonstrates that multistrategy learning enhances extraction performance of real text. The work is a prerequisite towards deriving structured information of messy email and web data. The Smart Reply system by Henderson et al. [12] was based on n-gram feed-forward neural networks that use feedforward neural networks to suggest email responses based on feedforward neural networks. They use a method that is efficient in pairing messages and candidate responses based on vector similarity. The system surpasses the sequence-to-sequence models with similar quality and lower latency. It has been effectively implemented to a large email system. This paper identifies how AI could be used to automate email-related activities in real-time. Itsnaini et al. [13] examined abstractive summarization on the text in Indonesian language through T5 transformer model. In their work the authors consider summarization as a text-to-text activity, and assess performance in terms of ROUGE measures. Findings indicate a high performance in summarization but also illustrate a weakness in the performance of abstraction. The work demonstrates the activity of pretrained transformer models to be useful in practice-oriented summarization tasks. The Azaria et al. [14] paper described an instructable intelligent personal agent, which learns new commands by using natural language instructions. A semantic parser and lexicon induction are used to map user instructions to executable actions which are part of their system. A pilot study indicates active teaching of the agent by people and a saving in the time spent. The article proves that natural-language-controlled personal agents are feasible. The model by Bhattacharya et al. [15] is a comparative analysis of abstractive summarization models of clinical radiology reports. They tested T5, the BART, the PEGASUS, and the LLMs on a variety of metrics. They presented their strengths, problematic areas in the domain, and the risk of hallucination. They emphasize the significance of the factual consistency and evaluation rigor in the high stake's summarization work.

III. METHODOLOGY

The methodology is based on fault-tolerant and modular architecture that is aimed to provide reliable text-understanding and intelligent automation in a browser-based setup. The system combines two main pipelines, a text-extraction and summarization pipeline that can be used to process webpage content in both online and offline environments, and an email-automation pipeline that can be used to extract actionable tasks, analyze emails, and plan events. Continuous operation, efficient information processing and automated working environment under changing connectivity conditions are guaranteed by adaptive model selection, secure authentication, and NLP-based decision mechanisms.

A) Text-Extraction & Summarization Pipeline:

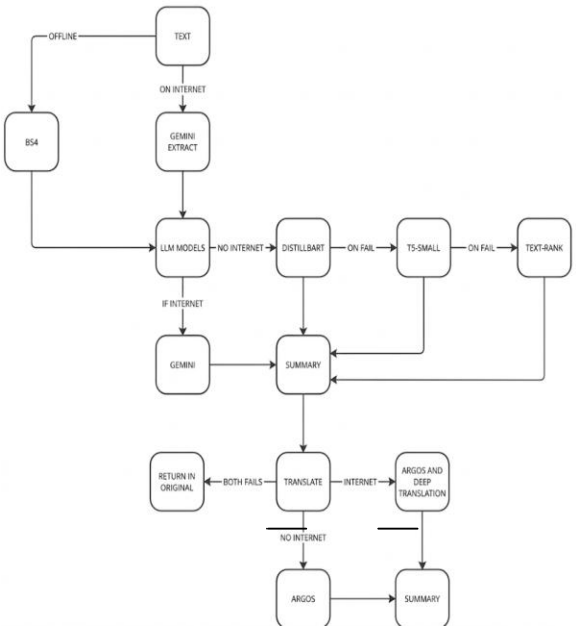


Figure 1: Fault Tolerant Pipeline

The proposed system shown in Figure 1 deploys a fault-tolerant text processing pipeline that is to be utilized reliably in both connected and low-connectivity environments. It is modular, which allows graceful degradation, but still guarantees the generation of a usable summary. The entire processing entails four phases; (1) acquisition and extraction of the text, (2) cleaning and normalization, (3) adaptive summarization, and (4) translation. Every step will involve backup processes to sustain operations.

1) Text Acquisition and Extraction.
The pipeline takes in text of both offline and online sources. On of the offline sources, locally stored HTML files are exploited and their contents are parsed with Beautiful Soup (BS4), retrieving only visible textual information and discarding scripts, styles and markup clutter. In the case of online sources, a web extracting service (e.g., a Gemini-based extractor) takes the contents of web pages and transforms them into clean text. With any network problems that cannot result in successful online extraction, the system will automatically switch to local HTML retrieval and BS4 parsing where it can occur. This two-way structure will guarantee that text can be learnt irrespective of the situations of connectivity.

