Deception Detective Fact - Checker

Team Members

- Andrew Nease: *neaseaw@mail.uc.edu*
- Daniel Wood: *wooddj@mail.uc.edu*
- Lando Slack: *slacklj@mail.uc.edu*

Advisor

- Dr. Yiming Hu: *huyg@ucmail.uc.edu*

Goals

This project was conceived to address the growing issue of misinformation spread online. Our goals are as follows --

- Provide a convenient and widely usable fact-checking experience
- Assign an easily understood factual rating to questionable statements
- If a factual rating cannot be assigned through our software, applicable resources (pertinent web links) are provided to the user to investigate further

Intellectual Merits

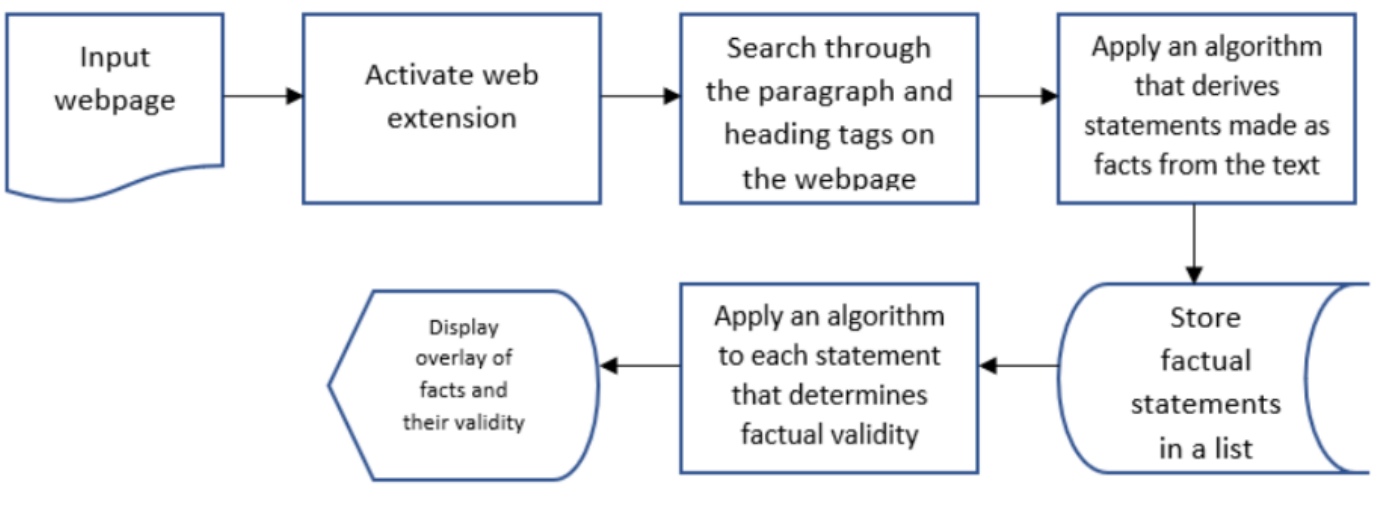
- “Atomic Statement Finding”:
  - Using part-of-speech tagging and syntax rules, we were able to develop a tool that approximates all the statements made within a body of text
- "Website Scraping and Crawling"
  - Querying a URL, reading the website's content, sorting the content for links to other pages, and consequently querying the links retrieved

Broader Impacts

Misinformation is a powerful tool used in manipulating populations. We believe that convenient access to the truth can help individuals overcome the novel dangers of today's Internet experience.

With continued development on this project, we envision automated fact-checking that would highlight falsehoods without user input.

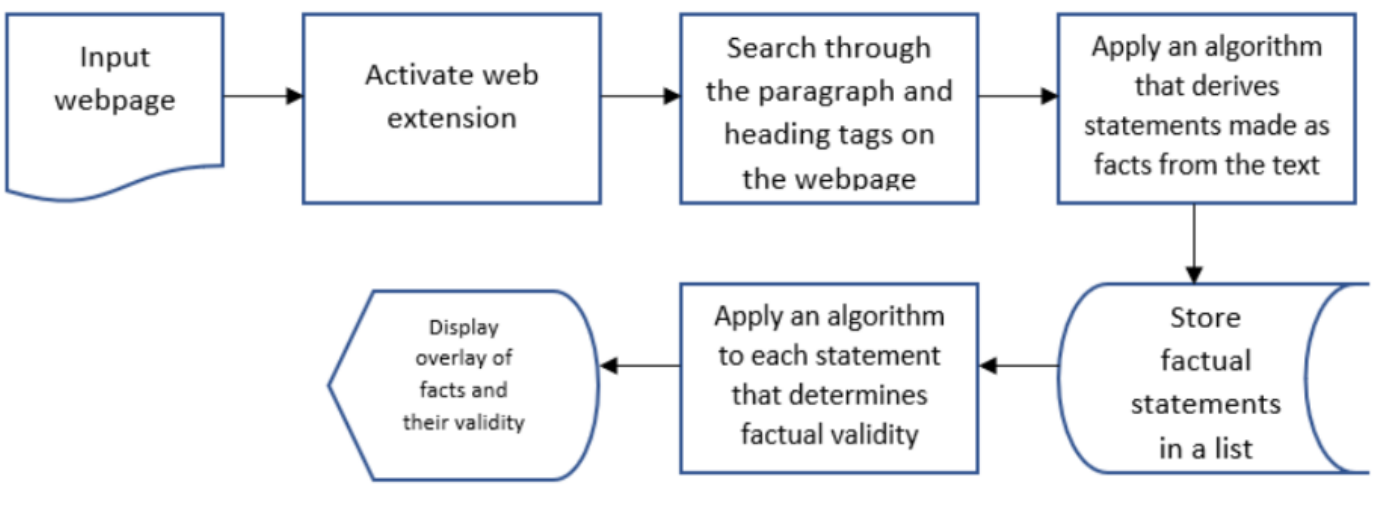
Design Specifications



Our project consists of four main components:

1. Web extension
2. Statement-finder backend
3. Fact-checker backend
4. Output report

The above design diagram for our project shows how these components work together.



