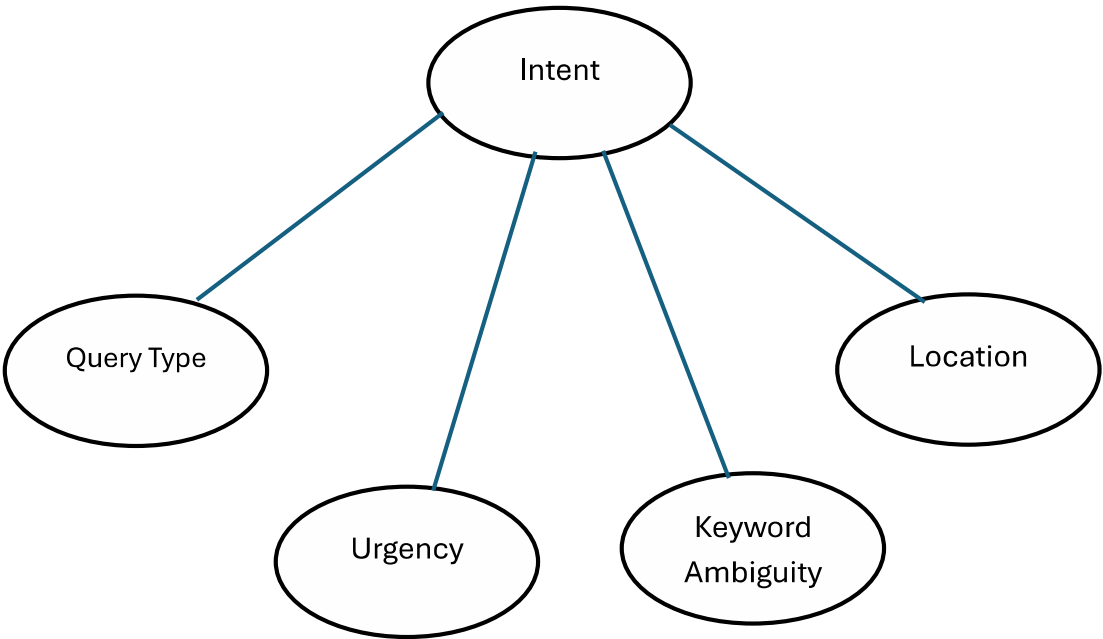


AI Theory Project

Group 5

Sayan Dey 25AI06009
Arindam Das 25AI06003
Sagnik Kayal 25AI06021
Soutrik Das 25AI06011
Deepta Kiran Das 25AI06005
Fenil Shah 25CS06020
Vunnam Vivekanand Chowdary 21CS02002
Avik Sarkar A25CS09008

Bayesian Network Diagram



Conditional Probability Tables

P(Intent)

Intent	P(Intent)
WaterSafety	0.25
Compensation	0.25
PestControl	0.25
Health	0.25

P(Query Type | Intent)

Intent	Water	Compensation	Agriculture	Health	Other
WaterSafety	0.80	0.02	0.03	0.03	0.12
Compensation	0.02	0.85	0.03	0.03	0.07
PestControl	0.01	0.02	0.85	0.01	0.11
Health	0.02	0.03	0.02	0.85	0.08

P(Keyword Ambiguity | Intent)

Intent	Low	High
WaterSafety	0.80	0.20
Compensation	0.70	0.30
PestControl	0.60	0.40
Health	0.65	0.35

P(Urgency | Intent)

Intent	High	Low
WaterSafety	0.80	0.20
Compensation	0.60	0.40
PestControl	0.55	0.45
Health	0.70	0.30

P(Location | Intent)

Intent	Flooded	Rural	Urban	Nationwide
WaterSafety	0.50	0.20	0.20	0.10
Compensation	0.10	0.30	0.20	0.40
PestControl	0.05	0.65	0.15	0.15
Health	0.20	0.25	0.35	0.20

Posterior Computation

For a query with observed evidence (E):

$$P(\text{Intent} | E) \propto P(\text{Intent}) \times P(\text{QueryType}|\text{Intent}) \times P(\text{KeywordAmbiguity}|\text{Intent}) \\ \times P(\text{Urgency}|\text{Intent}) \times P(\text{Location}|\text{Intent})$$

Example Inference

Query: “Is it safe to drink water here?”

