

Software Requirements Specification

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 4, 2025

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Intended Audience	1
1.3	Project Scope	1
1.4	Overview of Developer's Responsibilities	1
1.4.1	Data Collection	2
1.4.2	Text pre-processing	2
1.4.3	Model Development	2
1.4.4	Sentence level detection	2
2	Overall Description	3
2.1	Product Perspective	3
2.2	Product Functions	3
2.3	User Class and User Characteristics	3
2.4	Operating Environment	3
2.5	Design and Implementation Constraints	4
2.6	User Documentation	4
2.7	General Constraints	4
2.8	Assumptions and Dependencies	4
3	External Interface Requirements	5
3.1	User Interfaces	5
3.2	Hardware Interfaces	5
3.3	Software Interfaces	5
3.4	Communication Interfaces	6
4	Hardware and Software Requirements	7
4.1	Hardware Requirements	7
4.2	Software Requirements	7
5	Functional Requirements	8
5.1	Functional Requirement 1: Analyze News Articles	8
5.2	Functional Requirement 2: Classify Bias	8
5.3	Functional Requirement 3: User-Friendly Interface	8
5.4	Functional Requirement 4: Real-Time Processing	8
5.5	Functional Requirement 5: Support Multiple Browsers	8

6	Non-Functional Requirements	9
6.1	Non-Functional Requirement 1: Performance Efficiency	9
6.2	Non-Functional Requirement 2: Security Compliance	9
6.3	Non-Functional Requirement 3: Data Privacy Compliance	9
6.4	Non-Functional Requirement 4: Accuracy and Reliability	9
6.5	Non-Functional Requirement 5: Maintainability and Updates	9

1

Introduction

1.1 Purpose

The purpose of this document is to capture in natural language and at a functional level the description and requirements of Media Bias Classification System. The system uses deep learning techniques to classify news articles and detect misleading sentences. The project aims to promote critical thinking and increase media literacy by detecting political bias in news articles.

1.2 Intended Audience

This document serves to explain both the functional and non functional requirements of Media Bias Classification System. It provides readers with a clear understanding of the features, architecture, and functionality of the system. The intended audience includes:

- Machine Learning Engineers: Responsible for developing classification model.
- Developers: Responsible for implementing and maintaining the system
- Stakeholders: Those who use the system and examine news articles

1.3 Project Scope

Politics are a very polarizing topic and news media carry their biases. News articles often use misleading and biased statements with the goal of promoting the political agenda of the people writing and editing the article. Often, such statements escape the notice of a layperson. Our system aims to solve this issue using deep learning model that could warn readers about potentially biased or misleading sentences present in the article. Objectives:

- Detect bias in articles
- Flag misleading sentences
- Increase political awareness

1.4 Overview of Developer's Responsibilities

The system will be built using sophisticated deep learning techniques to detect political biases. Below are the specific responsibilities:

1.4.1 Data Collection

- Collect data from online sources
- Assign label to data
- Augment data and inject noise to ensure robustness

1.4.2 Text pre-processing

- Use tokenization, lemmatization, lowercasing, and multi-word grouping to pre-process text
- Develop pre-processing pipeline that processes news articles before classification

1.4.3 Model Development

- Develop and train deep learning based classification model using BERT
- Tune hyperparameters

1.4.4 Sentence level detection

- Design a web extension to detect and highlight misleading sentences using the trained model

Overall Description

2.1 Product Perspective

The Media Bias Classification Web Extension is a browser-based tool designed to analyze the bias present in online news articles. It integrates seamlessly with web browsers, acting as an auxiliary tool for readers who wish to gain insight into the credibility and political slant of news content. This extension leverages Natural Language Processing (NLP) techniques and machine learning models to evaluate articles and categorize them based on their bias.

2.2 Product Functions

- Analyze the content of a news article upon user request.
- Classify the article into predefined bias categories (e.g., left, right, center, extreme, unreliable).
- Display bias information in an intuitive and user-friendly interface.

2.3 User Class and User Characteristics

- **General Users:** Individuals interested in assessing the reliability and bias of news articles they read.
- **Journalists and Researchers:** Professionals seeking unbiased news consumption and fact verification.
- **Students and Academics:** Users conducting studies on media bias and misinformation.

Users will have varying levels of technical expertise, so the extension must be intuitive, easy to use, and require minimal configuration.

