Software Design Document

for

Media Bias Classification System

Prepared By:

Group 14
22CSA16 MDL22CS049 Aravind Ashokan
22CSA30 MDL22CS093 Harishanker S Nair
22CSA50 MDL22CS154 Pradyumn R Pai
22CSA51 MDL22CS155 Pranav P S

February 19, 2025

Contents

1 Introduction **1**

2 Overall Description **2**

3 System Design **3**

4 Component Level Design **4**
- 4.1 User Interface Component 4
- 4.2 Pre-processing Component 4
- 4.3 Classification Component 4

5 Interface **5**
- 5.1 Extension Interface 5
- 5.2 Detailed Analysis Web Page 5
- 5.3 About Section of the Website 5

6 Data Flow **6**
- 6.1 User Input: 6
- 6.2 Pre-processing: 6
- 6.3 Classification: 6
- 6.4 User Feedback: 6

7 UML Diagrams **7**
- 7.1 Activity diagram 7
- 7.2 Use Case diagrams 8
- 7.3 DFD Level 0 9
- 7.4 DFD Level 1 10

1

Introduction

The Media Bias Classification System is designed to detect and categorize political bias in news articles using deep learning. It aims to highlight misleading or opinionated statements by analyzing linguistic patterns, contextual nuances, and sentiment shifts within the text. By leveraging advanced natural language processing techniques, the system provides users with a clearer understanding of how bias manifests in media. This system seeks to assist readers in identifying biased narratives, making informed judgments about news content, and fostering critical thinking. Additionally, it enables researchers, journalists, and policymakers to analyze trends in media bias, aiding in the creation of more balanced reporting and discourse.

Objectives:

- Detect and classify political bias in news articles with high accuracy.
- Flag misleading, opinionated, or agenda-driven content based on context and tone.
- Provide detailed insights and explanations to enhance media literacy and critical analysis skills.
- Assist researchers and policymakers in understanding and mitigating media bias.
- Encourage the development of unbiased and fact-based journalism.

2

Overall Description

The System provides functional and non-functional requirements, outlining its features and operational framework. The target users include:

- **Machine Learning Engineers**: Develop and train the model.
- **Developers**: Implement and maintain the system.
- **Stakeholders**: Journalists, researchers, and policymakers.

Testing and Maintenance:

- **Model Evaluation**: Using precision, recall, and F1-score.
- **Dataset Validation**: Ensuring data diversity.
- **User Testing**: Gathering feedback for improvements.
- **Performance Monitoring**: Updating the model for accuracy.

3

System Design

The system comprises a user interface, application logic, and a bias detection machine learning model. The extension interface will be developed using HTML and CSS, while JavaScript will handle the application logic. Standard web scraping tools will be used to extract data from the current webpage, which will then be sent via an API to the machine learning model. The model, running on the backend - using Python - will analyze the bias in the extracted data. The processed results will be sent back to the extension and displayed in the user interface. The project will utilize standard machine learning libraries commonly associated with Python.

4

Component Level Design

4.1 User Interface Component

- Responsible for handling User interactions.
- Consists of a web extension and a website.
- Handles errors from user inputs.
- The web extension sends text to the website.

4.2 Pre-processing Component

- Pre-processes text before classification.
- Performs lemmatization, multiword grouping, and lower casing.
- BERT tokenizer is used

4.3 Classification Component

- The pre-processed text is used as input for the trained machine learning model.
- The model classifies sentences as left, right, or center bias.

5

Interface

5.1 Extension Interface

The extension interface appears as a pop-up when the user interacts with the browser extension icon. It includes:

- **Article Analysis Display:** Shows the detected bias category (e.g., left, right, center) and a confidence score.
- **Brief Summary:** Provides a short textual explanation of the detected bias.
- **Read More Button:** Redirects the user to a web page with a detailed summary.

5.2 Detailed Analysis Web Page

When a user clicks on the **Read More** button, they are redirected to a dedicated web page containing:
Comprehensive Bias Report: Includes an in-depth analysis of the article's language, tone, and sources.

5.3 About Section of the Website

The website features an **About** section to provide users with information about the extension, its purpose, and the technology behind it. This section includes:

- **Purpose of the Extension:** Explains the goal of the tool in identifying media bias and promoting informed news consumption.
- **Technology Used:** Brief overview of NLP models, machine learning techniques, and datasets employed.
- **Developer Information:** Credits and background information about the development team.
- **Contact and Support:** Provides ways for users to reach out for inquiries, feedback, and support.