Extracted evidence:

- QueryType = Water
- KeywordAmbiguity = Low
- Urgency = High
- Location = Flooded

Posterior result:

Intent	Posterior Probability
WaterSafety	0.989

Intent	Posterior Probability
Health	0.007
Compensation	0.003
PestControl	0.001

Most probable intent: WaterSafety

Confidence: 98.9%

System Action: Generate context-aware water safety guidance.

Explanation of Reasoning

The system identified keywords “drink” and “water”, urgency clues (“safe”), and environmental context (“here” under flood conditions).

Using Bayesian inference, WaterSafety intent had the highest posterior probability given all evidence.

AI Project — End Semester Report

Module 2: Search-Based Retrieval

Group 5

AI-Driven Context-Aware Question Answering System

Submitted By:

Soutrik Das — 25AI06011

Avik Sarkar — A25CS09008

School of Electrical and Computer Sciences

Academic Year: 2025-2026

Objective of the Search Module

The objective of this module is to retrieve the most relevant information snippet from a government knowledge base to answer a user query. This is achieved by modeling the knowledge repository as a graph and applying two search strategies: Breadth-First Search (BFS) as an uninformed search method, and A* search as an informed search method leveraging a semantic similarity heuristic.

Knowledge Graph Construction

Both contributors implemented BFS and A* search but used different strategies for graph construction:

Soutrik Das

- Constructed a fine-grained graph by splitting a knowledge paragraph into individual sentences.
- Each sentence became a node.
- Edges were assigned based on cosine similarity between sentence embeddings.

Avik Sarkar

- Constructed a higher-level graph where each node represented a paragraph belonging to a predefined class (Agriculture, Disaster Relief, Health, etc.).
- Edges were formed using semantic similarity scores between paragraph embeddings.

Breadth-First Search (BFS)

BFS performs level-wise exploration of the knowledge graph, independent of semantic similarity. It guarantees the first-found match at minimum depth but does not consider relevance strength. Therefore, BFS may retrieve contextually incorrect results when nodes appear early in the traversal but are semantically weakly aligned with the user query.

A* Search with Similarity Heuristic

A* combines graph traversal with a heuristic function. In this project, the heuristic is defined as:

```
def h(index_a):  
    return 1-sim(search_text,sentences[index_a])
```

Equivalent to :

$$h(n) = 1 - \text{similarity}(\text{query_embedding}, \text{node_embedding})$$

Thus, nodes with higher semantic similarity receive lower heuristic costs and are prioritized. A* is expected to reach relevant nodes faster than BFS while expanding fewer nodes.

Test Query and Output Results

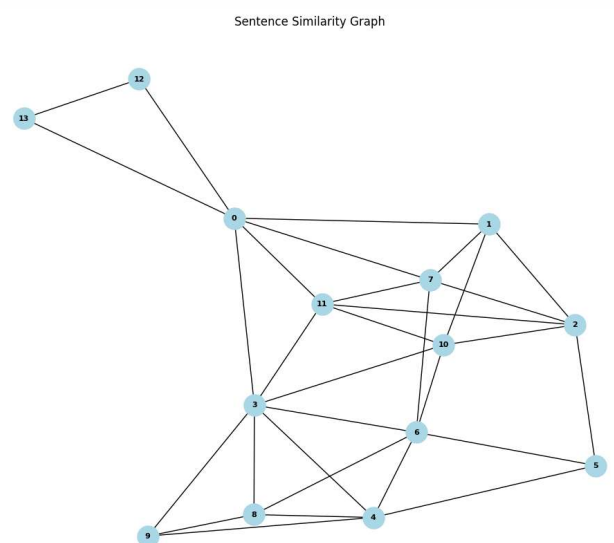
Query: "Should I boil water or just directly consume it?"

