# **Global News Topic Tracker**

We're drowning in news these days. Headlines everywhere—tech, sports, the world melting down (again), you name it. Honestly, who's got the time to scroll through all that mess? Not me. That’s why this Streamlit app exists: you open it, pick your favorite news flavor, and boom—it grabs trending headlines, lumps them by topic, and spits out bite-sized summaries. Skim, scroll, done.

## **Problem Statement**

In the age of information overload, it's hard to:

* Hard to spot what's trending (unless you love doomscrolling).
* Headlines are all over the place—finding the real story is a pain.
* Nobody wants to read 30 versions of “AI is taking your job.”

## **Goal**

Create an app that **automatically fetches, groups, and summarizes** trending news headlines — saving time and improving clarity. Ie, Auto-fetch, auto-group, auto-summarize. You get the gist in seconds, not hours.

## **Solution Summary**

We built a modular, AI-powered dashboard that:

* Grabs the latest news off Google News (not stealing, just borrowing)
* Bundles headlines together using sentence embeddings + KMeans
* Feeds each group into a Hugging Face BART model for quick summaries
* Shows it all off in a slick Streamlit dashboard—category pickers and all

## 

## 

## 

## **Solution Architecture**

+----------------------+

| User Interface | ← Streamlit

| (Category Selectbox)|

+----------+-----------+

|

v

+----------------------+

| News Fetcher | ← news\_scraper.py

| (Google News / API) |

+----------+-----------+

|

v

+----------------------+

| Topic Grouper | ← topic\_grouper.py

| (Sentence Embeddings |

| + KMeans Clustering)|

+----------+-----------+

|

v

+----------------------+

| Summarizer (LLM) | ← summarizer.py

| (Hugging Face Model) |

+----------+-----------+

|

v

+----------------------+

| Streamlit Output | ← frontend/news-app.py

| (Clustered, summarized) |

+----------------------+

## **Features**

* Pick-your-poison news categories (World, Sports, etc.)
* Live scraping—fresh news, not yesterday’s leftovers
* Smart grouping so you see big stories, not just random noise
* Quick summaries, so you don’t need to read the full article (ain’t nobody got time for that)
* Super clean, interactive UI

## **Project Structure**

global-news-tracker/

├── backend/

| ├── main.py

│ ├── news\_scraper.py

│ ├── summarizer.py

│ ├── topic\_grouper.py

├── frontend/

│ └── news-app.py

├── requirements.txt

## **How It Works**

1. User selects a category using a **select box**:  
     
   category = st.selectbox(

"Choose News Category",

["World", "Technology", "Sports", "Business", "Health", "Entertainment"]

)

1. The app fetches **top 10 headlines** from Google News
2. Headlines are grouped using **sentence-transformers + KMeans**
3. Each group is summarized using Hugging Face’s **BART model**
4. Results are displayed clearly in a list

## **Setup Instructions**

### **1. Create a Virtual Environment**

python -m venv venv

.\venv\Scripts\activate # On Windows

### **2. Install Dependencies**

pip install -r requirements.txt

### **3. Run the App**

streamlit run frontend/news-app.py

## **Sample Output**

Output based on selected category:

"Google launches Gemini 2"

→ Summary: Google unveiled its next-gen AI model to compete with OpenAI.

"Meta delays release of Llama-3"

→ Summary: Meta confirmed Llama-3 will be postponed due to training issues.

"Apple unveils M4 chip"

→ Summary: Apple introduces M4 chip with better AI performance.

"Qualcomm partners with TSMC"

→ Summary: Qualcomm collaborates with TSMC for advanced AI hardware.

Each headline comes with a short AI-generated summary.

## **Testing (Optional)**

You can test summarization manually:

from transformers import pipeline

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

print(summarizer("This is a long sentence that needs summarization.", max\_length=20, min\_length=5))

## **Requirements**

*streamlit*

*beautifulsoup4*

*requests*

*transformers*

*sentence-transformers*

*scikit-learn*

*torch*

## 

## **Technologies Used**

| **Technology** | **Purpose** |
| --- | --- |
| Streamlit | Web UI |
| BeautifulSoup | Google News scraping |
| Hugging Face (BART) | Text summarization |
| sentence-transformers | Headline embeddings |
| scikit-learn | KMeans clustering |
| FastAPI (optional) | REST API backend |

## **Future Enhancements**

* Add some sentiment analysis (find out if the world’s actually ending)
* Export those summaries as PDFs
* Multi-language support (not everyone loves English, shocking, I know)
* Let folks search for custom stuff, not just click categories

## **License**

**MIT License** — Free to use and adapt.