Tech Trends News Agent – Project Report

Divya Thoppae and Sriganesh Srinivasan

Abstract

The problem we addressed is information overload in technology news: millions of articles are published globally each year, making it nearly impossible for individuals to track developments manually. Our approach was to design an AI agent that retrieves articles from NewsAPI, preprocesses and tokenizes them, enables semantic search using TF-IDF, extracts trending topics through co-occurrence clustering, and provides an interactive chat interface powered by a local LLM. We used the Qwen2.5-1.5B-Instruct model for response generation, custom TF-IDF search for retrieval, and dynamic topic extraction with sub-trend analysis. Key outcomes include a conversational Q&A interface, automatic detection of topic clusters from article content, and visualizations showing topic distribution and sub-trends. The project demonstrates how AI can reduce cognitive load and improve access to timely, structured information.

Overview

What is the problem?

We are trying to solve the challenge of information overload in the technology domain. Professionals, students, and businesses face fragmented coverage, repetitive reading, and difficulty detecting emerging themes. The motivation was to reduce the cognitive burden of staying informed and to provide a system that can automatically highlight what matters most.

Why is this problem interesting?

This problem is interesting because it connects directly to how society keeps pace with innovation. Technology evolves rapidly, with advances in AI, cybersecurity, and cloud computing happening weekly. Missing updates can mean falling behind in knowledge, research, or business strategy. Use cases include study assistants, industry monitoring tools, and research dashboards that help communities stay informed without being overwhelmed.

What is the approach?

We proposed an agentic workflow that integrates article retrieval, NLP preprocessing, TF-IDF-based semantic search, LLM-powered answer generation, and dynamic topic visualization. This approach makes sense because it enforces structure, enables natural language querying, and surfaces recurring themes automatically. Alternative approaches such as manual reading or keyword-only search would not scale or provide the conversational, contextual responses our system offers.

Rationale

Our design builds on established NLP techniques (TF-IDF, tokenization, lemmatization) combined with modern instruction-tuned LLMs for natural response generation. Unlike simple summarization tools, our agent emphasizes interactive querying, source attribution, and automatic topic discovery through statistical co-occurrence analysis.

Key components and results

- **Retrieval** via NewsAPI (technology headlines endpoint)
- **Preprocessing** using NLTK (tokenization, stopword removal, lemmatization)
- **Search** using custom TF-IDF with cosine similarity
- **Answer generation** using Qwen2.5-1.5B-Instruct model
- **Topic extraction** with TF-IDF weighting and co-occurrence clustering
- **Visualization** with Streamlit bar charts and progress indicators

Results: A functional interactive agent with chat interface and topic visualization.

Limitations: Dependent on NewsAPI availability and rate limits; LLM response quality varies; topic extraction works best with larger article sets.

Approach

Methodology

Iterative workflow: User query → TF-IDF search over preprocessed corpus → Retrieve top-k articles → Generate contextual answer with LLM → Display with source citations. Parallel workflow: Corpus → Extract top terms via TF-IDF → Cluster by co-occurrence → Identify sub-trends → Visualize.

Algorithms/Models

- **Qwen2.5-1.5B-Instruct:** Instruction-tuned causal language model for answer generation (runs on GPU with float16 precision)
- **TF-IDF with cosine similarity:** For semantic document retrieval; title tokens weighted 3x for relevance

- **NLTK pipeline:** word_tokenize, stopword filtering, WordNet lemmatization
- **Co-occurrence clustering:** Custom algorithm that groups frequently co-occurring terms into topic clusters
- **Sub-trend extraction:** Secondary term frequency analysis within topic document subsets

Design choices

- **Modular architecture:** Separate Python modules for fetching, preprocessing, searching, LLM interface, and app
- **GPU acceleration:** Automatic detection and use of CUDA when available (Tesla T4 in Colab)
- **Caching:** Streamlit's `@st.cache_resource` for LLM and `@st.cache_data` for visualization data
- **Interactive UI:** Two-tab interface separating chat functionality from topic visualization

Limitations

- NewsAPI free tier limits article volume and historical access
- Qwen2.5-1.5B is lightweight; larger models would improve response quality
- Topic extraction requires minimum article counts for meaningful clusters
- No persistent storage between sessions

Experiments

Dataset

Live technology headlines from NewsAPI. During testing, approximately 68–100 articles collected per fetch. Articles sourced from major technology publications. Basic statistics: headlines and descriptions processed; content truncated by API.

