**Development Report: Generative AI-Powered Summarization Extension**

**Group 12:**Marius Hauger, Hauk Hauge Mo and Lars Erik Risholm

**Introduction:**

After seeing newspaper companies using GPT API to summarize their articles on the top of the page, we got the idea to make a chrome-extension that summarizes a full webpage. This way you could summarize whichever blog or article you'd like. It’s also a cool way to get introduced to both making an extension, as well as implementing the GPT-3.5 turbo API into our application.

**Objective:**

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

**Division of Labor:**

With prior experience in Flask, setting up a backend development and API routing was the right initial job for Marius.

With some prior experience with using and installing Chrome extension-apps, Hauk took the challenge to learn the necessary difference between creating a webapp and chrome extension.

Lars took the job to signing up for the OpenAI account and focusing on the integration and optimal utilization of the GPT-3.5 Turbo API.

Although this was the initial labor-division, it was important that we worked together.

**Challenges and Solutions:**

**API Token Limitation:**

During the integration phase, we identified a token constraint in the GPT-3.5 Turbo model. If an article was too long, all the words on the page would not be handed, leading to a thrown exception. Changing to GPT4 helped, but this is a more expensive model, so we chose another solution for this problem.

We chose 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, a the input-text would be truncated, summarized and with a warning printed after: «This summary is based on a truncated text. Please upgrade to a premium subscription for complete text summaries.»

**Extension User Experience:**

It was a delight when we got the first summary to work. However, we noticed several things that needed improvement.

Through a lot of testing, we 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 we added a blinking text to show the user that the process is being handled. Another idea was to simulate a real-time text writing similar to ChatGPT, but we had some performance issues regarding quality of the summary, which led us to discarding that idea.

We also added buttons that let you copy the text to the clipboard, download as a txt file, and refresh the summary. These functions were made in popup.js and popup.html.

We tries first to make a language detector that detects the language of the input-text and makes sure the output text is the same language. This worked. However, we figured that by giving the user the chance to select between English and Norwegian summaries himself would be an even better user experience.

**Conclusion:**

This was a very fun and interesting project that we are proud of. It’s impressive that such a great asset as the GPT API could relatively easily be integrated into our web extension app.

We have learned a lot with this project, and knowing how to integrate a Generative AI-Powered API into our applications really opens up for new opportunities in the future!

We are surfing the Ai-wave🌊🏄‍♂️👨‍💻🏖😎