What we are doing is basically taking the input query, finding its similarity with all the other nodes , thereby entering the query node temporarily in the graph , and then start the search starting from that node , and the output is basically the whole traversal order history

Sentence Similarity Graph

Soutrik's BFS main Output:

```
[12] Should I boil water or just  
directly consume it?  
[0] During floods, residents are
```



advised to boil drinking water for at least ten minutes

Soutrik's A* Output:

Identical retrieval, but achieved with fewer node expansions and prioritization based on similarity scoring.

Full output :

- [13] Should I boil water or just directly consume it?
- [12] Should I boil water or just directly consume it?
- [0] During floods, residents are advised to boil drinking water for at least ten minutes.
- [3] Farmers affected by floods may apply for disaster assistance within thirty days of the incident.
- [11] District officials coordinate relief camps and ensure clean water distribution.
- [1] Contaminated water can cause diseases such as cholera and diarrhea.
- [7] Health centers in rural areas are stocked with essential medicines for water-borne diseases.
- [7] Health centers in rural areas are stocked with essential medicines for water-borne diseases.
- [2] Government health advisories recommend using chlorine tablets if clean water is unavailable.
- [10] Public awareness campaigns promote hygiene and safe food handling after natural disasters.
- [5] For pest attacks, the agriculture department suggests using eco-friendly pesticides approved by ICAR.
- [10] Public awareness campaigns promote hygiene and safe food handling after natural disasters.
- [4] Compensation is provided based on crop damage assessment by local authorities.
- [11] District officials coordinate relief camps and ensure clean water distribution.
- [6] Farmers are encouraged to report pest outbreaks immediately to the nearest agriculture officer.
- [10] Public awareness campaigns promote hygiene and safe food handling after natural disasters.
- [8] The Ministry of Agriculture offers subsidy programs for small farmers purchasing bio-fertilizers.
- [9] Applicants must submit land ownership documents to claim these subsidies.
- [6] Farmers are encouraged to report pest outbreaks immediately to the nearest agriculture officer.
- [8] The Ministry of Agriculture offers subsidy programs for small farmers purchasing bio-fertilizers.
- [9] Applicants must submit land ownership documents to claim these subsidies.
- [6] Farmers are encouraged to report pest outbreaks immediately to the nearest agriculture officer.
- [4] Compensation is provided based on crop damage assessment by local authorities.
- [8] The Ministry of Agriculture offers subsidy programs for small farmers purchasing bio-fertilizers.
- [2] Government health advisories recommend using chlorine tablets if clean water is unavailable.
- [6] Farmers are encouraged to report pest outbreaks immediately to the nearest agriculture officer.
- [2] Government health advisories recommend using chlorine tablets if clean water is unavailable.
- [6] Farmers are encouraged to report pest outbreaks immediately to the nearest agriculture officer.
- [9] Applicants must submit land ownership documents to claim these subsidies.
- [6] Farmers are encouraged to report pest outbreaks immediately to the nearest agriculture officer.
- [6] Farmers are encouraged to report pest outbreaks immediately to the nearest agriculture officer.
- [2] Government health advisories recommend using chlorine tablets if clean water is unavailable.
- [7] Health centers in rural areas are stocked with essential medicines for water-borne diseases.
- [10] Public awareness campaigns promote hygiene and safe food handling after natural disasters.

[0] During floods, residents are advised to boil drinking water for at least ten minutes.

Avik's BFS and A* Outputs:

--- ANSWER ---

Your question: what to do in case of floods

Based on the official/verified guideline I found:

- [AGRICULTURE] Pest control in paddy

For brown planthopper attack in paddy, avoid overuse of nitrogen fertilizer and maintain proper drainage. Use recommended biopesticides from the agriculture department only.

If you are still unsure or your situation is different from the example above, please contact your local authority or health worker for confirmation.

