iCARE: Ontology-Guided Intent Routing for Multi-Agent LLM-Based Dialogue Systems*

Nirmalie Wiratunga[1], Vihanga Ashinsana Wijayasekara[1], Ikechukwu Nkisi-Orji[1], Pedram Salimi[1], Kyle Martin[1] and Cristina Bolaños[2]

[1]*Robert Gordon University, Aberdeen, UK*
[2]*Universidad de Castilla – La Mancha, Spain*

Abstract

We present *i*CARE, a modular multi-agent reasoning framework that integrates *i*CARE-Onto with Case-Based Reasoning (CBR) to support ontology grounded, LLM-based dialogue management. The framework addresses intent disambiguation and intent routing within an agentic architecture, where specialised agents handle domain-specific tasks. Agent outputs are combined to produce accurate responses through a composite strategy that couples few-shot prompting with Retrieval-Augmented Generation (RAG) and a CBR-based fallback mechanism. Conversational content is first grounded in ontological concepts that capture situational, personal, and dialogue knowledge. These are stored as short-term memory entries and subsequently organised into semantic, procedural, and episodic summaries that form long-term memory. This structured memory ensures factual accuracy, coherence, and contextual relevance beyond surface fluency in generated responses. While the framework encompasses multiple reasoning components, in this paper we focus on how the system reasons over user utterances to form clarified intents for routing to appropriate agents. Preliminary experiments on a synthetic dataset show a 20% relative improvement in routing accuracy for the LLM-as-Router when ontology-based disambiguation is applied. A comparative evaluation with a lazy-learner, ProtoKNN, demonstrates its potential as a low-latency, offline alternative, suggesting opportunities for hybrid routing in dialogue systems.

Keywords

Dialogue with LLM, Multi-agent Intent Routing, Ontology driven RAG, Healthcare AI

1. Introduction

Large language models (LLMs) have transformed natural–language interfaces from brittle rule engines into fluid, mixed-initiative dialogue partners [1, 2, 3]. Yet, in high-stakes domains like health, law, and finance, unqualified LLM output is risky since hallucinated facts, hidden biases, and inconsistent reasoning can harm users [4]. We argue that interface fluency must be paired with rigorous knowledge grounding and transparent control. Ontologies provide the semantic scaffolding for such grounding, while modular, specialist agents keep decision logic interpretable and auditable.

Based on our experience developing intelligent healthcare systems, such as a conversational framework for embodied care robots, we observed that even routine tasks (e.g., "remind me to take my pills") often require cross-checking medication rules, user history, and sensor data, going well beyond a single turn of question and answer. More recently, we are developing a digital-avatar health coach to support adolescent and young adult cancer survivors in managing cardiovascular health through a non-intrusive conversational interface embedded in their environment. Such scenarios demand dialogue to be personalised, evidence-grounded, and safety-constrained. To meet these needs, we are introducing *i*CARE as a modular multi-agent reasoning framework with these properties:

1. An ontology (*i*CARE-Onto) to model structured domain and dialogue knowledge;
2. A semantic router for intent classification for agent actioning;
3. The dialogue context control using retrieval-augmented generation (RAG), with fallback to case-based retrieval when attribute-level matching is essential for accurate responses; and

LACATODA2026: The 10th Linguistic and Cognitive Approaches to Dialog Agents Workshop at the 40th AAAI conference

n.wiratunga@rgu.ac.uk (N. Wiratunga); v.wijayasekara@rgu.ac.uk (V. A. Wijayasekara); i.o.nkisi-orji@rgu.ac.uk (I. Nkisi-Orji); p.salimi@rgu.ac.uk (P. Salimi); k.martin3@rgu.ac.uk (K. Martin); Cristina.Bolanos@uclm.es (C. Bolaños)

4. A dual memory approach, combining short-term (dynamic) session memory and distilled long-term memory, to support continual personalisation.

We also present results from a comparative evaluation study on ontology-guided intent routing using the *i*CARE framework (specifically evaluating properties 1 and 2).

