Discussions

Technology choices

Front-end:

- Tailwindcss: CSS framework that provides modern styling that can be easily applied to the html without having to worry about the stylesheet. This reduces the codebase size as we don't have to maintain a CSS file.
- Fetch: Easy to use library that facilitates connection to web APIs.

Back-end:

- Express: Popular web framework for Node.js. I have decided to use Express to build my API as it is widely used in the industry and there is a large user community that provides documentations and answers to questions that come up during development.
- Puppeteer: Node.js library which provides API to control Chromium with JavaScript. Within this project Puppeteer was used to access the user provided website and to extract information from this.
- Apos-to-lex-form: Tool to convert apostrophe-connecting words to their standard lexicon form.
- Lemmatizer: Package that converts words to their lexicon form. It helps us to compare normalized words, for example from opportunities we get opportune.
- Stopword: Package that removes so called stop words, that are words that do not modify the context of the information. By removing stopwords we are reducing the number of words that have to be analyzed.

Proposed features

- Author's writing tone: In this feature we would categorize an author based on their posts' combined the sentiment-analysist.
- API endpoint to extract positive/negative phrases: As we categorize a post‘s sentiment it would be interesting to extract the phrases that contribute to the sentiment analysis result.

- Image to text description: Using AI image processing tools we could extract more information from the website if we would also analyze the images.

Learning experiences

This project offered me an exciting opportunity to discover interesting technologies and to learn about the challenges within natural language processing.

The requirement of extracting information from a website pushed me to learn more about the DOM and how different elements can be reached within the document tree.

Having to implement both front-end and back-end offered me an insight into a full-stack developer's duties.

Being at the beginning of my software developer career this project provided me challenges which improved my problem solving capabilities and task management skills, having to break down the requirements into smaller tasks.