2) Cleaning and Normalization.
Preprocessing is done to extracted text in order to enhance the quality of summarization downstream. This involves elimination of boilerplate materials, duplicate matching and normalization of white space and basic noise removal. The cleaning module is a rule-based system and works entirely offline and ensures uniformity of preprocessing across settings. The product of this step is standardized plain text that can be used as the input of language models.

3) Decision Layer Adaptive Summarization.
An internet-monitoring decision layer detects the availability or not and directs the text that has been cleansed to a relevant summarization model. In the case of internet connectivity, a context-sensitive abstractive summary is produced with the help of a high-capacity cloud LLM (e.g., Gemini). Offline sequential fallback strategy is used, first Distil BART to achieve efficient transformer-based summarization, then T5-Small in case of resource constraints or failures, and lastly Text Rank to extract summaries in a lightweight manner. It follows a hierarchical fallback strategy so that even low-resource devices can be capable of doing summarization. Any generated summaries are sent to a single summary module, which is used to standardize length, formatting and structure.

4) With Graceful Degradation translation.
The single summary is optionally sent to a stage of translation. In the case that there exists internet a local neural system (Argos) is integrated with a deep online translation service to enhance fluency and coverage. As an offline mode, Argos has its own translation service to offer local services in different languages. In case of failure of translation or unavailability of language resources, the system displays the initial text of the summary. This guard ensures that the output is not disrupted.

Comprehensively, the approach lays stress on modularity, redundancy and graceful degradation. The pipeline ensures a stable operation in a variety of environments through the dynamically chosen models depending on connectivity and the available of resources. This design facilitates a stable user experience by design, that is, text extraction, text summarization, and text translation continue to offer functionality to the system even when it is not fully connected to the internet, or when not connected at all.

B) Email Automation & Scheduling Pipeline:

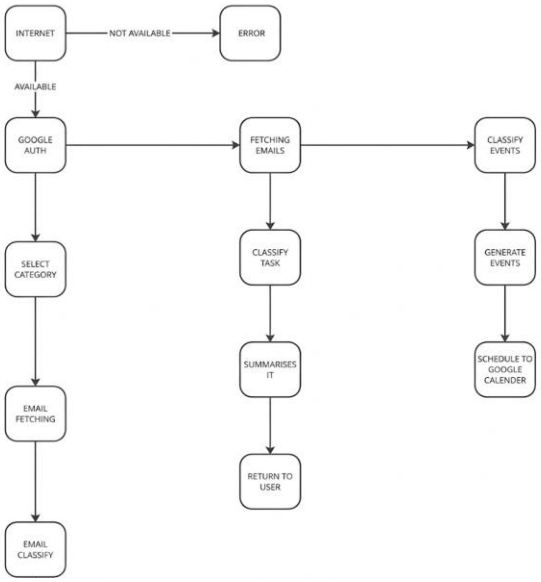


Figure 2: Email Automation with Fault Tolerant Pipeline

The proposed email-automation pipeline shown in Figure 2 will transform raw emails into the actionable output of email summaries, recovered and scheduled calendar events. The system is based on a modular and fault-conscious architecture which incorporates connectivity checks, secure authentication, email retrieval, natural language processing and automatic scheduling. The workflow is effective to provide the

reliability of the functioning and ensure the data security and privacy of the users.

1) Connectivity Verification.
The pipeline is initiated by a check of internet connection since email communication, authentication, and calendar organization relies on internet services. In case of unavailability of connection, a clear notification of error will be raised by that system to the user and the system will stop the further processing. This will not lead to partial execution and authentication or API failures. In the case of connectivity, the pipeline goes to obtain access phases.

2) Authentication and Authorization.
The system will be based on OAuth 2.0 authentication to gain access to the user Gmail, and Google Calendar in a secure manner. The permission is given with the use of a consent flow and access tokens are stored in the cache to be as efficient as possible in a session and will be refreshable on demand. There is never storage of sensitive credentials in plain text and all communication takes place in secure channels. This architecture makes this possible to meet the standard security practices but is able to accommodate repeated access in an easy way.

3) Retrieval of emails and categorizing.
After authentication the user will be able to choose categories or filters (e.g., primary, promotions, updates, or custom labels). The Gmail API accesses the email inbox and decodes the message content to readable text in the inbox with the help of the pipeline. Preprocessing eliminates signature, disclaimers, and formatting noise in order to create clean input to be analyzed. This step normalizes email contents in a uniform manner of downstream processing.

