

```

# Import libraries
import requests
from transformers import pipeline

# Read API key from file (ensure API_KEY file exists with your key)
API_KEY = open('API_KEY').read().strip() # Remove trailing whitespace

# Define search parameters
keyword = 'gold'
date = '2024-09-23'

# Create a pipeline for sentiment analysis using ProsusAI/finbert model
pipe = pipeline("text-classification", model="ProsusAI/finbert")

# Build the News API URL with keyword, date, sorting, and API key
url = (
    'https://newsapi.org/v2/everything?'
    f'q={keyword}&'
    f'from={date}&'
    'sortBy=popularity&'
    f'apiKey={API_KEY}'
)

# Send a GET request to the News API and get the response
response = requests.get(url)

# Check for successful response
if response.status_code == 200:
    # Parse the JSON response and extract articles
    articles = response.json()['articles']
    # Filter articles containing the keyword in title or description (case-insensitive)
    articles = [article for article in articles if
        keyword.lower() in article['title'].lower() or keyword.lower() in article['description'].lower()]
    # Initialize variables for sentiment analysis
    total_score = 0
    num_articles = 0
    # Loop through each article
    for i, article in enumerate(articles):
        print(f'Title:{article["title"]}')
        print(f'Link:{article["url"]}')
        print(f'Description:{article["description"]}')
        # Use pipeline to analyze sentiment of the article content
        sentiment = pipe(article['content'])[0]
        print(f'Sentiment: {sentiment["label"]}, Score: {sentiment["score"]}')
        print('-' * 40)
        # Update sentiment score based on sentiment label (positive/negative)
        if sentiment['label'] == 'positive':
            total_score += sentiment['score']
        elif sentiment['label'] == 'negative':
            total_score -= sentiment['score']
        num_articles += 1

    # Calculate overall sentiment score
    final_score = total_score / num_articles
    # Print overall sentiment based on score threshold
    print(
        f'Overall Sentiment: {"Positive" if final_score >= 0.15 else "Negative" if final_score <= -0.15 else "Neutral"}{final_score}')
else:
    print(f"Error: News API request failed with status code {response.status_code}")

```