

# Team Info

## Team members and Roles:

Graham Cobden: LLM/RAG training/integration

Chi Dang: LLM/RAG training/integration

Ty Kemple: Front End/UI Design

Yuyang Lou: Database management,

Evan Mao: Weekly Report Writer, Database management

Raghavi Putluri: LLM/RAG training/integration

**Github:** <https://github.com/gycobden/No-Deception>

**Communication:** Slack (no-deception)

# Product Description

## Major Features:

- LLM/RAG integration:
  - Train LLM/RAG model on our dataset
- Accuracy Score:
  - Compare accurate sources to highlighted text and calculate score
- Highlight inaccurate statements in article
- Links to relevant articles

## Stretch Goals

- **Additional Information/Relevant Articles:** Alternative articles about the same topic
- Text-to-speech accessibility feature

# Use Cases

## Graham

Actors: UW student scrolling their Twitter timeline for entertainment while waiting in line for Firecracker

Triggers: Opening Twitter

Preconditions: Extension is fully enabled

Postconditions: Student exits the extension without believing a misleading post on their timeline

List of steps:

1. User opens Twitter with extension enabled
2. Scrolls through their timeline
3. Gets to post with infactual political claim

4. The infactual sentence is highlighted, score is displayed
5. Student ignores infactual post and continues scrolling
6. Student leaves Twitter when it's their time to order

Extensions/variations:

Student instead scrolls through a news outlet  
 Clicks on the title of an article  
 Reads through article  
 Score is displayed and a misleading claim is highlighted  
 Student decides the article is too biased and clicks out

Exceptions:

A timeline post contains an image/graphic  
 Extension cannot read the image, but highlights misleading information in another claim made in the post

## Chi

Actors: User is wondering if vaccines are harmful or not based on the things they've heard so they decide to read an article that explains why they are harmful

Triggers: User opening the article

Pre-conditions: The Chrome extension is already installed and fully functioning

Post-conditions: User now understands what's the truth and what's not (and why)

List of steps:

1. User opens the article
2. The extension highlights which lines have misinformation
3. The extension explains what's the truth and gives sources to trusted sources (in this case, could be the WHO, actual doctors/researchers, etc.) that back the claim
4. User leaves website when they're done

Extensions/Variations:

- User stops reading the current article to read the sources that the extension attaches
- User themselves realise that the article with misinformation doesn't sound right so they leave on their own

Exceptions:

- User ignores what the extension says because they're convinced that the article is correct
- The extension links sources that also have misinformation
  - For example, in this case, it could be trying to find sources that debunk vaccines being harmful but the keywords "vaccines" and "harmful" together might confuse the LLM into attaching sources which have those keywords, but are supporting the vaccines are harmful claim

## Yuyang

Actors: UW students working at Odegaard on their laptop browser decide to browse some popular reddit posts to take a break.

Triggers: User clicks on a reddit post under the forum “/politics” and that briefs a speech made by a member of the congress on vaccines.

Pre-conditions: The Chrome extension for No Deception is enabled and running on the browser

Post-conditions: User learns about the part of the speech made by the congress member that is not validated by scientific research and part of the speech that is factually correct.

List of steps:

1. User clicks on the post on the popular reddit page and gets directed to the full post
2. The extension parses the specific lines highlighted by the user and analyzes if its factually correct.
3. The extension remarks sentences, words in the highlighted lines that are factually incorrect or correct.
4. The user hovers mouse cursor over the remarked areas and triggers a pop-up window.
5. The window details specific mismatch between the text and scientific evidence and lists out its sources for the users to check himself.
6. User learns about the discrepancies and finishes reading the post but with their own judgements.

Extensions/Variations:

1. User finished reading without highlighting anything for the extension to perform checks on.
2. User clicks on the sources that the extension provided for fact-checking and decides to gather more information about the topic in the source articles directly.

Exceptions:

1. The extension failed to parse the lines that the user highlighted and analyze it.
2. The parsing and analysis took too long and the user finished reading the post before the extension had gathered any report and the user left the page.