6

Data Flow

6.1 User Input:

- User interacts with the web extension (clicks the icon).
- Sends text (URL or input) to the website for processing.

6.2 Pre-processing:

- Text undergoes lemmatization, multiword grouping, and lowercasing.
- BERT tokenizer is used for text tokenization.
- Processed text is sent to the Classification component.

6.3 Classification:

- Pre-processed text is passed to the trained machine learning model.
- Model classifies text into categories: Left, Right, or Center.
- Classification results (bias category and confidence score) are sent to the website.

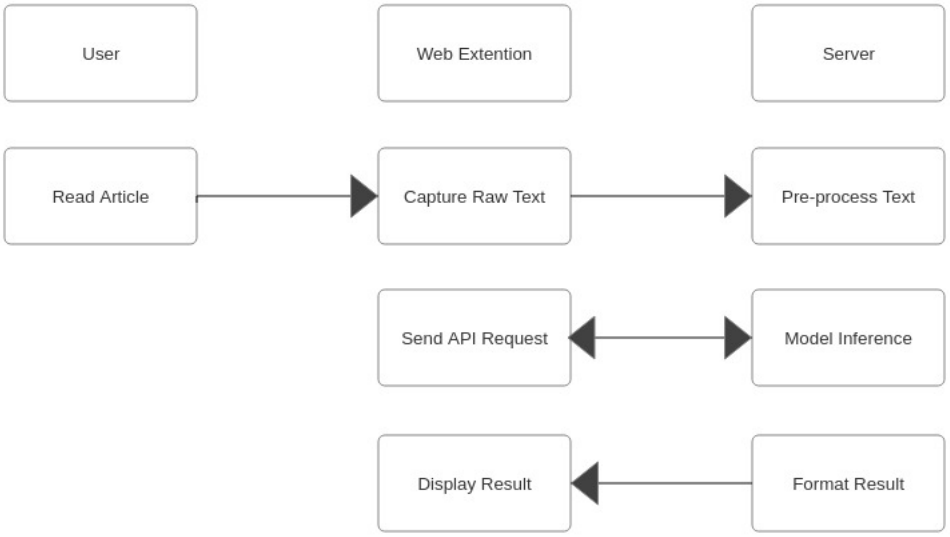
6.4 User Feedback:

- Website updates extension UI with:
  - Bias category and confidence score.
  - A brief summary of detected bias.
  - "Read More" button linking to a detailed analysis page.