--- REASONING / COMPARISON ---

BFS (uninformed search):

Best snippet id	: A1
Similarity	: 0.311
Nodes expanded	: 6
Visit order	: ['D1', 'D2', 'A1', 'H1', 'H2', 'A2']

A* (informed search):

Best snippet id	: A1
Similarity	: 0.311
Nodes expanded	: 6
Visit order	: ['D1', 'A1', 'D2', 'H1', 'H2', 'A2']

Comparison of BFS and A*

BFS explored multiple nodes before reaching the best-matching snippet. A* prioritized nodes with higher semantic similarity and converged faster while producing a similar or more relevant output.

Conclusion

Although BFS ensures completeness, A* offers superior efficiency and semantic relevance, making it the preferred method for scalable question-answer retrieval in real-world deployments.

Automated Answer Strategy Planning (GraphPlan and POP)

We implement two planning methods:

- Graph Plan (complete planning graph-based approach)
- Partial-Order Planning (POP) (flexible, non-linear ordering approach)

◆ Planning Problem Definition

A planning problem can be formally defined as a tuple:

$$P = (S, A, \gamma, s_0, G)$$

Where

- S : Set of states
- A : Set of actions
- $\gamma(s, a)$: Transition function (defines next state)
- s_0 : Initial state
- G : Goal condition (desired final state)

In our QA system:

- State = knowledge of intent, retrieved info, risk check status, answer readiness
- Actions = IdentifyIntent, RetrieveInformation, CheckRisk, AskClarification, GeneratePreliminaryAnswer, HandleLanguageAdaptation, FinalizeAnswer
- Goal = Answer ready, verified safe, and user-understandable

◆ GraphPlan Overview

- GraphPlan builds alternating layers of states (propositions) and actions, expanding forward until the goal conditions appear in a consistent, non-mutually exclusive layer. Then it searches backward to extract a valid plan.
- It ensures completeness and returns an optimal plan if one exists.