## **Evan**

Actors: User who is reading a developing story of a fight between two students at their high school

Triggers: User clicks on first link of an article after searching up “<insert high school name here> fight”

Pre-conditions: Chrome extension already running on user’s device

Post-conditions: User is provided many sides of the story which can help them better piece together the cause of the fight

List of steps:

1. User gets redirected to a news report of an altercation at his high school
2. The extension may not have a score as this story is new and no highly-reputable sources confirm facts about the fight, so the score will be displayed as “Unavailable”
3. The extension will pop up asking if the user wants hyperlinks to other sources about the subject (regardless of if the score is available or not)
4. Once done reading the current report, user will go to other suggested articles to get more information that may confirm or disprove claims from the original article

Extensions/Variations:

- User clicks on another related article before finish reading the article they’re currently on

- User ignores the extension prompt until finished reading the article, then decides they want more information so they click “Yes” on the prompt and the extension will grant links to related articles

#### Exceptions:

- User chooses not to receive other articles related to the current article because they’re too lazy or have no time to look at other sources
- There are no related articles reporting on the high school fight

### **Raghavi**

Actors: A student is doing research on current political events in the US for a history project.

Triggers: User clicks on the first article after searching “political events in US” on Chrome Web

Preconditions: The chrome extension is installed and running

Postconditions (success scenario): User is given reference articles that debunk the fake news they came across and are given relevant articles to help steer their research.

#### List of steps (success scenario):

- User opens the article on Chrome with the extension installed
- User opens the article
- The extension pops up with a negative accuracy score and the reasoning to show why the article is inaccurate
- User is given relevant information for accurate research
- User can use the given relevant articles to conduct their research

#### Extensions/variations of the success scenario:

- User uses the given relevant articles to continue their researching with awareness of inaccurate vs accurate information regarding their topic
- User is able to notice the inaccuracies with the article on their own and do not use the extension’s resources

#### Exceptions: failure conditions and scenarios

- User does not notice the extension’s report and closes it to continue their research with inaccurate information
- The extension cannot detect that the article is inaccurate.

### **Ty**

Actors: A user is reading about a political speech and wants to know if what is being said is true.

Triggers: User clicks on an article about a political speech

Preconditions: The user has already downloaded No Deception

Postconditions (success scenario): The user is given contextual information on the politician’s claim, and directed to research about the politician’s claim

#### List of steps (success scenario):

- The user highlights a sentence from the speech
- They activate No Deception
- No Deception analyzes the claim
- The user is given new information to contextualize the claim and redirected to other sources about the same topic

#### Extensions/variations of the success scenario:

- User is given broader context for a speech
- User is given more tools to decide if they believe what they are hearing.

#### Exceptions: failure conditions and scenarios

- User doesn't activate the extension despite having it downloaded
- User closes out any pop-ups from the extension
- User has pop-ups blocked on their chrome

## Non-functional Requirements

- Clean, appealing, and non-intrusive UI
  - Possibly inspired by the design of Grammarly, i.e. with text highlighting
  - Points out passages of text containing possible misinformation and bias, while not covering it up
  - Doesn't take up a large portion of the screen at any given time (unless an article or post is filled to the brim with misleading information)
- Extension has features that can be toggled
  - What sites/text it will analyze
  - Whether it highlights text or not
  - Whether it gives recommendations
  - Whether you want to see the score
- Ensures user privacy
  - Doesn't record user browsing activity
  - Doesn't record text the user is reading if they don't want it to
- Updates with new information
  - If existing information is outdated or more is provided/discovered, the extension will be agile in incorporating it
- Easy to read/concise explanation
  - When misinformation is highlighted, the chunk of text that explains what's incorrect & explains what's correct should be concise
  - If it is too long, users may not read it and just ignore it, which defeats the purpose of the extension
- Dark/light mode depending on the device's settings
  - Just matches whatever theme the device has so it's aesthetically pleasing

## External Requirements

*The product must be robust against errors that can reasonably be expected to occur, such as invalid user input.*

- Flexibility across all kinds of user inputs

- If the user highlights invalid sentences such as punctuations, single words, or incomprehensible sentences, the extension should recognize and alert the user to adjust their inputs
- The extension should limit the user's inputs to a certain length to avoid lengthy computing time and possible hanging services.
- Robustness against internet and database connectivity issues
  - When internet connection is cut off, or connection to the database is cut off, the extension shuts down its running services and disables all of its functions and alerts the users of possible internet disconnection.
  - The extension should time out if a current running analysis is unresponsive and saves the user inputs to allow possible retries.

