Sentinel: Probabilistic Semantic Memory for Disaster Response

Submitted By: M B Thejesshwar

Hackathon: Qdrant Convolve 4.0

Theme: Disaster Response & Public Safety

1. Problem Statement

1.1 The Challenge: Information Overload in Crisis

In the immediate aftermath of natural disasters such as wildfires, floods or earthquakes, first responders are often overwhelmed by information fragmentation. Critical data streams in from diverse sources such as drone video feeds, emergency radio logs, satellite imagery and text reports but these streams remain siloed.

Current autonomous systems typically lack **long term semantic memory**. They process data frame by frame but fail to retain context over time. For example, a drone might spot a fire in Sector A but ten minutes later, it has no memory of that hazard to correlate with a new radio report about smoke in Sector A. This lack of temporal reasoning leads to redundant searches and delayed response times.

1.2 Societal Relevance

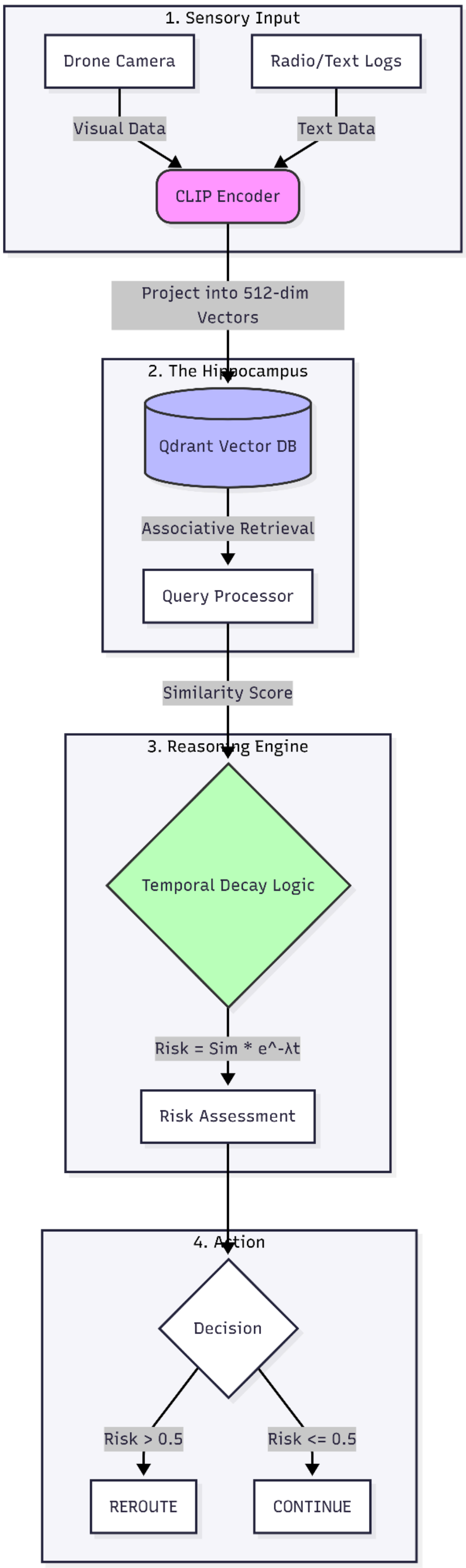
This project addresses the **Disaster Response & Public Safety** challenge. By enabling autonomous agents to remember and reason over multimodal data, Sentinel aims to:

- **Reduce Cognitive Load:** Automate the correlation of visual hazards with text reports, freeing up human operators to focus on strategy.
- **Improve Safety:** Allow drones to dynamically reroute away from evolving hazards without waiting for human commands.
- **Accelerate Rescue:** Prioritize search areas based on fresh, evidence based risk assessments rather than outdated intelligence.

2. System Design

2.1 Architecture Overview

Sentinel is an autonomous agent architecture that utilizes **Qdrant** not just as a database but as a Hippocampus, a short term semantic memory system that integrates sensory inputs and governs decision making.



- **Input:** Drone Camera and Radio Logs
- **Encoder:** Both feed into a box labeled OpenAI CLIP.
- **Memory:** The Encoder points to a central cylinder labeled Qdrant.
- **Reasoning:** A Logic Engine pulls data from Qdrant, applies a Time Decay formula and outputs a Navigation Decision.

2.2 Why Qdrant is Critical

Qdrant is the core engine of Sentinel. Traditional databases cannot efficiently link unstructured data like an image of a flood with a query like houses submerged in water. Qdrant enables this **Multimodal Associative Retrieval** through high performance vector similarity search. Furthermore, Qdrant's ability to handle complex **payloads** is essential for our temporal reasoning logic.

3. Multimodal Strategy

3.1 Data Fusion

Sentinel operates on the principle that Risk is a multimodal concept. A hazard can be *seen* or *reported*. To bridge this gap, we utilize **OpenAI CLIP**.