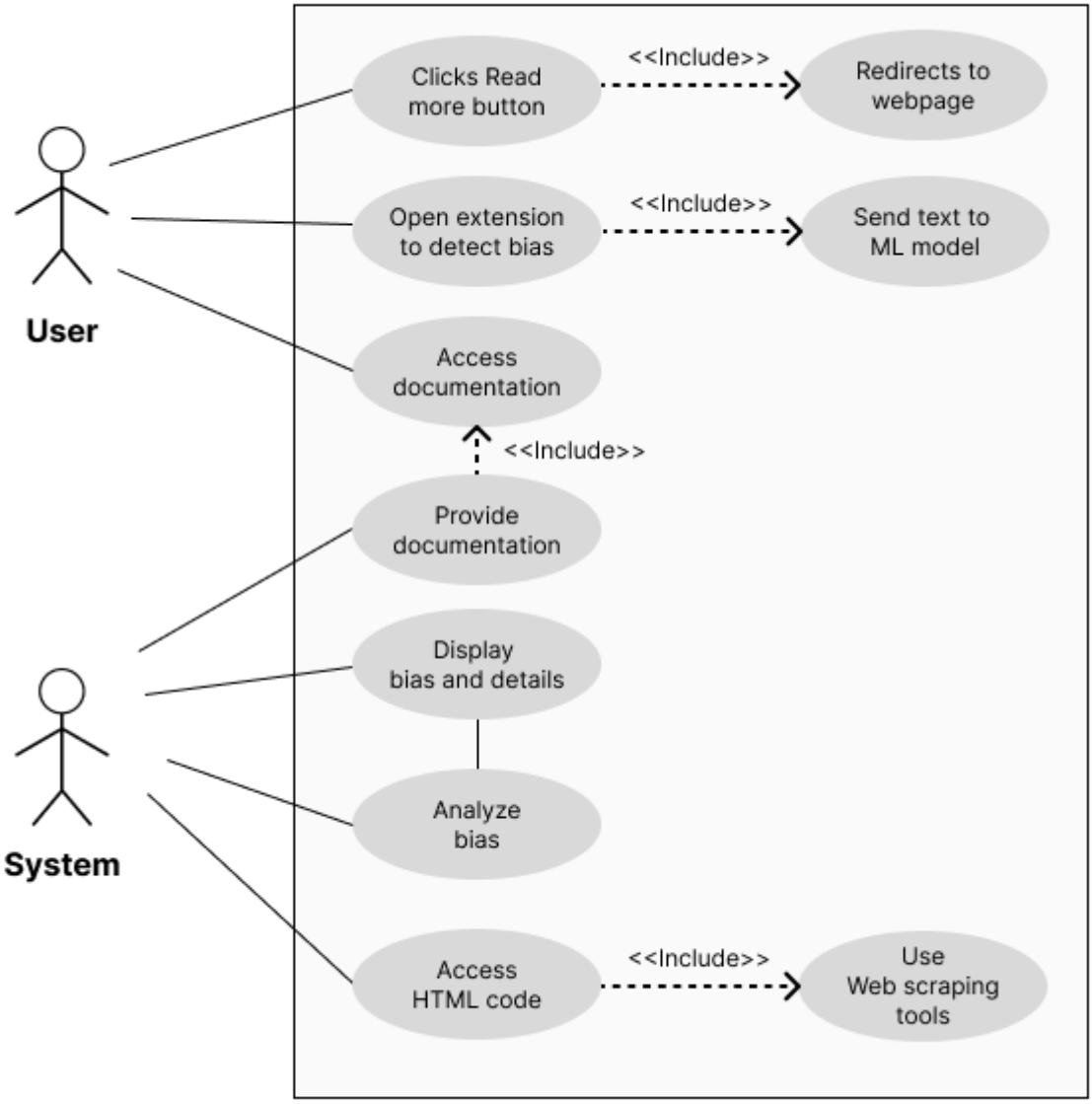
7

UML Diagrams

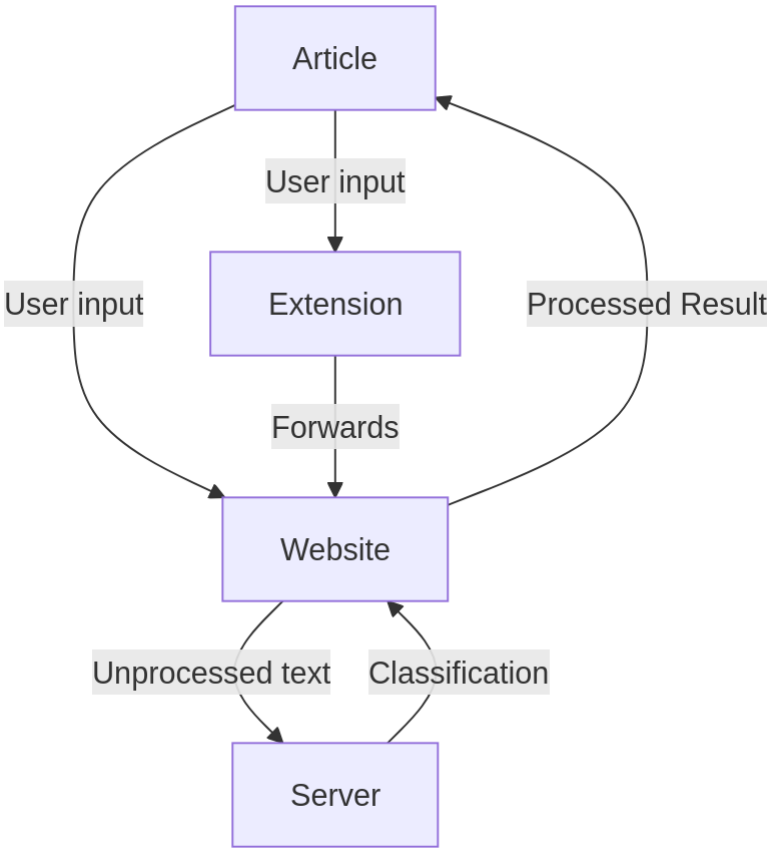
7.1 Activity diagram



7.2 Use Case diagrams



7.3 DFD Level 0



7.4 DFD Level 1