*The product must be installable by a user, or if the product is a web-based service, the server must have a public URL that others can use to access it. If the product is a stand-alone application, you are expected to provide a reasonable means for others to easily download, install, and run it.*

- We will be implementing this software as a Google extension
  - User just needs to receive a public URL to the google extensions webpage to our extension software
  - User will click “download” and it will automatically install into the Google browser, and the extension will start working once the user starts reading articles
- The user is free to disable or uninstall the extension with minimal effort

*The software (all parts, including clients and servers) should be buildable from source by others. If your project is a web-based server, you will need to provide instructions for someone else setting up a new server. Your system should be well documented to enable new developers to make enhancements.*

- The only software the user needs to have in order for this software to work is a working computer (Windows, MacOS, Linux), and the Google Chrome browser
- A strong internet connection is also ideal so that analyzing the highlighted text a user specifies won't take a long time
- Our github repository will have a detailed and clear README.md that will include a section that allows developers to contribute to the project by adding their own features

*The scope of the project must match the resources (number of team members) assigned.*

- Niche area of specialization
  - For the MVP, the extensions focus exclusively on analysis of scientific claims regarding vaccines. This very niche area can be covered with

relatively small size database and therefore won't require too much effort in data acquisition database management

- Utilization of existing LLM APIs
  - We fully utilize free, existing LLM APIs such as that provided by Gemini to perform retrieval augmented generation (RAG) that incorporate our own database to fact check reports.
- Manual analysis
  - Instead of automatically performing fact-checking on articles and blogs that users clicks into, which could lead into issues with scraping and parsing, for our MVP the users manually highlight and input lines of interests, which can be used directly by our extension to perform fact checking on.

## Team process description

### Group roles and schedule:

- **UI/front-end/backend:** This is important to implement the Google Extension and build the front-end that users can interact with

#### Members:

- Ty Kemple - Experience in web development and app creation for both google extensions and Discord, extensive experience in javascript

#### Software:

- React - Handles front end management
- Javascript - Handles any backend integration with our LLM and database work

#### Schedule:

- 4/23: Start working on app
- 4/30: Google extension created
- 5/7: Extension fully functional without full data implementation
- 5/14: Beta release
- 5/21: Full implementation with database and LLM
- 5/28: Bugs fixed, product finished

- **LLM/RAG training & integration:** Important in making sure that the LLM can learn what's true and what's not. The LLM can interpret the highlighted text and use our dataset to aid in calculating an accuracy score.

#### Members:

- Graham Cobden - Currently taking a machine learning course and has experience with Google collab and python
- Chi Dang - Worked with LLM APIs once, has experience in Python, and currently taking Intro to AI course
- Raghavi Putluri: Worked with LLM API's and data analysis, taken AI course.

#### Software

- Gemini API - Free to use and can be used to interpret both article text and our dataset

- Python - Useful for incorporating Gemini API and build its interaction with our data to calculate accuracy score.

#### Schedule

- 4/23: Finish requirements document and submit
- 4/30: Finish Architecture and design: set up Gemini API & install necessary packages.
- 5/7: Finish building an abstraction for LLM interaction with our dataset and user data.
- 5/14: Finish Beta Release and have use case 2 working
- 5/21: Get feedback and refine based on what users found to be hard to use/not working properly
- 5/28: Product finished
- **Database management:** We need a database management system in order to store a dataset of scholarly and trustworthy articles, research papers, and other texts that the extension can reference when determining how accurate a piece of text the user highlights in the article

#### Members:

- Evan Mao - Took CSE 344 (Introduction to Data Management) last quarter and has sufficient knowledge on PostgreSQL, SQLite3 and Java
- Yuyang Lou - Took CSE 344 Last quarter and has experience maintaining a chemical mass spectrometry database. Software toolset: PostgreSQL, MySQL, java.

#### Software:

- PostgreSQL- Stores and manages the source articles, research papers that we used for fact checking
- Java - We use java for the interface that connects the google extension to our database to perform queries on.

#### Schedule:

- 4/23: Finish requirements documents and submit
- 4/30: Draft ER diagram for our database and build a minimal dataset
- 5/7: Finish implementing our java interface to allow communication between the google extension and our database for RAG
- 5/14: Add more articles and research papers into our database
- 5/21: Full implementation with database and LLM
- 5/28: Bugs fixed, product finished

### **3 major risks:**

1. Our LLM doesn't work as intended e.g. links sources with misinformation when "correcting" the current misinformation the user is viewing
2. Chrome extension doesn't get approved by Google
3. Fact-checking for a news article vs a social media post may be different because of how they're written. In addition, if it's trying to fact-check the captions from a video or a reel, the captions might not translate what's being said accurately and that might affect the fact-checking



**External Feedback**

- We will get external feedback from the project manager after implementing a new feature, such as the bias score, or after changing the UI