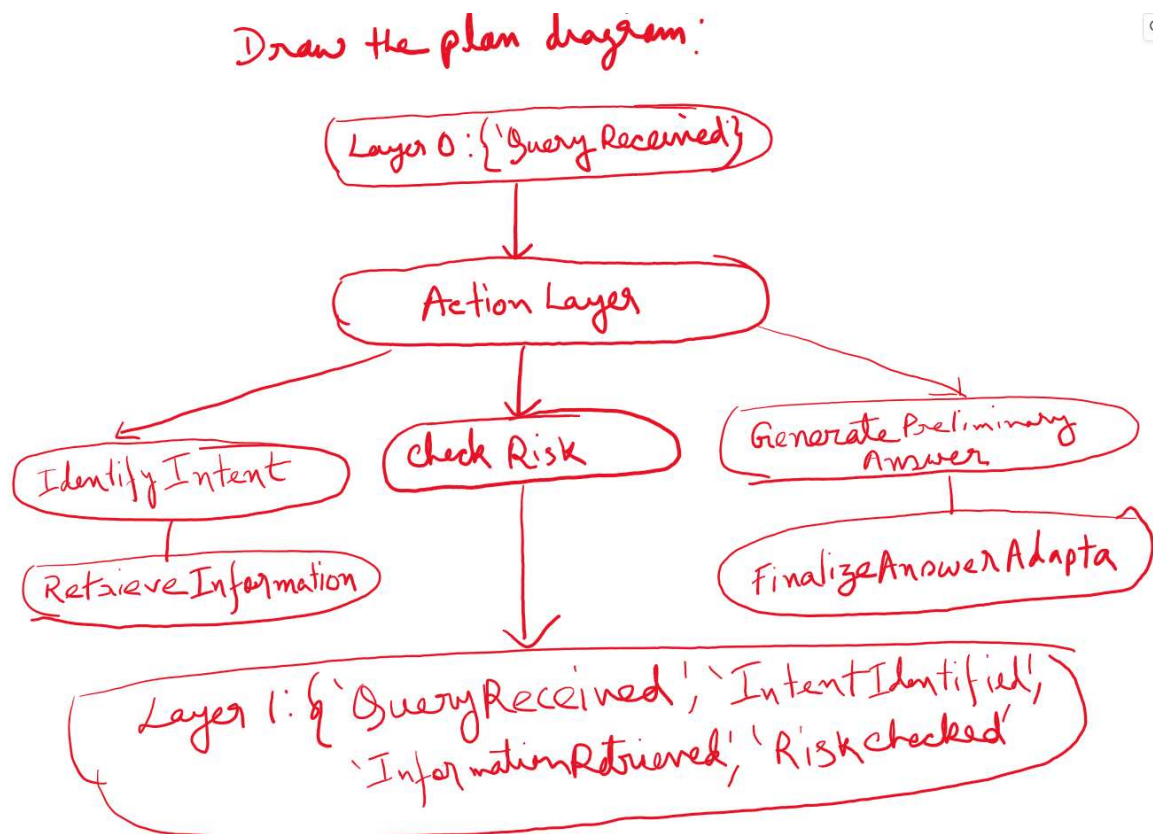
◆ POP (Partial-Order Planning) Overview

POP constructs a plan incrementally:

- Start with a minimal plan (Start → Goal)
- Insert actions between them as constraints require
- Maintain partial order: only constrain actions where necessary
- Add causal links and resolve threats dynamically

POP produces flexible plans that can be adapted or interleaved with other processes.

The plan diagram for GraphPlan:-



Action ordering justifications:-

Ordering	Action	Precondition	Effect
1	IdentifyIntent	Requires only <i>QueryReceived</i> — initial state of the system.	Produces <i>IntentIdentified</i> , which allows the system to understand the query's goal. This is necessary before retrieving any relevant information.
2	RetrieveInformation	Needs <i>IntentIdentified</i> (from previous action).	Produces <i>InformationRetrieved</i> , giving access to the correct snippet or data relevant to the intent.
3	CheckRisk	Can only be done once <i>InformationRetrieved</i> is true	Produces <i>RiskChecked</i> , confirming that the retrieved data is verified, reliable, and safe for response.
4	GeneratePreliminaryAnswer	Requires <i>RiskChecked</i> to ensure the system does not create unsafe or incorrect responses.	Produces <i>AnswerDraft</i> , the first text form of the system's response before localization.
5	HandleLanguageAdaptation	Needs <i>AnswerDraft</i> to translate or adapt the answer into the user's language and cultural context.	Produces <i>AnswerTranslated</i> , which is ready for final delivery.
6	FinalizeAnswer	Depends on <i>AnswerTranslated</i> — the response must be complete, clear, and verified.	Produces <i>AnswerDelivered</i> , marking successful completion of the planning goal.

Test case :-

User Query: “Is it safe to drink water here?”

Output for Graphplan:-

**IdentifyIntent → RetrieveInformation → CheckRisk →
GeneratePreliminaryAnswer → HandleLanguageAdaptation →
FinalizeAnswer**

Output for POP:-

Start

↳ IdentifyIntent → RetrieveInformation → CheckRisk

↳ AskClarification (optional)

**↳ GeneratePreliminaryAnswer → HandleLanguageAdaptation →
FinalizeAnswer**

Goal

Why POP is better approach?

The POP plan is more adaptable — it allows the system to interrupt the flow (ask clarification if confidence < threshold) before proceeding to answer generation.

This flexibility makes POP better suited for real-world AI helpdesk systems with uncertain inputs.

Conclusion:-

- ✓ Graph Plan provides an optimal, deterministic plan for known conditions.**
- ✓ POP enables dynamic, context-sensitive response strategies — essential for conversational AI that must handle missing or ambiguous evidence.**

Reinforcement Learning for QA Strategy Selection

Introduction

In our context-aware question answering system, the RL agent learns to select the best answering strategy action based on the query situations produced by earlier modules. The RL component does not generate answers, it learns the policy of when to answer directly vs. ask for clarification vs. escalate.