Implementation

fetch_news.py:

- Uses NewsAPI client to fetch technology category headlines
- Paginates through results up to max_articles limit
- Saves raw JSON to `data/raw/` with timestamp

preprocess.py:

- Loads raw articles and applies NLTK preprocessing

- Tokenizes, removes stopwords, lemmatizes
- Outputs processed corpus to `data/processed/`

search_articles.py:

- Implements TF-IDF vectorization from scratch
- Computes document frequency, inverse document frequency, and cosine similarity
- Topic extraction using term co-occurrence matrices
- Sub-trend discovery within topic clusters

llm_interface.py:

- Loads Qwen2.5-1.5B-Instruct with Hugging Face Transformers
- Configures for GPU (float16) or CPU (float32)
- Generation config: max_new_tokens=256, temperature=0.7, do_sample=True

app.py (Streamlit):

- Sidebar for data fetching and configuration
- Chat tab with conversation history and source citations
- Topic visualization tab with bar charts and expandable deep dives

Parameters

- Summary/answer length: ~200 new tokens
- Top-k search results: 5
- Topic clusters: 5
- Sub-trends per topic: 5
- TF-IDF document frequency bounds: 2%–70% of corpus

Environment

Python 3.x, Google Colab with Tesla T4 GPU, PyTorch, Transformers, NLTK, Streamlit, ngrok for public URL tunneling.

Results

Main results

- **Chat functionality:** Users can ask natural language questions about technology news; the system retrieves relevant articles and generates contextual answers with source attribution

- **Topic discovery:** Automatic identification of 5 major topic clusters from article corpus with associated key terms
- **Sub-trend analysis:** Each topic includes 5 sub-trends with relevance scores
- **Performance:** Model loads in ~60 seconds on Tesla T4; inference is near-instantaneous for queries

Supplementary results

- Title weighting (3x) in TF-IDF significantly improved retrieval relevance
- Co-occurrence threshold of 2+ documents prevented spurious topic associations
- Document frequency bounds (2%–70%) filtered out both rare and ubiquitous terms
- Temperature of 0.7 balanced response creativity with factual grounding

Discussion

Our results show that combining traditional IR techniques (TF-IDF) with modern LLMs creates an effective news analysis system. The chat interface provides natural interaction, while topic visualization offers at-a-glance trend awareness.

Strengths:

- End-to-end pipeline from data collection to interactive UI
- No external API calls for inference (runs locally on GPU)
- Modular design allows easy component replacement
- Source attribution maintains transparency

Weaknesses:

- NewsAPI free tier limitations constrain article volume
- Qwen2.5-1.5B occasionally produces generic or repetitive responses
- Topic extraction quality depends on corpus size
- No fine-tuning on news domain

Comparison: Unlike cloud-based summarization services, our system runs entirely locally after data fetch, preserving privacy and reducing latency. Unlike keyword search, our LLM integration provides natural language answers.

Future directions:

- Upgrade to larger LLM (e.g., Qwen2.5-7B) for better responses
- Add RSS feed support for broader source coverage
- Implement article deduplication

- Fine-tune on technology news corpus
- Add temporal trend analysis (how topics evolve over time)

Conclusion

We built the Tech Trends News Agent to solve the problem of information overload in technology news. By combining NewsAPI retrieval, NLTK preprocessing, TF-IDF search, Qwen2.5 LLM generation, and dynamic topic extraction, we created an interactive system that helps users query news naturally and discover emerging themes automatically. The Streamlit interface provides both conversational access and visual trend analysis. While limitations exist in article volume and model size, the agent demonstrates a practical approach to building AI-powered information tools that run efficiently on consumer GPU hardware.

References

1. Qwen2.5 Technical Report, Alibaba Cloud (2024)
2. Hugging Face Transformers library
3. NewsAPI documentation
4. NLTK: Natural Language Toolkit (Bird et al.)
5. Streamlit documentation
6. CS 4100 Fall 2025 Course Project guidelines