Title: An Application that Drafts Summary Emails with the Help of ChatGPT -

Interim Report

Department of Computer Science, The University of Hong Kong
Kwok Hin Tsoi - 3035743223

Supervised by Professor Xu, Dong

##

## Abstract

Some people spend a lot of time drafting emails. They may even lose their work due to crashes of computers. These issues may lead to a decrease in productivity. This project is developing an application utilising ChatGPT to draft summary emails to alleviate the negative impact. The application takes screenshots at time intervals to store the progress of the work. With the stored screenshots, the application will communicate with the server which will connect the ChatGPT application programming interface(API) to generate a personalized drafted email by converting the screenshots into useful information. The report highlights the successful implementation of the client-side code. However, several challenges remain, including improving OCR accuracy, enhancing the text classification algorithm's accuracy, and mitigating the negative impact of ChatGPT's instability. Testing, debugging, and ongoing efforts are essential to enhance the application's performance and stability.

## Table of Contents

[**Table of Contents**](#_6zru3vb3f7yr) **III**

[**List of Figures**](#_ho7eejxgzyl4) **IV**

[**Abbreviations**](#_3fe36v8q9lxw) **V**

[**Introduction 1**](#_hds5yb5ii51b)

[**Methodology 2**](#_f6ku0gtzvnul)

[**Result and Discussion 3**](#_6emhwdt5rss9)

[Figure 1 Sample user interface of the application. 3](#_2nl6t1w9ts6v)

[**Challenges 4**](#_4dca4cfsraz)

[**Future Plan 5**](#_1ejwhqifr8nu)

[**Conclusion 5**](#_8sejxshd8fsk)

[**References 6**](#_ptq7smh2t4ok)

## List of Figures

**Figure 1[sample user interface]****3**

##

## Abbreviations

| Abbreviation | Definition |
| --- | --- |
| OCR | Optical Character Recognition-technology that converts texts in images into texts in computer |
| API | Application Programming Interface |
| UI | User Interface |

##

## Introduction

Modern workers spend significant time and effort drafting emails. However, this task can be time-consuming and mentally taxing, taking away from more important work. According to Chui et al. (2012), workers allocate approximately 28% of their work time to managing emails on average. Upon the successful development and implementation of an email drafting application, it is expected that workers will experience a substantial productivity improvement. Meanwhile, the application mentioned not only tackles the challenges encountered by workers but also extends its benefits to general users. This inclusive approach enables users to save valuable time when composing emails, whether it be for personal communication with friends or other purposes.

Artificial Intelligence(AI) has achieved remarkable advancements in recent years, revolutionizing various industries and sectors. One of the most significant breakthroughs in this field is ChatGPT, an advanced language model developed by OpenAI. With the appearance of ChatGPT, a large amount of new applications with enhanced functionalities can be developed by integration. Therefore, this project aims to create an intuitive and user-friendly interface that empowers individuals to draft high-quality summary emails effortlessly to enhance productivity with the help of ChatGPT.

The user will be able to experience two key advantages with this application. Firstly, they will enjoy a significant reduction in the effort required to write emails, as the process is streamlined and productivity is enhanced. By leveraging ChatGPT's language generation capabilities, the application provides a reliable and efficient solution for summarizing emails. Users can effortlessly condense complex information into concise summaries, saving time and effectively conveying essential messages. Secondly, the application acts as a safeguard against work loss caused by crashes, ensuring that progress is preserved and protected. With built-in data review features, the risk of losing work is minimized. Users can confidently draft emails, knowing that their progress is safeguarded. This application empowers individuals to overcome the challenges of email composition and boost productivity in their email-related tasks.

The remainder of this paper will address the aims of the application, the implementation strategies employed, the functions that have been successfully implemented, potential challenges that may arise, and the conclusion. The ‘Methodology’ section will describe the approaches used to develop the application. The ‘Result and Discussion’ section will highlight the successfully implemented features. The ‘Challenges’ section will discuss possible obstacles and solutions. Finally, the ‘Conclusion’ will summarize project accomplishments and discuss the impact and benefits of the application.

## Methodology

The deliverable for this project will be a cross-platform application for Personal Computers including computers that work with macOS, Windows, and Linux operating systems. The application will be mainly developed using Python with a user interface(UI) library - ‘PyQt5’, image processing library - ‘Pillow’, and backend processing libraries including ‘sqlite3’. Furthermore, the Python library ‘Pytesseract’ will be used for doing Optical Character Recognition(OCR) in the code and the library ‘openai’ will be used for application programming interface(API) connection in the server-side code.

Python is chosen as the main programming language due to its simplicity and extensive library diversity. Python's simplicity allows for faster project completion, enabling focusing on other important aspects such as the accuracy of the OCR image-to-text technology and the efficiency of the search algorithm required in the data-processing phase. Meanwhile, the large amount of libraries of Python ensures access to suitable solutions for diverse needs in the project, accelerating the development process. These reasons make Python an ideal choice for development.