2. Multi-Agent Dialogue Framework

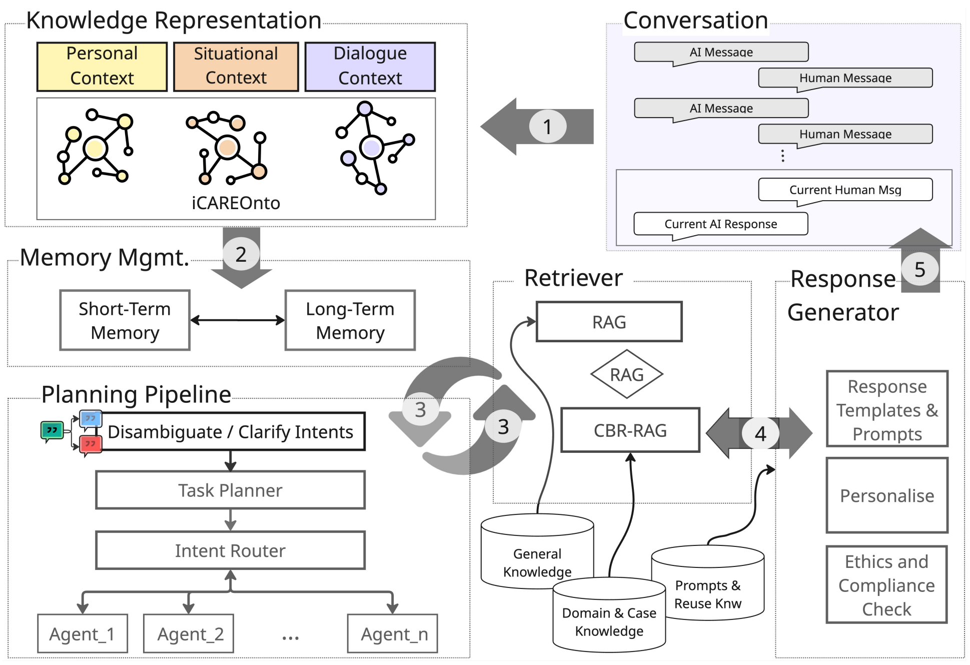

Figure 1: *i*CARE framework components. (1) The current conversation turn is grounded in the *i*CARE-Onto, combining personal, situational, and dialogue facets. (2) Episodic facts are synchronised to long-term and short-term memory. (3) The planning pipeline disambiguates intents and routes them to specialised agents for execution and can also retrieve information for standard RAG or use *CBR-RAG*. (4) The output then feeds the response generator, which applies personalisation and an ethics/compliance check before final output.

The *i*CARE framework couples fact-grounded dialogue with modular control (Figure 1). One or more dialogue intents are handled by agents. Each agent interfaces with internal or external tools (e.g., a risk prediction model) to execute its assigned tasks and can retrieve relevant information to validate responses and incorporate educational and explanatory content. Contextual information is drawn from shared **long-term** and **short-term** memory, organised by an ontology that represents concepts relevant to the user's **personal** context, necessary **situational** awareness, and ongoing **dialogue** context (including current conversation goals) [5, 6].

The planning pipeline begins with an ontology-guided intent disambiguation. If the human dialogue utterances are found to be under-specified and requiring slots or maps to multiple ontology classes/intents, *i*CARE asks one or two targeted clarifying questions. The user's replies are then stored in triple form within short-term memory, supporting the maintenance of dynamic conversational context. The resulting (disambiguated) context is then mapped to canonical intent ID(s). If no suitable semantic intent match is found, it falls back to the label OTHER and is flagged for human intervention. These intents and resolved slots update short-term memory and are promoted to long-term memory to manage conversational continuity. The mapping between short and long-term memory must consider not only what to summarise from an extended dialogue but also how to do so appropriately [7].

Given these fully clarified intents, the planner decomposes them into mini-plans consisting of intent steps that can be executed by specialised agents. Determining which agent to invoke becomes a routing task, which can draw on knowledge about the agents, such as capability and dependency information, including agent descriptions, preconditions, and data dependencies. The planner also coordinates sequential or parallel execution to maintain coherent dialogue flow [8]. For example, when asked "Would a run this afternoon be recommended?", the planner may first gather required vitals, then fuse outputs from Biomarker, Risk, and Exercise agents, perhaps suggesting a brisk walk instead. Finally, the Ethics–Policy agent checks compliance and, if it blocks the request, returns a counterfactual guided response, such as: "If your heart-rate threshold were lower, I could approve that run".

In trivial plans, an intent may directly map to an agent; however, multiple intents may be handled by a single agent (many-to-one), and a given intent may be relevant to different agents across plan steps (many-to-many). Such routing structures have precedent in specialist-agent coordination frameworks[9]. In the many-to-many case, a case library of prior plan steps (queried via case-based retrieval or few-shot exemplars) can guide mapping when multiple handlers are available or when policy/availability matters (e.g., cost tiers, privacy).

2.1. Knowledge Representation

An example of domain concepts (with some instantiations) and relationships modeled by *i*CARE-Onto appears in Figure 2. Unique to this modeling are the memory type annotations (coloured tags in the figure) that help structure short and long-term memory, supporting coherence in dialogue management. Specifically, three types can be identified: **semantic**, such as factual knowledge that should ideally be represented as relational triples; **episodic**, where information is tied to specific events, timestamps, or conversational workflows (e.g., an intent followed by a recommendation and later a user-reported outcome); or **procedural**, where a known procedure needs to be carried out step by step by an agent (see examples in Table 1).

