Development Report: Generative AI-Powered Summarization Extension

Introduction:

After seeing newspaper companies using OpenAI's API to summarize their articles on the top of the page, I got the idea to make a Chrome extension that summarizes a full webpage. This way, I could summarize whichever blog or article I would like, and even important learning-material. It is also a cool way to get introduced to both making an extension, as well as implementing the GPT-3.5 Turbo API into my application. Similar techniques can be used to implement any existing API that uses highly trained Large Language Model (LLM) into Non-Playable Characters (NPCs) in games, chat robots and more. This project is highly work relevant and has futuristic potential.

Objective:

Develop a Chrome web extension that leverages generative AI to summarize the webpage a user is browsing into concise text.

Challenges and Solutions:

API Token Limitation: During the integration phase, I identified a token constraint in the GPT-3.5 Turbo model. If an article was too long, not all the words on the page would be handled, leading to a thrown exception. Changing to GPT-4 helped, but this is a more expensive model, so I chose another solution for this problem. I decided to truncate the input text to fit within the model's acceptable range, ensuring the essence of the content was retained. So basically: If the article was short enough to be summarized without being truncated, it just summarized as normal, but if the article was too large, the input text would be truncated, summarized, and with a warning printed after: "This summary is based on truncated text. Please upgrade to a premium subscription for complete text summaries."

Extension User Experience:

It was a delight when I got the first summary to work. However, I noticed several things that needed improvement. Through a lot of testing, I adjusted the design and added some new features to improve the user experience. For example, the summarization process takes roughly 10-15 seconds (which is slightly slower than expected), so I added a blinking text to show the user that the process is being handled. Another idea was to simulate real-time text writing similar to ChatGPT, but I had some performance issues regarding the quality of the summary, which led me to discard that idea. I also added buttons that let users copy the text to the clipboard, download it as a txt file, and refresh the summary. These functions were implemented in popup.js and popup.html. I tried first to make a language detector that detects the language of the input text and makes sure the output text is in the same language. This worked. However, I figured that giving the user the chance to select between English and Norwegian summaries themselves would be an even better user experience.

Conclusion:

This was a very fun and interesting project that I am proud of. It is impressive that such a great asset as the GPT API could be relatively easily integrated into my web extension app. I have learned a lot with this project, and knowing how to integrate a Generative AI-Powered API into my applications really opens up new opportunities in the future!

I am surfing the AI wave! 🌊 🏄 🤖 🚣 😎