4) Classification and Routing based on NLP.
Lightweight classifiers and language models are used to analyze fetched emails in order to classify the type and intent. The system classifies emails and determines whether they are informational, actionable and event-related. According to this classification, the emails are directed to one of two intelligent workflows, the task or event workflow. This level of decision allows its specific processing instead of performing a single approach on each and every email.

5) Task Workflow.
The actionable emails are managed with the objective of identifying tasks, deadlines, and important information. The system summarizes the task setting and produces brief results in form of TO do items or priority notes. These summaries will be made available to the user in structured and readable format so that one can understand and act fast without having to reread long emails.

6) Calendar Automation and Event Workflow.
Dates, times, durations, and participants are identified after processing emails that contain details about the event. The system transforms information that has been extracted into organized calendar data and automatically submits it in Google Calendar via the API. Automation is kept transparent and controlled as the users are also informed of upcoming events.

In general, the methodology incorporates connectivity awareness, secure authentication, intelligent classification, summarization, and scheduling in one pipeline. The modular architecture enables uninterrupted enhancement of every part even as it can go ahead to operate continuously end to end. The system makes emails easier to complete by converting documents to tasks, summaries, and scheduled activities, making emails simpler to manage and efficiently complete the workflow.

IV. RESULT AND FINDING

This section will give both the quantitative and qualitative results of the system implemented. The quantitative study is coupled with the comparative outcomes of the existed summarization models whereas the quality evaluation is reflected in the system outputs in real-world and accessible user functionalities.

a) Performance- Summarization Model:
In order to place the performance of summarization in perspective, a comparative analysis of the transformer-based summarization models was cited in the article by Bhattacharya et al. [15], where the study comparing the performance of T5, BART, PEGASUS, and large language models are evaluated on multiple evaluation metrics. They provide their analysis on the strengths of transformer architectures to generate coherent and

informative summaries and also identify the known shortcomings of using them such as the risk of hallucination and sensitivity to the domain.

The comparative Table 1 presented in the given work is tailored to the one by Bhattacharya et al. [15] and it assisted in the justification of the choice of transformer-based summarization methods in the system pipeline. Rest of these findings support that design decision where lightweight local models will be combined with powerful cloud-based models to strike a balance between summary quality, latency and reliability.

Table 1: - comparative analysis of different fine-tuned summarization models.

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERT
BART-base	0.37	0.22	0.33	0.34	0.88
T5-base	0.37	0.21	0.32	0.39	0.88
PEGASUS-X-base	0.30	0.17	0.26	0.33	0.87
ChatGPT-4	0.29	0.13	0.23	0.35	0.87
LLaMA-3-8B (without fine-tuning)	0.14	0.05	0.12	0.13	0.82
LlaMA-3-8B (fine-tuned)	0.27	0.11	0.25	0.23	0.87
PGNwithCOV	0.13	0.04	0.11	0.11	0.81

b) Webpage Summarization:

The webpage summarization system takes raw HTML and extracts meaningful text content and filters structural noise. The computer then creates brief contents in the language of choice by the user. The Probing of different webpages demonstrated that the summary retained the important facts and significantly reduced reading time spent on reading.

This feature allows information to consume in a fast manner and it proves to have realistic use when one has a heavy information load to browse in multiple languages as shown in Figure 3.

Figure 3: Shows Successful Summarization in Multiple Languages Using Fine-Tuned Models

c) Email Summarization:

The assistant is linked to G-mail by authenticated OAuth. Upon logging in, the system reads the new emails and creates brief, professional summaries. Emails have been sorted as Professional and Updates where the users can give priority to the important communication.

Monitored results in Figure 4 shows that the summaries were able to uncover the essence of the message of emails to facilitate effective review of the inbox.

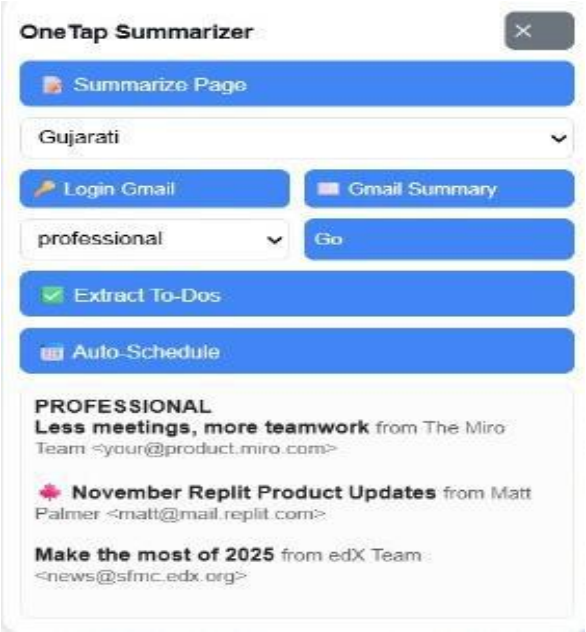


Figure 4: Shows Successful Classification of Mails Using NLP.