Web extension: The user interacts with the extension to activate the tool

Statement-finder: The input from the web extension is processed to create a list of statements within the provided text

Fact-checker: Each statement is checked against one or more online source to determine its factual integrity

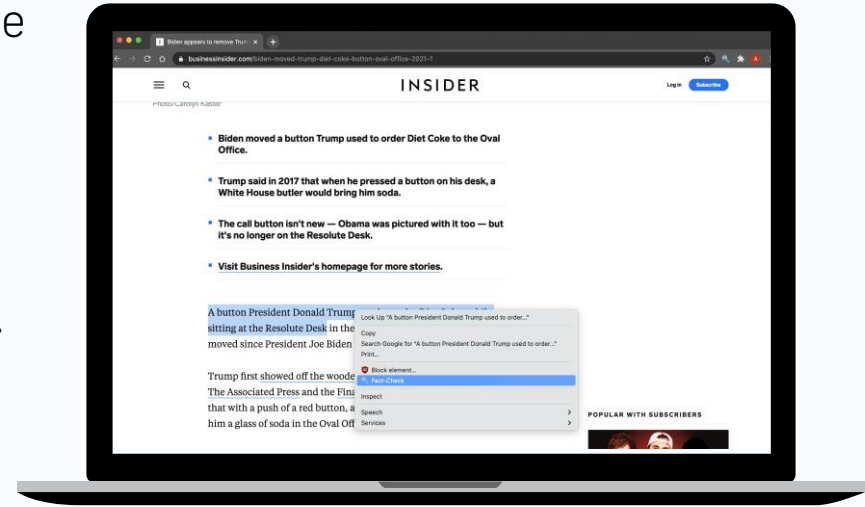
Output report: Using the statements and their results from the fact-checker tool, a report is displayed to show each statement's factuality rating and citation links for the user to explore

Design Specifications

Technologies

Fact-checking is accomplished by a separate server, so our front-end implementations are quite flexible.

We chose to create a Google Chrome extension for maximum end-user simplicity. Communication to the back-end server is accomplished with REST API endpoints.



Technologies

- neuralcoref: a Python library that provides tools for coreference resolution. We used this to develop a simple method for pronoun replacement
- Natural Language Toolkit (nltk): Python library with natural language processing methods. We used this for preprocessing our text data before running it through our statement-finder function
- Spacy: Another natural language processing library in Python. We used this for dependency and part-of-speech tagging in our atomic statement finder function

Technologies

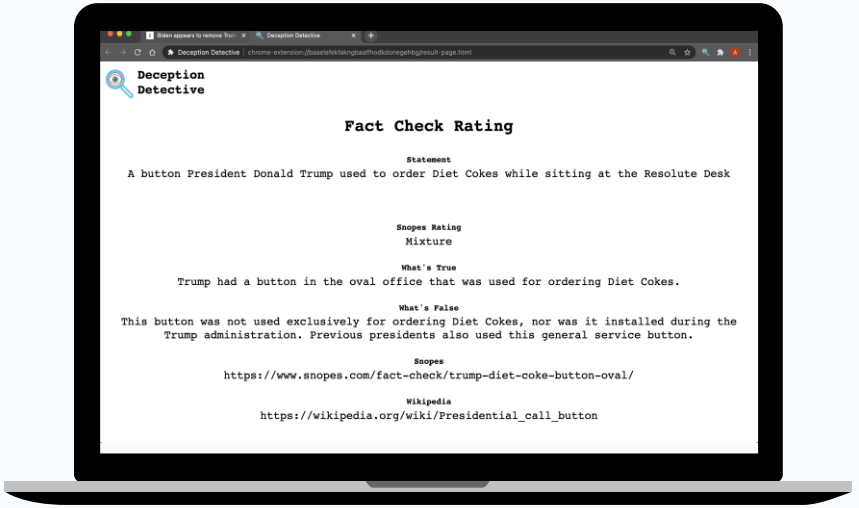
- Beautiful Soup – a Python library that allows smooth HTML parsing and splitting. Finding specific element tags and checking them for desired content such as links or paragraph text is at the crux of our web scrapers.
- requests – a Python library with tools to query website URLs straight from script code. We use the request() method to retrieve the HTML from the links we're interested in. The HTML is then passed into a Beautiful Soup object.

Milestones

	Start	Finish
Research	October 2020	January 2021
Back-End Design	January 2021	March 2021
Front-End Design	January 2021	February 2021
Testing and Refactoring	March 2021	April 2021

Results

- Manual fact-checking interface on Google Chrome
- Atomic statement finder that returns all statements that meet the syntax criteria
- Web scraper that parses Wikipedia and Snopes pages for fact-check ratings plus text content



Challenges

- Natural language complexity complicates programmatic statement finding
- What to do when our algo fails
- Automated fact-check
- Efficient web crawling and scraping
- Speeding up our software execution time for scalability