- **Visual Data:** Simulated drone imagery is encoded into 512 dimensional vectors using the **ViT-B/32** visual transformer.
- **Textual Data:** Emergency queries are encoded into the same vector space using the CLIP text encoder.

3.2 The Common Vector Space

By projecting both data types into the same latent space, Sentinel can perform cross modal queries. When the Reasoning Engine asks, *Is there a fire?*Qdrant retrieves the closest matching *visual memories* based on semantic proximity, without requiring manual image tagging.

4. Search, Memory & Reasoning Logic

4.1 Associative Retrieval

The retrieval process is triggered when the agent scans a sector. It generates a query vector from the sector's description and performs a **Cosine Similarity Search** in Qdrant.

- **Query: client.query_points(query=text_vector, limit=1)**
- **Result:** The system identifies the specific image in memory that correlates with the current textual context.

4.2 Evolving Representations: Time Decay Logic

A key requirement of this challenge was Memory Beyond a Single Prompt and Evolving Representations. A static database of hazards is dangerous because risks change over time like a fire spreads but a flood might recede.

Sentinel implements a **Probabilistic Decay Function** to model this evolution. The Risk Score of a retrieved memory is not constant; it decays as the memory ages.

The Formula:

$$Risk(t) = R_{initial} \times e^{-\lambda t}$$

Where:

- $R_{initial}$ is the base risk score of the hazard.
- t is the elapsed time since the memory was formed.
- λ is the decay constant.

Implementation: For this hackathon demonstration, a rapid half-life was utilized to showcase the effect within a short testing window. In a real world deployment, this would be tuned to hours or days. This logic ensures that the agent prioritizes **fresh evidence** over stale data, effectively forgetting irrelevant hazards.

4.3 Decision Matrix

The agent makes decisions based on the *Decayed Risk Score*, not the raw similarity.

- **IF Risk > 0.5: DECISION: DANGER DETECTED**. The agent autonomously reroutes.
- **IF Risk ≤ 0.5: DECISION: AREA SECURE**. The agent proceeds, treating the old hazard as resolved or irrelevant.

5. Evidence of Performance

5.1 Experimental Setup

We simulated a mission timeline where the drone encounters three scenarios at different times:

1. **Sector Alpha:** A safe park.
2. **Sector Beta:** An active fire.
3. **Sector Gamma:** A flood.

5.2 Interaction Logs

Below is the actual output from the Sentinel Reasoning Engine, demonstrating the system's ability to differentiate between active and stale threats.

```
Context: 'a peaceful grassy path with trees'
EVIDENCE: Matched 'safe' image
(Source: safe.jpg)
(Time Decay: 1218.452318429947s elapsed)
(Similarity: 0.32180042688411714)
DECISION: AREA SECURE (Risk: 7.705870940840538e-08)
ACTION: Continue Search & Rescue path.
Context: 'huge raging fire and smoke'
EVIDENCE: Matched 'fire' image
(Source: fire.jpg)
(Time Decay: 18.759700298309326s elapsed)
(Similarity: 0.2634124045194446)
DECISION: DANGER DETECTED (Risk: 0.7246394401078661)
ACTION: Reroute Autonomous Agent immediately.
Context: 'houses submerged in deep water'
EVIDENCE: Matched 'flood' image
(Source: flood.jpg)
(Time Decay: 618.7156193256378s elapsed)
(Similarity: 0.2958067871465967)
DECISION: AREA SECURE (Risk: 0.0006293477013115754)
ACTION: Continue Search & Rescue path.
```

- **Analysis of Sector Beta :** The system retrieved the **fire.jpg** image. Because the timestamp was recent (~19s elapsed), the Risk Score remained high (0.72), triggering a **DANGER** alert.
- **Analysis of Sector Gamma :** The system retrieved the **flood.jpg** image. However, because the memory was stale (619s elapsed), the Risk Score decayed to 0.0006. The agent correctly decided the area was **SECURE**, avoiding a false positive reroute.

6. Limitations & Ethics

6.1 Limitations

- **Model Bias:** The current system relies on pre-trained CLIP weights. While effective for general hazards, it may struggle with domain specific nuances without fine tuning.
- **Context Window:** The current implementation retrieves the *single* top match (**limit=1**). A more robust system would aggregate risk across multiple retrieved vectors to reduce noise.

6.2 Ethical Considerations

- **Human in the Loop:** While Sentinel is designed for autonomous *reasoning*, decision authority regarding life saving operations must remain with human commanders. This system is a decision support tool not a replacement for human judgment.
- **Privacy:** In real world deployments, storing drone imagery of private residences in a vector database raises privacy concerns. Future iterations would utilize **Qdrant's Payload Filtering** to automatically redact or flag sensitive regions before ingestion.

7. Conclusion

Sentinel demonstrates that **Qdrant** is more than a search engine, it is a viable foundation for **Artificial Episodic Memory**. By combining multimodal embeddings with temporal reasoning logic, a system that addresses the critical societal challenge of disaster response, transforming raw data into actionable, life saving intelligence is built.