HACKATHON PROBLEM STATEMENT

Track: AI/ML & Financial Technology

Powered by **Tradl**

AI-Powered Financial News Intelligence System

Challenge Overview

Build an intelligent multi-agent system that processes real-time financial news, eliminates redundancy, extracts market entities, and provides context-aware query responses for traders and investors.

1. Background

Financial markets generate thousands of news articles daily from multiple sources—regulatory filings, business media, exchange announcements, and analyst reports. All through scrapping periodically. Professional traders need systems that can:

- Identify unique news stories from redundant coverage across sources
- Extract market entities (companies, sectors, regulators) automatically
- Map news events to impacted stocks with confidence levels
- Retrieve relevant news based on context-aware natural language queries

The Challenge: Build a system that understands semantic similarity between articles, entity relationships, and query intent to deliver precise, actionable financial intelligence.

2. Problem Statement

Objective: Design and implement a multi-agent AI system using LangGraph that processes financial news, identifies unique stories, maps stock impacts, and enables intelligent querying.

Core Capabilities Required

A. Intelligent Deduplication

Expected Behavior: Multiple articles covering the same event should be identified as duplicates and consolidated into a single unique story. The system must handle semantic similarity—articles with different wording but identical meaning should be recognized as duplicates.

Technical Hint: Consider using RAG-based approaches with vector embeddings for semantic similarity detection. Target: ≥95% accuracy on duplicate detection.

```
Example:

Input Article 1: "RBI increases repo rate by 25 basis points to combat
inflation"

Input Article 2: "Reserve Bank hikes interest rates by 0.25% in
surprise move"

Input Article 3: "Central bank raises policy rate 25bps, signals
hawkish stance"

Output: Single consolidated story (all three identified as duplicates)
```

B. Entity Extraction & Impact Mapping

Expected Behavior: Extract structured entities (Companies, Sectors, Regulators, People, Events) from each news article and map to impacted stocks with confidence scores.

Technical Hint: Use NER models for entity recognition. Map entities to stocks with confidence levels: direct mention (100%), sector-wide impact (60-80%), regulatory impact (variable). Target: ≥90% entity extraction precision.

```
Example:

Input: "HDFC Bank announces 15% dividend, board approves stock
buyback"

Output:

  Companies: [HDFC Bank]

  Sectors: [Banking, Financial Services]

  Impacted Stocks: [{symbol: HDFCBANK, confidence: 1.0, type: direct}]
```

C. Context-Aware Query System

Expected Behavior: Natural language queries must retrieve relevant news with intelligent context expansion. When a user queries a company, return both direct mentions AND sector-wide news. When querying a sector, return all related news across companies.

Technical Hint: Implement entity recognition on queries, semantic search capabilities, and hierarchical relationship understanding (company → sector, regulation → industry).

Query Behavior Specification

Your system must handle these query patterns correctly:

Query	Expected Results	Reasoning
"HDFC Bank news"	N1, N2, N4	Direct mentions + Sector-wide banking news
"Banking sector update"	N1, N2, N3, N4	All sector-tagged news across banks
"RBI policy changes"	N2 only	Regulator-specific filter
"Interest rate impact"	N2, related articles	Semantic theme matching

Sample News Dataset (Reference):

```
N1: HDFC Bank announces 15% dividend, board approves stock buyback
N2: RBI raises repo rate by 25bps to 6.75%, citing inflation concerns
N3: ICICI Bank opens 500 new branches across Tier-2 cities
N4: Banking sector NPAs decline to 5-year low, credit growth at 16%
```

3. Technical Stack

Component	Technology
Agent Framework	LangGraph (required)
LLM	Claude/GPT-4/Llama
Embeddings	sentence-transformers (or equivalent)
Vector Database	ChromaDB/Pinecone/FAISS (for RAG)
Structured DB	PostgreSQL/SQLite
NER	spaCy/Stanza
API Framework	FastAPI/Flask

4. Evaluation Criteria

Category	Weight	Focus Areas
Functional Correctness	**40%**	Deduplication accuracy, entity precision, query relevance, impact mapping
Technical Implementation	**30%**	LangGraph design, RAG effectiveness, code quality
Innovation & Completeness	**20%**	Novel approaches, feature completeness, bonus challenges
Documentation & Demo	**10%**	Code clarity, docs quality, demo effectiveness

5. Deliverables

Submit a complete package including:

Code Repository

- Ingestion service to continuously poll data
- LangGraph multi-agent implementation
- Mock news dataset (minimum 30 diverse articles)
- API endpoints for querying the system

6. Demo

- **Live demo (CLI/web interface)** showing all core capabilities
- **5-10 minute video walkthrough** covering key features

7. Bonus Challenges

Stand out by implementing advanced features:

- **Sentiment Analysis:** Predict price impact using historical sentiment-return patterns
- **Real-time Alerts:** WebSocket notifications for breaking news

8. Submission Guidelines

Deadline: [To be announced by organizers]

1. **GitHub Repository:** Public repo with complete code and documentation
2. **Demo Video:** 5-10 min walkthrough (YouTube/Vimeo/Loom)
3. **Presentation:** PDF deck (max 10 slides)
4. **Submission Form:** [Link provided by organizers]

Data Sources

- NSE India: https://www.nseindia.com/
- BSE India: https://www.bseindia.com/
- RBI: https://www.rbi.org.in/
- RSS feeds of news channels

Questions or Clarifications?

Contact the organizing team or post in the official hackathon channel. We encourage creative approaches and innovative solutions!

Good luck, and happy hacking!