

TASK

Create a feedback and reprocessing hub with real-time monitoring of narrative volatility, triggering analysis when contradiction or manipulation signals emerge.

PROGRAMS/TOOLS USED (IN ORDER OF USAGE)

Google

Google AI

Claude.ai

Notepad ++

Google.Colab

STEP-BY-STEP BREAKDOWN

- Searched Google and Google AI extensively to understand the assignment. Essential terminology for this project is called “Web scraping”.
- Used ChatGPT to generate first variant of code. This code worked, but was drawing information from a simulated database. Small code. Generated simple alerts popping up in the window.
- Used Claude.ai to generate second variant of code. This code worked, but was drawing information from a simulated database. Very large code.
- Tried running the python code in Notepad++, no success.
- Googled a program/tool to run python code, found Google.colab.
- Ran the Claude.ai code in Google.Colab. The program worked, but was drawing information from the simulated data base. Code roughly 675 lines.
- Asked Claude.ai to draw information from a live source. Claude.ai failed to deliver stating that the “size of the message was too large”. It could not generate a code that long.
- Returned to ChatGPT and asked it to generate a code for me, but using a live source. Code generated, but encountered a persistent error. Run ModuleNotFoundError: No module named ‘gnews’. (Figure 1)
- Did research on Google Colab and ChatGPT, and eventually ran the command “!pip install GoogleNews” at the very start of the python program, as suggested by Google Colab. (Figure 2)
- Ran in Google Colab and the program worked perfectly, pulling information from GoogleNews. (Figure 3)
- Reported to Rob and demonstrated the result.

