

[3-4 mins] Script for Video Demo:

Hi everyone!

We are the TextRulez team from UIUC's Fall 2023 CS 410 class, entitled Text Information Systems. The members of our team are Timothy Crosling, Deepthi Abraham, Avi Nayak, and Srishti Sharma.

Our project was centered around developing a tool that allows users to identify the key topics from a webpage and provides easy access to topic-related information. Via leveraging an inverted index along with text analysis to provide supplemental information to the user, we have developed a Chrome extension that possesses both a sidebar capability as well as a popup window upon click actions, in order to scrape text as needed from a web page. Tokenization and regular expressions are also present in JavaScript for pre-processing.

We discovered a method to connect our Python script to our Chrome extension, which is described in the demo. In order for new users to be familiar with product, we will exhibit our project from the standpoint of a new user.

First, it needs to be ensured that the appropriate Python libraries are present on the computer that the users computer, namely by running the command "pip install flask spacy numpy openai" (need to display on the screenshare)

Secondly, the python server needs to be kickstarted by running "python textAnalyticsServer.py" (need to display on the screenshare). And thirdly, the server's localhost address and port must be verified as 127.0.0.1:5000.

After these preliminary checks are completed, one must navigate (navigate to it on the screen) to chrome://extensions in the address bar, toggle 'Developer Mode' to ON, and click 'Load Unpacked', while navigating to the local directory where the project is stored. From there, the user can browse to a text-heavy web-page (such as a news site like CNN or Reuters), and toggle the side panel to the right of the plugin to get the popup and or sidebar view capability as needed (display on the screen).

Here is a sample result for <https://www.cnn.com/{some article}>

Here is another sample result for <https://www.reuters.com/{some article}>

Thank you for watching our demo, and as we have demonstrated, our project provides significant value to its users!

[2-3 mins] Script for Code Walkthrough:

Our submission consists of several files, which we will go into detail and explain the purpose of now, while walking through each file.

The meat of our project's analytics resides within the `textAnalyticsServer.py` file (display on screen), which has the top-level critical function of `"analyze_content"`. This function allows SpaCy-based named-entity recognition to take place, counts occurrences of entities to select the top 5 by importance, and then generates a GPT-based description with an HTML string to send back to the Chrome extension client.

Various `html/css/js/png` files are used as templates from which the pop-up is structured and generated, and in addition, we have a `manifest.json` file and a `background.js` file. The manifest file provides an overview of all the key components and settings, including permissions based fields/attributes and action-based fields/attributes (display on screen). Secondly, the `background.js` file consists of the necessary listeners set up with chrome runtime actions, in order to verify basic admin of the extension installation, and the passing of a response back to the chrome extender, based on information collated from text analytics.

[Can describe icons and runtime files as well if needed]