

User Manual

Dependency / Package	Version / model
Python	3.9.11
Django	~=4.2.2
Django REST Framework	~=3.14.0
transformers[tf-cpu]	TensorFlow CPU version
Pre-trained model	nlptown/bert-base-multilingual-uncased-sentiment

Project Guideline:

1. Open the project folder "sentiment_analysis" with a suitable IDE such as PyCharm/ VS Code
2. Open a new terminal window
3. Create a virtual environment with the following command:
Windows OS
> py -m venv venv

Unix/ Mac OS
> python -m venv venv
4. Activate the virtual environment using the following command for Windows OS:
Windows OS
> venv/Scripts/activate

Unix/ Mac OS
> source venv/bin/activate
5. Then install the dependencies using the command:
pip install -r requirements.txt
6. After the successful installation, run the following command to run the server:
python manage.py runserver

Now the project is running on the localhost server.

You can now hit the url: <http://localhost:8000/analyze> in a browser to access the browsable API or test it through Postman in the following way:

1. Send a POST request with request body in JSON format s follows:

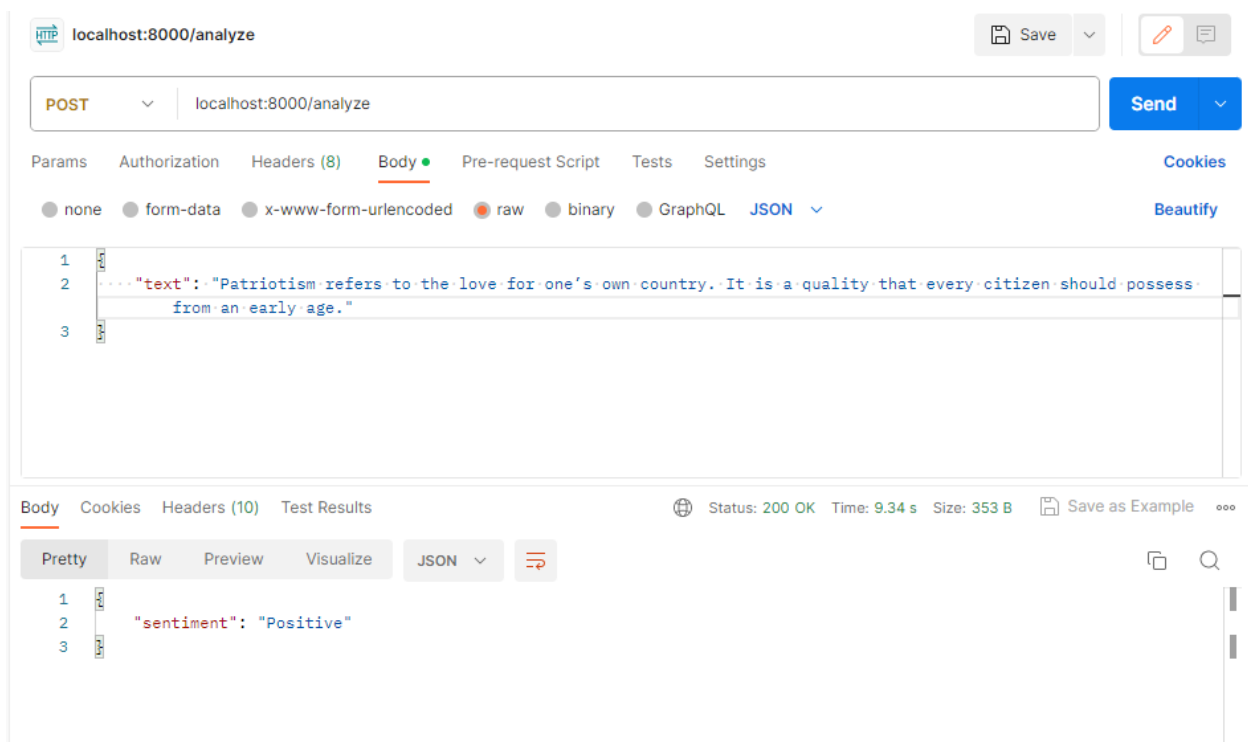
```
{  
  "text": "Your text to be analyzed"  
}
```

2. Then wait for a while to get the response. The response will appear in the following format:

```
{  
  "sentiment": "Positive / Negative / Neutral"  
}
```

Here some examples are given both from Postman and browsable API:

Positive sentiment:



Negative sentiment:

The screenshot shows a REST client interface with a POST request to `localhost:8000/analyze`. The request body is a JSON object with a `text` field containing a sentence about load-shedding. The response status is 200 OK, and the response body is a JSON object with a `sentiment` field set to "Negative".

```
POST localhost:8000/analyze
```

```
{  "text": "Load-shedding means the discontinuation of the supply of electricity. Load-shedding occurs when the generation of power is less than the demand and also for the unplanned distribution of electricity."}
```

Status: 200 OK Time: 6.67 s Size: 353 B

```
{  "sentiment": "Negative"}
```

Neutral sentiment:

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8000/analyze`. The request body is a JSON object with a `text` field containing a sentence about having lunch after work. The response status is 200 OK, and the response body is a JSON object with a `sentiment` field set to "Neutral".

```
POST http://127.0.0.1:8000/analyze
```

```
{  "text": "You may have lunch after finishing the work"}
```

Status: 200 OK Time: 6.67 s Size: 352 B

```
{  "sentiment": "Neutral"}
```

In my project, I have set a maximum limit of 500 characters for the input text. So the system gives an appropriate response if the text is above 500 characters.

The screenshot shows a REST client interface with a POST request to `localhost:8000/analyze`. The request body is a long string of text. The response status is `400 Bad Request` with a time of `14 ms` and a size of `401 B`. The response body is a JSON object:

```
1 {
2   "text": [
3     "Ensure this field has no more than 500 characters."
4   ]
5 }
```

It also provides a proper response if the input is not a valid text:

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8000/analyze`. The request body is a JSON object with an invalid text field. The response status is `400 Bad Request` with a time of `14 ms` and a size of `370 B`. The response body is a JSON object:

```
1 {
2   "text": [
3     "Not a valid string."
4   ]
5 }
```

Here are some examples using the browsable API in web browser:

Request:

Media type:

application/json

Content:

```
{  
  "text": "Patriotism can be defined as one's immense devotion to his/her country. A country is from its citizens. The love for own country is the feeling of patriotism. A person who has a true feeling of love and devotion towards his country is called a patriot."  
}
```

POST

Response:

← ↻ ⓘ localhost:8000/analyze

A ☆ 📄 ⌂ ⚙️ 🔍

Django REST framework

POST /analyze

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{  
  "sentiment": "Positive"  
}
```

Media type:

application/json

Content:

POST

Request:

Media type:	<input type="text" value="application/json"/>
Content:	<pre>{ "text": "Load-shedding means the discontinuation of the supply of electricity. Load-shedding occurs when the generation of power is less than the demand and also for the unplanned distribution of electricity. It creates problems of far-reaching consequences in the socio-economic development of a country. Houses, mills, factories, industries, shops, hospitals all fall a victim to it." }</pre>
<input type="button" value="POST"/>	

Response:

Api

POST /analyze

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{
  "sentiment": "Negative"
}
```