d) Automatic To-Do Extraction:

The system goes beyond summarization by being able to derive actionable items on the content in mail. Extract To-Dos feature will identify the commitment like meetings, deadlines and follow-ups as shown is Figure 5. Tasks that are extracted are organized into easily understandable entries that can be scheduled.

This ability shows that it can be actionable intelligence as opposed to passive summarization.

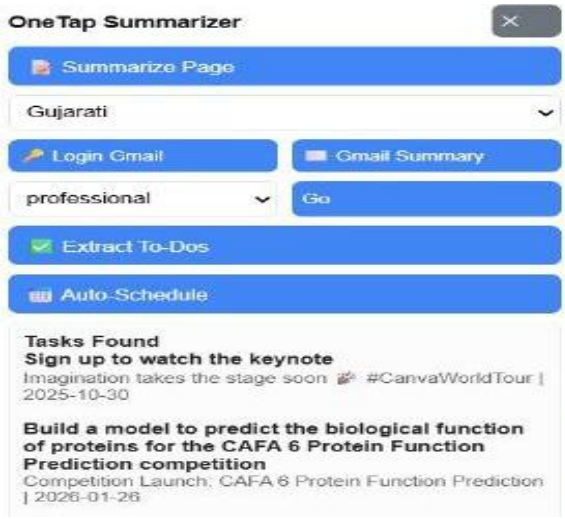
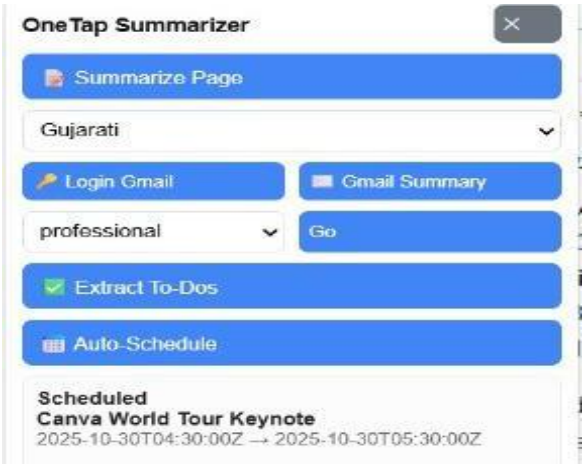


Figure 5: Shows Successful Extraction of Tasks

e) Automatic Calendar scheduling.

The scheduling feature transforms the tasks extracted into the calendar events through Google Calendar integration. The time slots are taken and the events are allocated without contradictions as shown in Figure 6. Successful scheduling is ensured by the calendar interface.

This automation minimizes the level of error in manual effort and schedule minimizes the chances of an important commitments being missed.



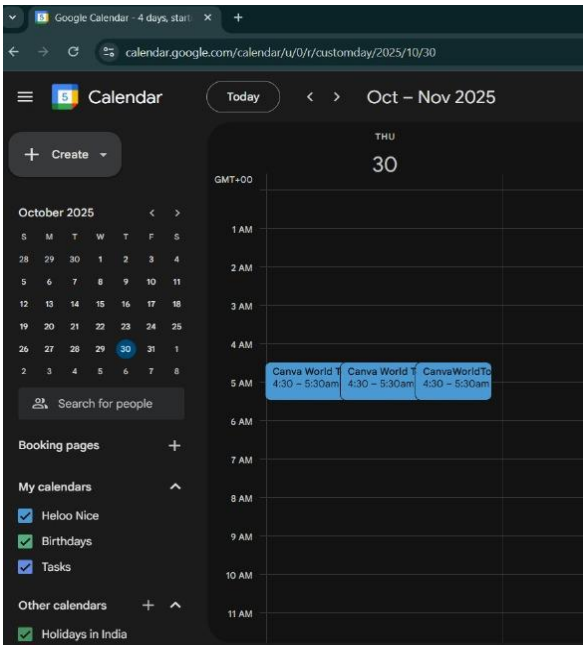


Figure 6: Shows Successful Scheduling of Events in Calendar

Overall Findings:

The findings suggest the meaningfulness of incorporating summarization, the task extraction, and the scheduling feature into an assistant that lives on a browser platform. Unstructured information is converted by the system into actionable outputs, therefore, minimizing coordination work and assists in managing workflow operations effectively.

