

Emotion and Gibberish Detection Project

Table of Contents

1. Project Overview
2. Architecture
3. Setup Instructions
4. Docker Setup
5. Quality Metrics

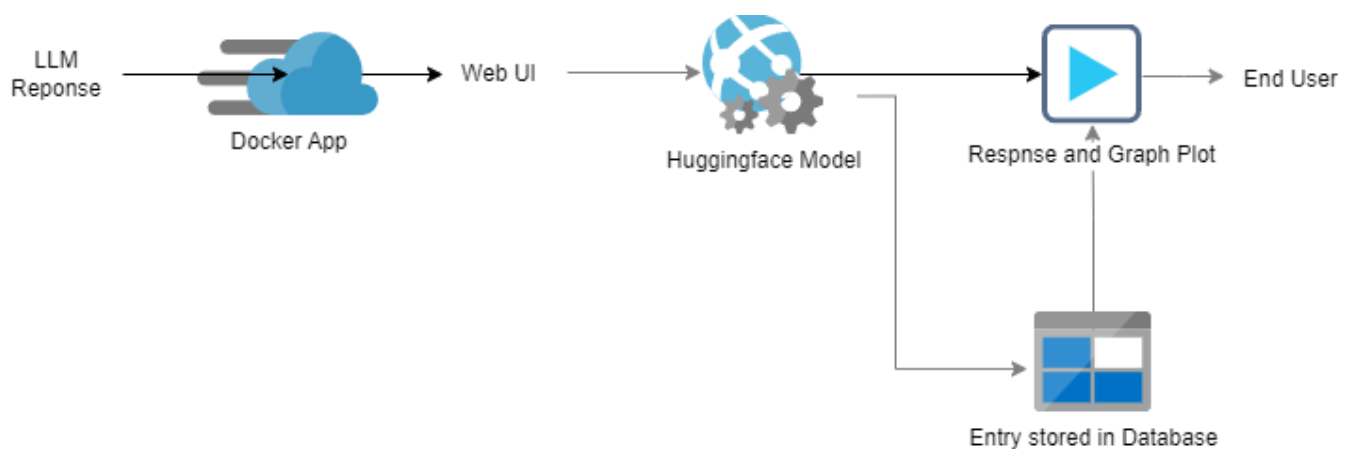
Project Overview

This project is a web application that allows users to analyze text for emotions and gibberish content using pre-trained models from HuggingFace. The application is built using the Python and Flask for the backend API, and Docker is used for containerization to ensure easy deployment on any cloud provider.

Architecture

The architecture of this project is designed to be modular and scalable. The main components are:

1. **Frontend:** The web UI built with HTML and Bootstrap.
2. **Backend:** A Flask application that serves the REST API.
3. **Database:** SQLite for storing the analysis results.
4. **ML Models:** HuggingFace pipelines for emotion and gibberish detection.
5. **Docker:** Containerization for easy deployment.



Component Interaction

- The user interacts with the web UI to input text.
- The web UI sends the text to the Flask backend via a POST request.
- The Flask backend uses the HuggingFace models to analyze the text for emotions and gibberish.
- The results are stored in the SQLite database and displayed back to the user.
- Docker is used to containerize the entire application for deployment.

Setup Instructions

Prerequisites

- Python 3.8+
- Docker

Docker Setup

Build and Run the Docker Container

1. Build the Docker Image:

```
docker build -t emotion-gibberish-app .
```

2. Run the Docker Container:

```
docker run -p 5000:5000 emotion-gibberish-app
```

3. Access the Web UI:

Open your web browser and go to `http://localhost:5000`.

Quality Metrics

Functional Metrics

- **Accuracy of Emotion Detection:** The precision of the emotion classification model.
- **Accuracy of Gibberish Detection:** The precision of the gibberish detection model.
- **Response Time:** The time taken to analyze the text and return the results.

Non-Functional Metrics

- **Scalability:** The ability to handle multiple simultaneous requests without degradation in performance.
- **Reliability:** The stability of the application under various conditions.
- **Maintainability:** The ease of updating and maintaining the codebase.
- **Portability:** The ability to deploy the application on various platforms using Docker.

Key Files

- `App.py`: Main entry point for the Flask application.
- `Dockerfile`: Docker configuration file for containerizing the application.
- `requirements.txt`: List of Python dependencies.
- `index.html`: The main HTML template for the web UI.

Emotion and Gibberish Analysis

Received text:

This is a text example

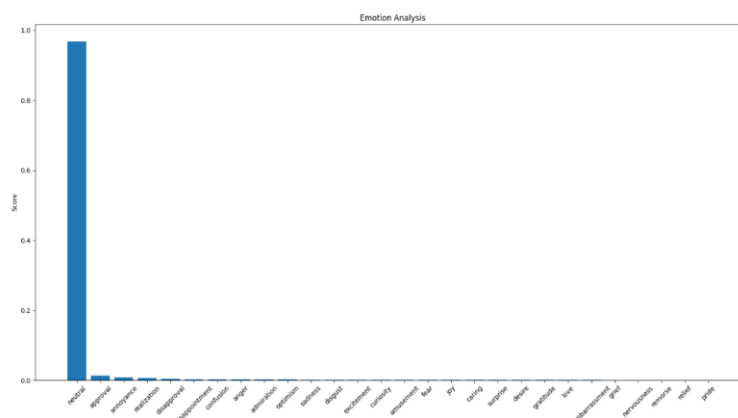
SubmitReset

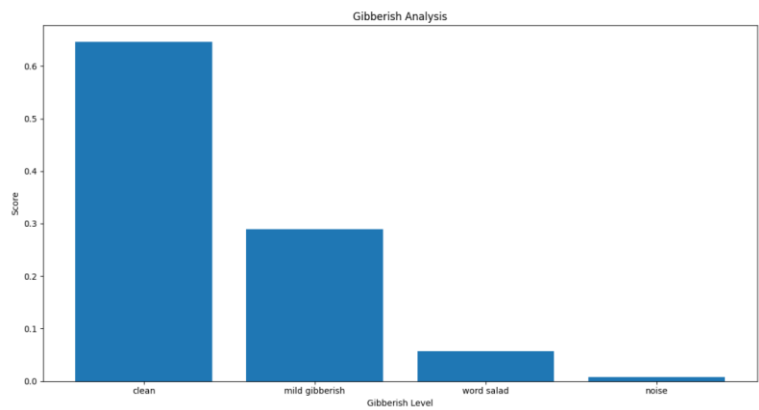
Response

Emotion: neutral (Score: 0.968)

Gibberish: clean (Score: 0.646)

Plots





History

ID	Text	Emotion Results	Gibberish Results
1	This is a text example	Emotion: neutral (Score: 0.968)	Gibberish: clean (Score: 0.646)