

CS510 Context Aware Email Response Tool

1. [Functions and Users]

The software tool we are planning to implement is a browser-based extension that integrates with existing email platforms such as Gmail and Outlook. Rather than being a standalone app or new communication system, this extension enhances the functionality of current email interfaces by offering intelligent, AI-powered assistance for drafting personalized email responses.

The major functions of the tool include-

- Contextual email analysis to understand the tone, content, and intent of the original message.
- AI-generated draft replies that go beyond short, generic responses and instead offer thoughtful, well-structured replies.
- Tone customization, allowing users to choose or set their preferred tone (e.g., formal, friendly, assertive, etc.).

The primary users of this tool are-

- Professionals across various industries who engage in high volumes of email correspondence.
- Students and academic researchers communicating with professors, advisors, or collaborators.
- Non-native English speakers who may need help composing clear and professional messages.

2. [Significance]:

- a. Why do we need the tool/system that you propose to develop?
 - i. In today's world people spend a huge amount of time writing and responding to emails, especially in a professional setting. Although AI-generated short replies exist (ie gmail's smart reply), they are often

very simplistic and generic. These responses aren't suitable for a long complex response. Thus, there needs to be a tool to generate thoughtful and more complex responses to reflect the user's unique communication style and tone, helping the user save time and improve communication.

- b. Does your tool/system address any existing "pain point"?
 - i. Yes it addresses a pain point because writing such personalized emails can be time consuming especially when responding to long email threads or communication with clients, coworkers, or professors. Oftentimes, people struggle to find the right words to maintain tone consistency or quickly draft professional responses under time pressure. This can result in email fatigue and decreased productivity. Thus by developing this plugin, the user will be able to more efficiently draft a response.
- c. How would our world be different because of your new tool/system?
 - i. Using this new tool, users are able to type up smarter and personalized email responses much more quickly, allowing them to have much more time to focus on other meaningful tasks. This will also alleviate stress for people who find writing emails stressful. These include non-native speakers, students, or professionals who regularly deal with high volumes of communication.
- d. Does it address a societal need?
 - i. Yes this addresses a societal need for effective communication tools, especially considering that email is core communication method for work, education, healthcare, and more.
- e. Why is it important to address this need?
 - i. Remote work and global collaboration continues to increase. Thus, the importance of email communication and the volume of it will continuously grow. If this need isn't addressed, people will continue to waste time on writing such long personalized email responses. This in turn will lead to inefficiency and communication burnout. Furthermore, people who don't have strong writing skills will continue to be at a disadvantage. Therefore it is important to address this need.

3. [Approach]

- a. How do you plan to build it?

Technology Stack & Architecture

- **Browser Extension Framework:**
Build the tool as a browser extension using HTML, CSS, and JavaScript (or TypeScript). Leverage existing browser APIs (Chrome, Firefox, Edge) to integrate seamlessly into Gmail and Outlook web interfaces.
- **Backend Integration:**
Integrate with an AI service (such as the OpenAI API or another NLP model) to perform contextual analysis and generate thoughtful email responses. This backend can be built with Node.js or a similar framework and will communicate with the extension via secure RESTful endpoints.
- **Modular Design:**
Design the system with separate modules for:
 - **Email Context Extraction:** Parsing the email content, tone, and intent.
 - **AI Response Generation:** Communicating with the AI service to generate customized replies.
 - **User Interface:** Allowing users to review and customize draft replies, select tone preferences, and send responses.
- **Detailed functionality and plan for building**
 - **Phase 1 – Core Functionality:**
Implement basic email content extraction and AI-based draft generation.
 - **Phase 2 – Tone Customization & UI Enhancements:**
Enable tone settings and a user-friendly interface for reviewing AI drafts.

b. Existing Resources & Leveraging APIs

- **Email Platform APIs:**
Utilize Gmail and Outlook APIs or DOM manipulation techniques to extract email content.
- **AI & NLP Libraries:**
Leverage existing AI frameworks and libraries (e.g., TensorFlow, PyTorch) or utilize pre-trained models available via API or cloud services to handle language understanding and response generation.

c. Risk Mitigation & Barriers

- **Data Privacy Concerns:**
 - **Risk:** Sensitive email content may be exposed when processed by external APIs.
 - **Mitigation:** Process data only after explicit user consent, and use encryption for data transmission. Consider on-device processing options if feasible.
- **Performance & Latency:**
 - **Risk:** AI-generated responses may introduce delays in email drafting.
 - **Mitigation:** Optimize backend processing and implement caching for repeated interactions. Provide visual feedback to users during response generation.

4. [Evaluation]

- a. We plan to do a small user study where we will assign other people to use our plugin for drafting up an email. We plan to create a dedicated questionnaire for our users to evaluate the performance and experience based on different metrics like content correctness, time etc. Then they would evaluate how accurate this information is in regards to what they intend to send.

5. [Timeline]

- a. From now till April 15th our group plans to work on the core functionality, which is Phase . This includes basic email extraction and AI content generation.
- b. From April 15th till deadline we plan to work on creating a UI and possible secondary features as well as incorporate user testing.

6. [Task division]

- a. Praneeth will be working email extraction and storing info in the backend.
- b. Manav will be looking into how to integrate AI to generate responses.
- c. Ziyin will be looking into creating a frontend for users to interact with.