

Software Requirement Specification (SRS)

Project Name: Fake News Detection using NLP

1.0 Title of the project

Fake News Detection using NLP

2.0 Introduction

Fake news has become a significant issue in today's digital world, affecting politics, public opinion, and even global health. The rapid spread of misinformation via online platforms calls for automated systems that can effectively identify and filter out such content. This project aims to address this issue using natural language processing (NLP) and machine learning techniques.

2.1 Purpose

The primary purpose of this project is to design and implement a fake news detection system using Natural Language Processing (NLP) and machine learning techniques. The system will classify news articles as real or fake based on textual analysis. The focus is on achieving accurate classification using lightweight, interpretable models suited for academic use.

2.2 Document Conventions

None

2.3 Intended Audience and Reading Suggestions

This document is intended for students, project guides, and evaluators. It can also benefit:

- **Journalists** looking to verify news articles quickly.
- **Government agencies** for monitoring misinformation campaigns.
- **Social media platforms** for detecting misleading content before it spreads.

2.4 Project/Product Scope

In addition to academic demonstration, the system can be extended to:

- **Integrate into newsroom pipelines** to verify incoming news.
- **Support fact-checking portals** and public education platforms.

- **Provide APIs** for developers to build fake news detection apps.
- **Assist researchers** in analyzing misinformation trends over time.

- Misinformation analysis dashboards for journalists and researchers.

Research papers on fake news detection, TensorFlow, Keras, and NLP documentation.

3.0 Overall Description

This section provides an overview of the system, including its functionality, use cases, and constraints. The fake news detection system utilizes NLP to preprocess text and then applies machine learning models to classify content as fake or real.

3.1 Project/Product Perspective

Governmental or NGO information verification platforms

1. Accept news content as input.
2. Preprocess the text using NLP.
3. Classify the news using a trained LSTM model.
4. Display the result as 'Fake' or 'Real'.

3.2 System Constraints

The implementation is based on open-source tools and is constrained to Python-based environments.

MLOps Integration: For real-world deployment, the model can be wrapped in a pipeline that supports continuous monitoring, logging, and periodic retraining using tools like MLflow, Docker, and GitHub Actions for CI/CD.

3.3 Operating Environment

3.3.1 Server:

OS: Windows

Language: Python

Database: None

Application Server: Localhost

Tool: Jupyter Notebook, Anaconda

3.3.2 Client:

Browser-based UI or command-line interface

3.4 Design and Implementation Constraints

Limited to open-source tools and academic license.

Python-based implementation.

3.5 User Documentation

User manual and a ReadMe file with instructions will be included.

4.0 External Interface Requirements

4.1 User Interfaces

Simple web or command-line interface. Compatible with browsers like Chrome, Firefox.

4.2 Hardware Interfaces

Standard development machine with minimum 8 GB RAM and 256 GB HDD.

4.3 Communications Interfaces

No external communication needed. Optional integration with automated email for alerts.

5.0 System Features

1. User Input Interface
2. Text Preprocessing
3. LSTM-based Classification
4. Result Display

6.0 Non-functional Requirements

- Performance Requirements: Should process input and return result within 5 seconds.
- Safety Requirements: No critical safety concerns.
- Security Requirements: Local system only, no network exposure.
- Software Quality Attributes: Usability, Reliability, Maintainability

7.0 Acceptance Criteria

The system should correctly classify at least 90% of the test dataset.
Must run without errors on the test environment.

8.0 Deliverables

1. Source code
2. Trained model
3. Documentation
4. Sample dataset
5. Project report