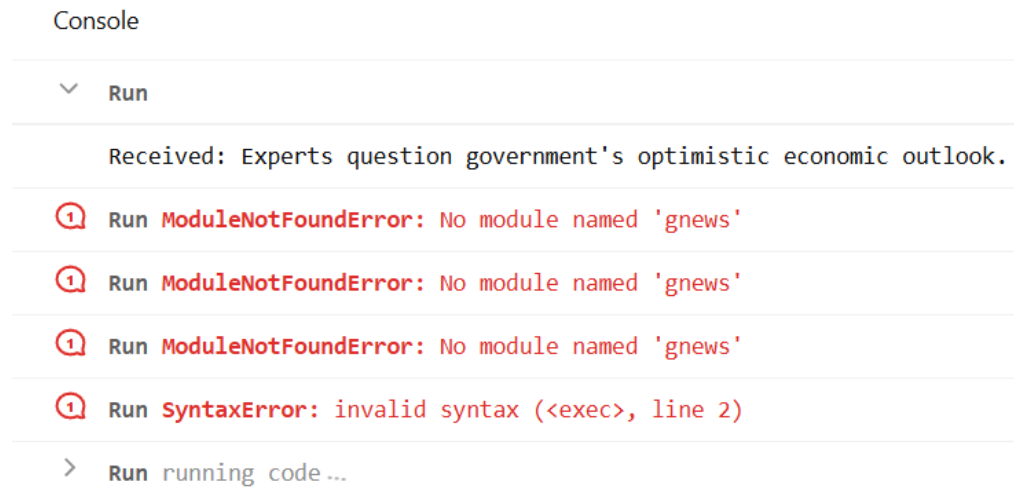


Figure 1. Error in ChatGPT

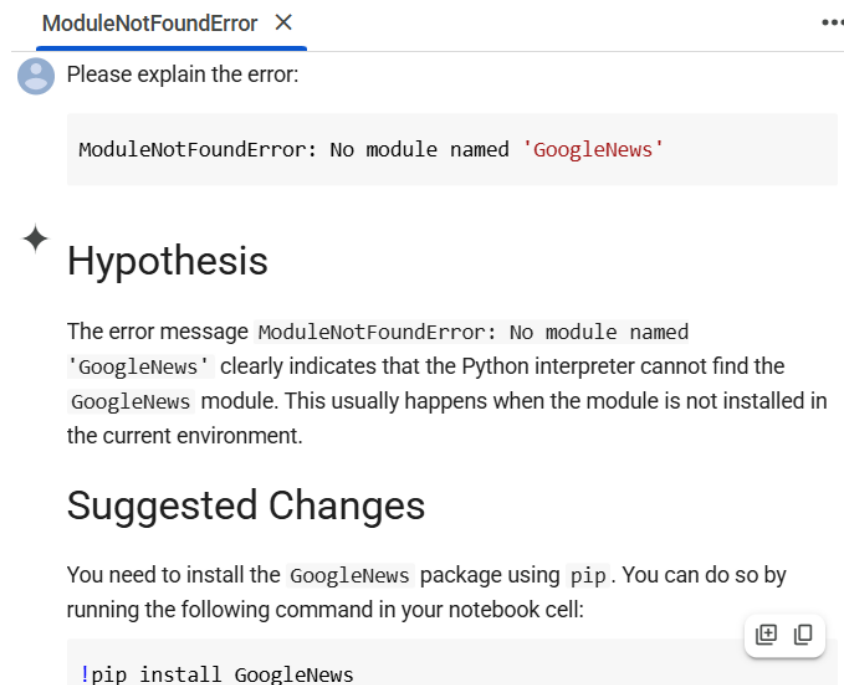


Figure 2. Explanation and what to do. Google.Colab

```
time.sleep(REFRESH_INTERVAL)

Collecting GoogleNews
  Downloading GoogleNews-1.6.15-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from GoogleNews) (4
Collecting dateparser (from GoogleNews)
  Downloading dateparser-1.2.1-py3-none-any.whl.metadata (29 kB)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.11/dist-packages (from GoogleNews) (:
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4-)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beau
Requirement already satisfied: pytz>=2024.2 in /usr/local/lib/python3.11/dist-packages (from dateparser->Goog
Requirement already satisfied: regex!=2019.02.19,!=2021.8.27,>=2015.06.24 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: tzlocal>=0.2 in /usr/local/lib/python3.11/dist-packages (from dateparser->Goog
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil->Goog
Downloading GoogleNews-1.6.15-py3-none-any.whl (8.8 kB)
Downloading dateparser-1.2.1-py3-none-any.whl (295 kB)
295.7/295.7 kB 6.6 MB/s eta 0:00:00
Installing collected packages: dateparser, GoogleNews
Successfully installed GoogleNews-1.6.15 dateparser-1.2.1
Received: Coming Soon: World Economic Outlook, April 2025 - The World Economic Outlook (WE...
Received: CNBC Daily Open: Consumers are worried over the economy. The White House insists...
Received: Opinion | Voters are not happy with Trump's trade war - Leon Sit and Max McCall ...
Received: China's Economy Likely Grew 5.2% in Months Before Big US Tariffs - China will li...
Received: What soaring uncertainty means for the U.S. economy - As experts and business le...
Received: Trump's Polling Problem: Voters No Longer Blame Economy on Biden - Half of all v...
Received: Confusion over electronics tariffs injects more uncertainty into economy - The e...
Received: Trump's China tariffs are slamming small businesses - the heart of our economy a...
Received: Confusion over electronics tariffs injects more uncertainty into economy - The e...
Received: US Treasury Secretary Bessent leaves Economy Ministry to meet Argentina's Milei ...
Received: Coming Soon: World Economic Outlook, April 2025 - The World Economic Outlook (WE...
Received: CNBC Daily Open: Consumers are worried over the economy. The White House insists...
```

Figure 3. Output. Google.Colab

FUNCTIONING CODE, COPIED FROM GOOGLE.COLAB. GENERATED BY RUNNING THROUGH CLAUDE.AI AND CHATGPT.

```
!pip install GoogleNews

import time

from GoogleNews import GoogleNews

from typing import List, Dict

# === CONFIGURATION ===

SEARCH_TERM = "economy"

REFRESH_INTERVAL = 30 # seconds

# === CONTRADICTION + MANIPULATION DETECTION ===

def detect_signals(new_item: Dict, memory: List[Dict]) -> Dict:

    contradiction_phrases = ["rise in unemployment", "question", "recession", "layoffs"]
```

```

manipulation_signals = ["sponsored content", "viral", "fake", "conspiracy"]
positive_phrases = ["economy is improving", "record highs", "strong growth", "booming"]

contradiction = False
manipulation = False
text = new_item["text"].lower()

for memory_item in memory:
    prev_text = memory_item["text"].lower()
    if any(c in text for c in contradiction_phrases) and any(p in prev_text for p in
positive_phrases):
        contradiction = True

if any(m in text for m in manipulation_signals):
    manipulation = True

return {"contradiction": contradiction, "manipulation": manipulation}

# === FEEDBACK & REPROCESSING HUB ===

class FeedbackReprocessingHub:
    def __init__(self):
        self.memory = []

    def process_item(self, item: Dict):
        print(f"Received: {item['text'][:80]}...")
        signals = detect_signals(item, self.memory)

        if signals["contradiction"] or signals["manipulation"]:
            self.trigger_analysis(item, signals)

        self.memory.append(item)

```

```

def trigger_analysis(self, item: Dict, signals: Dict):

    print(f"[ALERT] Signal detected in: '{item['text']}' from {item['source']}")

    if signals["contradiction"]:

        print("\t→ Contradiction with existing narrative.")

    if signals["manipulation"]:

        print("\t→ Potential manipulation markers detected.")

    # Future: Launch LLM, fact-checker, or human-in-the-loop process

```

```

# === GOOGLE NEWS FETCHER ===

```

```

def fetch_google_news() -> List[Dict]:

    googlenews = GoogleNews(lang='en', region='US')

    googlenews.clear()

    googlenews.search(SEARCH_TERM)

    articles = googlenews.results(sort=True)

    results = []

    for article in articles:

        results.append({

            "id": article.get("link", ""),

            "text": article.get("title", "") + " - " + article.get("desc", ""),

            "source": article.get("media", "Google News")

        })

    return results

```

```

# === MAIN LOOP ===

```

```

if __name__ == "__main__":

    hub = FeedbackReprocessingHub()

    seen_ids = set()

```

```
while True:

    articles = fetch_google_news()

    for article in articles:

        if article["id"] not in seen_ids:

            hub.process_item(article)

            seen_ids.add(article["id"])

    time.sleep(REFRESH_INTERVAL)
```