BiasLens

Track: Development
Team members:

- Xinyun Ye [xinyuny3]
- Qichen Wang [qichen12]
- Yucong Chen [yucong3]
- Jiayuan Hong [jh79]

[Functions and Users]
We propose to develop a Chrome browser extension that performs automated bias analysis on news articles - **BiasLens**.
When a user opens a news webpage, the extension extracts the article content, analyzes the news source, author background, and contextual factors, then queries an LLM API to:

- Identify potential political or ideological bias based on tones, backgrounds, authors, related events etc.
- Do fact-checks and collect counter-perspectives based on external resources on web
- Generate summaries with references

The primary users are everyday news readers seeking balanced perspectives and researchers interested in media bias.

[Significance]
BiasLens helps address this issue by offering a lightweight, real-time tool that analyzes the article as users read it. It helps users recognize potential bias and think more critically about what they read, encouraging a more balanced understanding of the news.

[Approach]
The extension will be built using Chrome Manifest V3 with a lightweight frontend implemented in HTML, CSS, and JavaScript (or a framework like React if needed). Content scripts will extract and preprocess news content directly in the browser. The processed data is then sent to an LLM API (e.g., OpenAI ChatGPT-4o) to perform bias analysis and generate summaries. With a client-side design, we will address risks such as API latency and cross-origin issues through efficient asynchronous requests and proper API key management.

[Evaluation]
We will validate the tool by testing it across various news websites to ensure accurate content extraction and effective bias analysis. User testing and comparative evaluations with manual analysis will demonstrate both the usefulness and correctness of our implementation.

[Timeline]

- **Week 1:** Requirements analysis, environment setup, and UI design.
- **Week 2:** Development of content extraction scripts and initial UI implementation.
- **Week 3:** Integration of LLM API for bias analysis and counter-perspective generation.

- **Week 4:** Cross-site testing, optimization, and bug fixing.
- **Week 5:** Final refinements, documentation, and preparation for release.

[Task Division]

- **Qichen Wang (Frontend Engineer):** Design and implement the extension’s UI and user interactions.
- **Yucong Chen (API Integration Specialist):** Integrate and test the LLM API for bias analysis.
- **Xinyun Ye (Content Extraction Engineer):** Develop and optimize the content scraping and preprocessing scripts.
- **Jiayuan Hong (Project Manager & Architect):** Oversee overall system integration, manage the project timeline, and ensure quality assurance.