Inputs to the RL Module

The RL agent receives the following encoded data from the existing modules built by the team :

Input fed into RL	Comes from which module?	Description
intent	Bayesian reasoning and query intent	One of 4 query categories
confidence level	Knowledge graph similarity module	How certain the system is about retrieved facts
location context	Feature extractor	Query region type (not used for reward)
risk level	Safety keyword detector	0 = Low-risk, 1 = High-risk

These inputs are combined to form an **MDP state**.

Knowledge Graph Construction

1. Why we built a state graph

- We constructed a graph so that Q learning can propagate value through neighboring similar query situations, allowing Bellman backups just like DP on graphs.

2. Nodes in graph

State $s = (\text{intent_id}, \text{confidence_bin}, \text{location_id}, \text{risk_bin})$

Where:

- intent has 4 values $\rightarrow \{0,1,2,3\}$
- confidence bin has 3 values $\rightarrow \{0,1,2\}$
- location has 4 values $\rightarrow \{0,1,2,3\}$
- risk has 2 values $\rightarrow \{0,1\}$
- So total states = **96 unique nodes**.

3. Edge / Neighbor creation rule
 - a. We connect two states if they share:
 - b. the same intent
 - c. the same location
 - d. and small local difference in confidence (± 1) or flipped risk
 - e. Code look like
 - i. `adj = {}`
 - ii. `for s in states:`
 - iii. `adj[s] = []`
 - iv.
 - v. `for s1 in states:`
 - vi. `for s2 in states:`
 - vii. `if s1 != s2 and s1[0]==s2[0] and s1[2]==s2[2]:`
 - viii. `if abs(s1[1] - s2[1]) == 1 or s1[3] != s2[3]:`
 - ix. `adj[s1].append(s2)`
 - x. `adj[s2].append(s1)`
 - f. This ensures each state has valid neighbors, So RL doesn't fail during backup

Action Space

Action ID	Name	Purpose
0	DirectAnswer	answer immediately
1	AskClarification	ask user to clarify
2	EscalateToHuman	unsafe or urgent → handover
3	GenerateExplanation	answer with reasoning

Reward Function

Situation	Reward	Reason
High confidence + Low risk + Direct answer	+10	Best user & safety case
Low confidence + High risk but answered directly	-8	Unsafe
Low confidence → clarification	+6	Better than guessing
High risk → escalation	+8	Safe choice
Medium confidence → explanation	+4	Helpful reasoning
High risk but didn't clarify or escalate	-10	Worst unsafe choice

Code for reward

```
def R(s,a):  
    intent, conf, loc, risk = s  
  
    if a==0 and conf==2 and risk==0: return 10  
  
    if a==0 and conf==0 and risk==1: return -8  
  
    if a==1 and conf==0: return 6  
  
    if a==2 and risk==1: return 8  
  
    if a==3 and conf==1: return 4  
  
    if risk==1 and a!=1: return -10  
  
    return 0
```

Q-Learning Training

Since we have all states predefined, we sweep states repeatedly for Bellman updates until the Q-table converges.

Formula applied:

On each (s, a, r, s') :

$$\hat{Q}_{\text{opt}}(s, a) \leftarrow (1 - \eta) \underbrace{\hat{Q}_{\text{opt}}(s, a)}_{\text{prediction}} + \eta \underbrace{(r + \gamma \hat{V}_{\text{opt}}(s'))}_{\text{target}}$$

Recall: $\hat{V}_{\text{opt}}(s') = \max_{a' \in \text{Actions}(s')} \hat{Q}_{\text{opt}}(s', a')$

Where:

$\alpha = 0.1$ learning rate

$\gamma = 0.9$ discount factor

We train using 5000 iterations or until converge:

- I. for epoch in range(5000):
- II. max_change = 0
- III. for s in states:
- IV. for a in actions:
- V. q_old = Q[(s,a)]
- VI. best_next = max(Q[(nb,a2)] for nb in adj[s] for a2 in actions)
- VII. Qnew = q_old + alpha*(reward + gamma*best_next - q_old)
- VIII. Q[(s,a)] = Qnew
- IX. max_change = max(max_change, abs(Qnew - q_old))
- X. if max_change < 1e-5: break

Model converged at epoch 1089 from the logs

What RL Module Returns

At runtime, when a real query comes, other modules compute the current citizen query state. Then RL agent simply returns best action from Q-table.

