A02

May 9, 2025

[1]:
```
# Question:
# How do different Wikipedia articles compare in sentiment?
```

[2]:
```
import requests
import re
from bs4 import BeautifulSoup
from collections import Counter
```

[3]:
```
# Some common words of sentiment
very_positive_words = [
    'best', 'fantastic', 'excellent', 'amazing', 'awesome', 'perfect',␣
 ↪'outstanding',
    'superb', 'magnificent', 'wonderful', 'marvelous', 'extraordinary',␣
 ↪'fastest',
    'phenomenal', 'incredible', 'sensational', 'fabulous', 'astonishing',
    'breathtaking', 'spectacular', 'flawless', 'impressive', 'champion', 'loved'
]

positive_words = [
    'happy', 'good', 'joy', 'nice', 'pleasant', 'cheerful', 'highest',␣
 ↪'popular', 'first',
    'delightful', 'content', 'satisfied', 'positive', 'optimistic', 'fast',␣
 ↪'survive'
    'success', 'professional', 'encouraging', 'bright', 'enjoyable',␣
 ↪'wonderful', 'skill',
    'great', 'top', 'useful', 'renowned', 'favorable', 'agreeable', 'pleasing',
    'record', 'most', 'won', 'win', 'easy', 'creative'
]

negative_words = [
    'bad', 'sad', 'unhappy', 'poor', 'negative', 'unpleasant', 'last',␣
 ↪'scandal', 'detained',
    'disappointing', 'depressing', 'gloomy', 'unfortunate', 'upset',␣
 ↪'backlash', 'guilty',
    'miserable', 'dismal', 'terrible', 'dejected', 'discouraging', 'slow',␣
 ↪'convict', 'clumsy'
    'hopeless', 'regretful', 'painful', 'least', 'lost', 'lose', 'trouble',
```

```
    'bottom', 'lawsuit', 'lazy', 'damage', 'prison', 'jail', 'death', 'dead'
]

very_negative_words = [
    'worst', 'terrible', 'horrible', 'awful', 'dreadful', 'disastrous',␣
 ↪'horrific',
    'appalling', 'abysmal', 'shocking', 'atrocious', 'ghastly', 'hideous',
    'dreaded', 'dire', 'frightful', 'nightmarish', 'gruesome', 'heinous',
    'monstrous', 'vile', 'slowest', 'hated', 'abuse', 'assault', 'harassment'
]

# Function to fetch the content of a Wikipedia page
def get_wikipedia_content(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    paragraphs = soup.find_all('p')

# Cleans and combines data into a string and converts it to lowercase
    content = ' '.join([para.get_text() for para in paragraphs])
    return content.lower()

# Function to calculate the sentiment score
def sentiment_score(text):
    very_positive_count = sum(1 for word in very_positive_words if word in text)
    positive_count = sum(1 for word in positive_words if word in text)
    negative_count = sum(1 for word in negative_words if word in text)
    very_negative_count = sum(1 for word in very_negative_words if word in text)

# Calculates score with different weights
    score = (1.5 * very_positive_count + positive_count) - (1.5 *␣
 ↪very_negative_count + negative_count)
    return score

# Main function to analyze the sentiment of two Wikipedia pages
def analyze_sentiment(url1, url2):
    content1 = get_wikipedia_content(url1)
    content2 = get_wikipedia_content(url2)

    score1 = sentiment_score(content1)
    score2 = sentiment_score(content2)

    print(f"Sentiment score for URL 1: {score1}")
    print(f"Sentiment score for URL 2: {score2}")

# URLs of the Wikipedia pages
url1 = 'https://en.wikipedia.org/wiki/Al_Capone'
url2 = 'https://en.wikipedia.org/wiki/Abraham_Lincoln'
```

```
# Run the analysis
if __name__ == "__main__":
    analyze_sentiment(url1, url2)
```

```
Sentiment score for URL 1: -5.5
Sentiment score for URL 2: 4.0
```

[ ]:
```
# Conclusion
"""
Wikipedia pages of generally well-liked people such as Abraham Lincoln and␣
 ↪Lionel are
usually more optimistic and tend to get more positive scores than wikipedia␣
 ↪pages that focus
on criminals such as Al Capone or Sean Diddy who both get negative scores.
"""
```