Sentiment Analysis Backend Using NLP & Embeddings

📌Overview

This project is a **backend-only Sentiment Analysis system** that analyzes user-provided text and classifies it into **Positive**, **Negative**, or **Neutral** sentiment. The backend is exposed as a **REST API**, making it easy to integrate with web or mobile applications.

The system uses **Natural Language Processing (NLP)** and **text embeddings** to understand the semantic meaning of text rather than relying only on keywords.

🕖Objectives

- Perform accurate sentiment analysis on text input
- Build a scalable and reusable backend API
- Help beginners understand NLP and embeddings
- Enable easy frontend or third-party integration

⚙️Features

- RESTful API for sentiment prediction
- Text preprocessing and normalization
- Embedding-based semantic analysis
- Fast and lightweight backend
- Clean and beginner-friendly code structure
- Easily extendable for future features

Technologies Used

- **Python**
- **Flask / FastAPI** (Backend Framework)
- **OpenAI Embeddings / NLP Models**
- **NLTK / spaCy** (Text Preprocessing)
- **JSON** for API communication

💣Project Structure

```
sentiment-analysis-backend/
│
├── app.py                  # Main API server
├── sentiment.py            # Sentiment prediction logic
├── embeddings.py           # Embedding generation
├── preprocessing.py        # Text cleaning and normalization
├── requirements.txt        # Project dependencies
├── .env                    # API keys (ignored in GitHub)
└── README.md               # Project documentation
```

🔦Working Flow

1. User sends text input via API request
2. Text is cleaned and preprocessed
3. Embeddings are generated for the text
4. Sentiment is predicted based on embeddings
5. Result is returned as a JSON response

🔌API Usage

Endpoint

```
POST /predict-sentiment
```

Request Body

```
{
  "text": "This product is amazing!"
}
```

Response

```
{
  "sentiment": "Positive",
  "confidence": 0.92
}
```

🗜️ How to Run the Project

✂️ Clone the Repository

```
git clone https://github.com/your-username/sentiment-analysis-backend.git
cd sentiment-analysis-backend
```

✒️ Install Dependencies

```
pip install -r requirements.txt
```

🛏️ Set Environment Variables

Create a `.env` file and add your API key:

```
OPENAI_API_KEY=your_api_key_here
```

🌂 Run the Server

```
python app.py
```

🌡️ Test the API

Use Postman, Curl, or a browser to send requests.

🔒 Security Considerations

- API keys are stored using environment variables
- `.env` file is excluded from GitHub
- No user data is stored permanently

🗒️ Future Enhancements

- Multi-language sentiment analysis
- Emotion detection (happy, sad, angry, etc.)
- Database integration
- Frontend dashboard
- Real-time social media sentiment analysis

💻Author

Ashwin Karuppusamy
Beginner Developer | NLP & AI Enthusiast

📜License

This project is open-source and available for educational and learning purposes.