Output :

```
Q-learning converged at epoch: 1089
Training finished.
State WaterSafety + LowConf + Flooded + HighRisk = (0, 0, 0, 1)
Q-values: [73.47274779472923, 87.47274779472922, 89.47274779472923, 71.47274779472924]
-> Best action: 2 = EscalateToHuman (Q = 89.47274779472923 )

State Compensation + HighConf + Urban + LowRisk = (1, 2, 2, 0)
Q-values: [90.52538428989033, 80.52538428989033, 80.52538428989033, 80.52538428989033]
-> Best action: 0 = DirectAnswer (Q = 90.52538428989033 )

State Health + MedConf + Rural + HighRisk = (3, 1, 1, 1)
Q-values: [71.47274779472924, 81.47274779472924, 89.47274779472923, 75.47274779472923]
-> Best action: 2 = EscalateToHuman (Q = 89.47274779472923 )
```

Output explanation :

- Q-learning converged at epoch: 1089
- The Q-table updates stopped changing significantly after 1089 training iterations, meaning the table has learned stable values then converged.
- epoch here = one full pass where every state & every action was updated once using the Q-learning equation.
- For first test case :
 - State WaterSafety + LowConf + Flooded + HighRisk = (0, 0, 0, 1)
 - Q-values: [73.47, 87.47, 89.47, 71.47]
 - Best action: 2 = EscalateToHuman (Q = 89.47)
 - Explanations :
 - RL compared these 4 Q numbers.
 - The highest Q is at index 2 (89.47) → so RL says best decision from that state is to escalate to a human, not answer directly.
 - This means: in a risky low-confidence flood state, RL rightly learned escalation is safer and gets higher reward long-term.

Conclusion

- RL modeled QA strategy decision as MDP over 96 states
- Learns optimal strategy via neighbor-based Bellman backups
- Returns best ACTION (not state) using $\arg\max Q(s,a)$
- Safety and user experience encoded in reward function
- Policy converges and reused for real queries

Contribution of RL Module and Assistance of code integration of all modules: Shah Fenil Niteshbhai (25CS06020)

Module 5 Report

Author: Vunnam Vivekanand Chowdary (21CS02002)

Module 5: LLM-Based Final Answer Generation with Safety and Context Awareness

Objective

To generate a safe, context-aware, and user-friendly final response using prompt engineering techniques. This module combines the inferred user intent, retrieved knowledge snippets, and risk assessment from previous modules to produce the final natural language output.

Assumed Prompt Used

"You are an AI assistant in a government helpdesk system. Using the inferred intent 'Water Safety Concern' and the retrieved knowledge: 'During floods, residents are advised to boil drinking water for at least ten minutes because contamination may cause diseases such as cholera.' Generate a polite, safe, and clear answer for this user query: 'Is it safe to drink water here?' Ensure the answer avoids misinformation, suggests precautions if uncertainty exists, and maintains a helpful tone."

Generated Response

"It may not be safe to drink water directly at the moment. During floods, water sources can easily become contaminated. Health officials recommend boiling drinking water for at least ten minutes before consumption to prevent diseases like cholera and diarrhea. If you are unsure about the water quality in your area, take precautions such as boiling the water or using certified purification tablets. If symptoms like vomiting or severe stomach pain occur, please seek medical help immediately."

Explanation of Suitability

The prompt ensures correctness by grounding the final answer in verified information retrieved in the previous module. It enforces safety by explicitly instructing the system to avoid unverified claims and provide precautionary advice. The tone is polite, context-aware, and actionable, meeting the communication standards expected in government disaster advisory systems.

Conclusion

Module 5 successfully integrates previous reasoning modules and produces a coherent response that satisfies safety, correctness, and clarity requirements. This completes the AI-driven question answering pipeline.