V. CONCLUSION

This paper introduced an integrated AI assistant in the form of a webpage summarizer, email summarizer, task extractor, and automated calendar scheduler within the same workflow. The system operates within the browser itself, which reduces the context switching and facilitates real-time since the users already read and make use of the information as they do. The architecture has integrated lightweight local models with the cloud-based language models, which enables the system to trade-off between latency, reliability, and quality of the summary, as well as, multilingual and noisy-text scenarios.

The findings indicate that manual coordination effort and productivity can be reduced by changing unstructured online materials into succinct summaries, action plans, and timetables. The combination of

protected access to email, intelligent classification, and automatic scheduling demonstrates that the transition on the next level beyond information display to the automating of tasks is possible. On the whole, this has the potential to drive AI assistants more practical and effective and workflow-friendly through the integration of several AI features into a familiar browsing environment.

Although the existing system is only being used in terms of summarization and email-based automation, the system can be expanded to other productivity capabilities and domains. The study is relevant to the current research in the human-AI interaction and intelligent personal agents in that closely integrated action-focused assistants can be meaningful in daily digital tasks.

REFERENCES

[1] Brzozowski, Mike, Kendra Carattini, Scott R. Klemmer, Patrick Mihelich, Jiang Hu, and Andrew Y. Ng. "Group Time: preference-based group scheduling." In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 1047-1056. 2006.

[2] Brachman, Michelle, Amina El-Ashry, Casey Dugan, and Werner Geyer. "How knowledge workers use and want to use llms in an enterprise context." In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1-8. 2024.

[3] Vedula, Nikhita, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. "Open intent extraction from natural language interactions." In *Proceedings of the web conference 2020*, pp. 2009-2020. 2020.

[4] Cranshaw, Justin, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. "Calendar. help: Designing a workflow-based scheduling agent with humans in the loop." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 2382-2393. 2017.

[5] Brachman, Michelle, Amina El-Ashry, Casey Dugan, and Werner Geyer. "Current and future use of large language models for knowledge work." *Proceedings of the ACM on Human-Computer Interaction* 9, no. 7 (2025): 1-24.

[6] Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 7871-7880. 2020.

[7] Huot, Fantine, Joshua Maynez, Chris Alberti, Reinald Kim Amplayo, Priyanka Agrawal, Constanza Fierro, Shashi Narayan, and Mirella Lapata. "μPLAN: Summarizing using a Content Plan as Cross-Lingual Bridge." In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2146-2163. 2024.

[8] Mowar, Peya, Yi-Hao Peng, Jason Wu, Aaron Steinfeld, and Jeffrey P. Bigham. "CodeA11y: Making AI Coding Assistants Useful for Accessible Web Development." In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1-15. 2025.

[9] Palen-Michel, Chester, and Constantine Lignos. "Comparing Approaches to Automatic Summarization in Less-Resourced Languages." *arXiv preprint arXiv:2512.24410* (2025).

[10] Intrator, Yotam, Matan Halfon, Roman Goldenberg, Reut Tsarfaty, Matan Eyal, Ehud Rivlin, Yossi Matias, and Natalia Aizenberg. "Breaking the language barrier: Can direct inference outperform pre-translation in multilingual LLM applications?." *arXiv preprint arXiv:2403.04792* (2024).

[11] Freitag, Dayne. "Machine learning for information extraction in informal domains." *Machine learning* 39, no. 2 (2000): 169-202.

[12] Henderson, Matthew, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. "Efficient natural language response suggestion for smart reply." *arXiv preprint arXiv:1705.00652* (2017).

[13] Itsnaini, Qurrota A'yuna, Mardhiya Hayaty, Andriyan Dwi Putra, and Nidal AM Jabari. "Abstractive Text Summarization using Pre-Trained Language Model 'Text-to-Text Transfer

Transformer (T5),’." *ILKOM Jurnal Ilmiah* 15, no. 1 (2023): 124-131.

[14] Azaria, Amos, Jayant Krishnamurthy, and Tom Mitchell. "Instructable intelligent personal agent." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1. 2016.

[15] Bhattacharya, Anindita, Tohida Rehman, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. "Comparative Analysis of Abstractive Summarization Models for Clinical Radiology Reports." *arXiv preprint arXiv:2506.16247* (2025).