2.4 Operating Environment

- **Supported Web Browsers:** All Chromium-based (e.g., Chrome, Edge, Brave) and non-Chromium-based browsers (e.g., Firefox).
- **Backend Services:** Hosted on cloud platforms.
- **Dependencies:** Machine learning models for text classification.
- **Minimum System Requirements:** Standard browser extension compatibility with low memory and CPU usage.

2.5 Design and Implementation Constraints

- Must comply with browser extension security policies and content script limitations.
- Accuracy of bias classification depends on the training dataset and ML model used.
- Should be lightweight to avoid browser performance issues.
- Adherence to data privacy laws.

2.6 User Documentation

- Installation guide for supported browsers.
- User manual detailing how to use the extension.
- Explanation of bias categories and classification methodology.

2.7 General Constraints

- Limited access to proprietary news sources due to paywalls.
- Accuracy constraints due to evolving media biases and language nuances.
- Must avoid excessive API calls to maintain cost efficiency and performance.

2.8 Assumptions and Dependencies

- Assumes users have an active internet connection for real-time analysis.
- Relies on an model trained on diverse and up-to-date datasets.
- Assumes the extension will be updated regularly to refine classification algorithms.

3

External Interface Requirements

3.1 User Interfaces

- The web extension will have a simple and intuitive UI integrated into the browser.
- It will display bias detection results via pop-ups or a dedicated extension panel.
- Users can manually input URLs or rely on automatic scanning of visited web pages.
- Color-coded bias indicators (e.g., green for neutral, yellow for slight bias, red for strong bias) will be displayed.
- A settings page will allow users to customize sensitivity levels and reporting preferences.

3.2 Hardware Interfaces

- The extension will not require any specialized hardware.
- It will interact with the local machine's storage to save user preferences and logs.
- If necessary, it may utilize a local GPU for model inference acceleration (optional).

3.3 Software Interfaces

- The extension will integrate with web browsers (e.g., Chrome, Firefox, Edge) via their respective extension APIs.
- It will communicate with a locally hosted backend server for model inference.
- API endpoints will handle:
 - Fetching and analyzing news articles
 - User authentication (if needed)
 - Model updates and retraining
- The extension will utilize machine learning libraries such as TensorFlow/PyTorch and NLP frameworks like spaCy or Transformers.

3.4 Communication Interfaces

- The extension will use HTTP requests to interact with the locally running backend.
- It may support WebSocket connections for real-time updates.
- Secure communication protocols (e.g., HTTPS, WSS) will be used for data transfer.
- If an external API is used for additional data, appropriate authentication mechanisms (e.g., API keys, OAuth) will be implemented.

4

Hardware and Software Requirements

4.1 Hardware Requirements

Server Side:

- High Performance CPU
- GPU for model training
- RAM 8GB at least

Client Side:

- Desktop, Laptop, or Mobile Phone with internet connection
- Browser (Chromium and Non-chromium based)

4.2 Software Requirements

- Operating system (Linux, MacOS, Windows etc)
- Frontend (HTML, CSS, Javascript, React)
- Backend (Python, FastAPI/Flask)
- Text pre-processing (NLTK, BertTokenizer)
- Deep learning model (Transformers, Tensorflow)

Functional Requirements

5.1 Functional Requirement 1: Analyze News Articles

- Extract and process text from user-selected articles using NLP.

5.2 Functional Requirement 2: Classify Bias

- Categorize articles into predefined bias categories (left, right, center, extreme, unreliable).
- Display classification results to the user.

5.3 Functional Requirement 3: User-Friendly Interface

- Provide an intuitive UI with color-coded labels.
- Display confidence scores and explanation tooltips.

5.4 Functional Requirement 4: Real-Time Processing

- Ensure quick analysis with minimal delay for seamless user experience.

5.5 Functional Requirement 5: Support Multiple Browsers

- Ensure compatibility with Chromium-based browsers (Chrome, Edge, Brave) and Firefox.

Non-Functional Requirements

6.1 Non-Functional Requirement 1: Performance Efficiency

- The extension must be lightweight, ensuring low memory and CPU usage.

6.2 Non-Functional Requirement 2: Security Compliance

- Must follow browser security policies to prevent unauthorized data access.

6.3 Non-Functional Requirement 3: Data Privacy Compliance

- No personal user data should be stored.
- Must comply with global privacy laws like GDPR.

6.4 Non-Functional Requirement 4: Accuracy and Reliability

- The classification must be based on an up-to-date ML model.
- The model should be trained on diverse datasets to minimize bias.

6.5 Non-Functional Requirement 5: Maintainability and Updates

- The extension should receive regular updates to refine classification accuracy.
- Must adapt to evolving media biases and trends.