Upon installation and clicking the ‘start’ button, the application will hide in the toolbar, allowing users to freely utilize other functions while it runs in the background. The application will automatically capture screenshots at regular intervals (for example 5 minutes), storing them in a local database. The screenshots will then be converted into text employing OCR technologies. The extracted text will then be categorized into various types such as recipient, sender, and content. Along with the text, the screenshot, timestamp information, etc. will be stored in the database.

Users will have the flexibility to review the captured screenshots at their convenience. When they wish to generate a draft of a summary email, they can access a designated button that triggers a pop-up window for inputting the recipient names and extra information. Then, the application will search for relevant text stored in the database and, along with additional commands, will send the collected information to the server and hence the ChatGPT API. Finally, the server will send the output of the ChatGPT API to the clients, assisting users in composing the summary email based on the captured information and AI-generated insights.

## Result and Discussion



#### Figure 1 Sample user interface of the application.

The initial phase of the project focused on developing a basic UI for the application. A significant milestone has been achieved with the successful completion of the UI design. The UI incorporates a clean and intuitive layout. The screenshot functionality has also been implemented, allowing the application to store the screenshot in the specified directory.

The UI includes the ‘Start’ button for starting and stopping the screenshot function by clicking on it. Furthermore, there is the input box and the ‘Generate’ box for the user to enter the recipient email and send the request to the server side by clicking on the ‘Generate’ button. At last, there is also an output box for showing the return value from the server.

For the screenshot function, the application will start taking screenshots at 5-minute intervals. The captured screenshots will be automatically saved to the 'screenshot' sub-directory within the directory where the application is installed.

Several functions are still pending to implement in the application, including relevant data search functions and server-side code. Additionally, testing and debugging in the future are required to enhance the overall performance and functionality of the application.

## Challenges

This project presents several challenges that need to be addressed.

Firstly, the accuracy of OCR technologies is crucial for optimal application performance. Issues such as low-resolution images and complex screenshot layouts can impact the reliability of text extraction. To address this challenge, several strategies can be employed to enhance OCR accuracy. These include using other OCR libraries, and deciding to switch to an approach - collecting the email information from Gmail by consent. Also, providing users with guidance on methods to enhance the outcomes can be one of the ways to mitigate the impact.

Secondly, searching all relevant text related to the target recipient can be expensive, since using the screenshot approach may lead to redundant information. To not exceed the token limit of the ChatGPT API, better algorithm design is needed as it directly influences the effectiveness of the summary email generation.

Thirdly, the performance of ChatGPT might be unpredictable, potentially resulting in irrelevant or delayed responses, which adversely affects the user experience and hinders the resolution of the specified problem. It is important to inform users about the potential instability of the results. Additionally, closely monitoring the performance of ChatGPT can contribute to improving overall reliability.

## Future Plan

As all the major functions on the client side are already finished, it is planned to finish the server-side code, which is connecting to the ChatGPT API through the ‘openai’ library, in the second semester. Furthermore, to improve the accuracy and stability of the result, different approaches from OCR will be considered, such as retrieving information through consent as mentioned in the ‘Challenge’ part. Also, better algorithm design is needed to mitigate the negative impact of the potential challenges mentioned. Not only will these considerations be addressed, but also improvements will be made to the UI/UX design, such as enabling automatic launch after the window's launch, hiding the app icon in the bottom right of the toolbar, and facilitating app installation through an executable file.

## Conclusion

The progress report highlights the significant advancements made in our application, which utilizes ChatGPT for drafting summary emails. Throughout the reporting period, the client-side code has already been successfully implemented.

However, there are several foreseeable challenges to be solved in the remaining tasks mentioned in the paper. One of them is to improve the accuracy of OCR image-to-text conversion. To improve the accuracy, ensuring the quality of screenshots by guiding users can be one of the methods. Other methods include switching to an approach that is not using the OCR technology. Also, it is important to improve the accuracy of the search algorithm by having a more careful algorithm design. To mitigate the negative impact of ChatGPT's instability, it is crucial to inform users about the issue and closely monitor its performance. This approach ensures a great user experience while also enhancing overall performance.

It is also important to note that the plan involves completing the server-side code in the second semester, which will connect to the ChatGPT API using the 'openai' library. Additionally, different approaches will be explored to enhance the accuracy and stability of the results, including retrieving information through consent and improving the algorithm design to overcome potential challenges. Furthermore, improvements will be made to the UI/UX design, such as enabling automatic launch, hiding the app icon, and facilitating easy app installation

## References

Chui, M., Manyika, J., Bughin, J., Dobbs, R., Roxburgh, C., Sarrazin, H., Sands, G., &

Westergren, M. (2012). The social economy: Unlocking value and productivity

through social technologies. McKinsey. Retrieved from https://www.mckinsey.com/

industries/technology-media-and-telecommunications/our-insights/the-social-econom

y