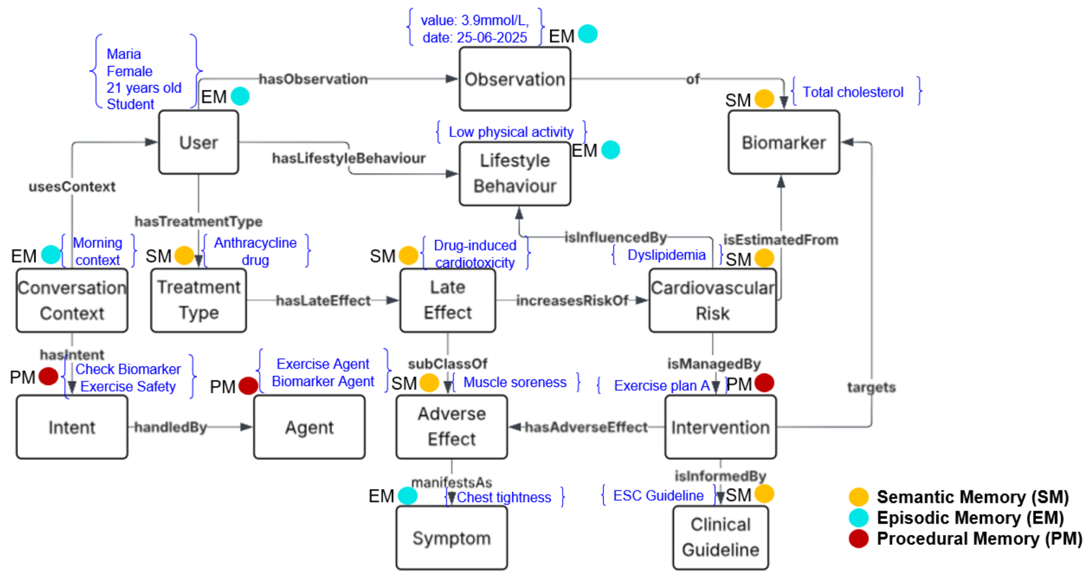

Figure 2: Memory annotation of the ontology. Coloured dots indicate where concepts typically instantiate as episodic (EM), semantic (SM), or procedural (PM) memory.

Table 1
Example system responses for (E)pisodic, (S)emantic and (P)rocedural

User Utterance	Intent (Memory)	System Response Construct
What's my last cholesterol level?	Recall Personal Info (E)	It was 3.9 mmol/L on 25 Jun 2025.
Why is my cholesterol a problem?	Explain or Educate (S)	High cholesterol inc cardiovascular risk.
What can I do to lower it?	Recommend or Guide (P)	Step through PhyxioPlanA exercises.

Accordingly, each ontological concept can be annotated with a preferred memory type, so that when concrete instantiations arrive, the system knows where to store them. For example, *Biomarker* instances such as “Total cholesterol = 3.9 mmol/L on 25–06–2025” are of episodic type memory; the class-level fact “Dyslipidaemia increases cardiovascular risk” associated with *Late Effect* → *Cardiovascular Risk* is semantic memory; and the multi-step *Exercise PhyxioPlanA* under *Intervention* is recognised as procedural memory (as it will have a fixed set of exercises).

2.2. Planning Pipeline: Disambiguation with iCARE-Onto

Since utterances are mapped to intents for planning, ill-formed or underspecified inputs must first be detected and disambiguated (via brief, ontology-guided clarifications) before routing [10]. For this, *i*CARE-Onto (see Figure 2) is used in the disambiguation phase to harness the structured relationships among classes, properties, and instances to detect when essential information is missing or underspecified. For example, if the user utterance provides a numeric value without specifying what it refers to (e.g., "is 140 okay?"), the ontology can be used to expose that several properties of the relevant class (e.g., Biomarkers - blood pressure, heart rate, glucose) could accept such a value. Additionally, the range of possibilities can further be narrowed given the context of the user, such as previous recorded measurements. When there is insufficient detail in the dialogue context to resolve the ambiguity, this ontological knowledge allows the system to identify the precise property that is missing and to generate a targeted clarification follow-up question (e.g., "What measurement does 140 refer to?" or "Did you mean your last BP reading?”). Similarly, when an utterance maps to multiple possible classes or relations, ontology constraints (domain, range, cardinality) are used to prune candidates or formulate the most discriminative question to minimise the number of clarification turns. By systematically detecting ambiguity patterns based on ontology structure (e.g., missing property type, missing relation arguments, multiple class candidates), short clarifying turns can be used to refine or complete the semantic representation of an utterance through follow-up questions. Figure 3 shows how brief ontology-guided clarification turns, together with short and long-term memory, collect the minimal information needed before it is handed over to the planner for agent assignment (here to the Triage agent).

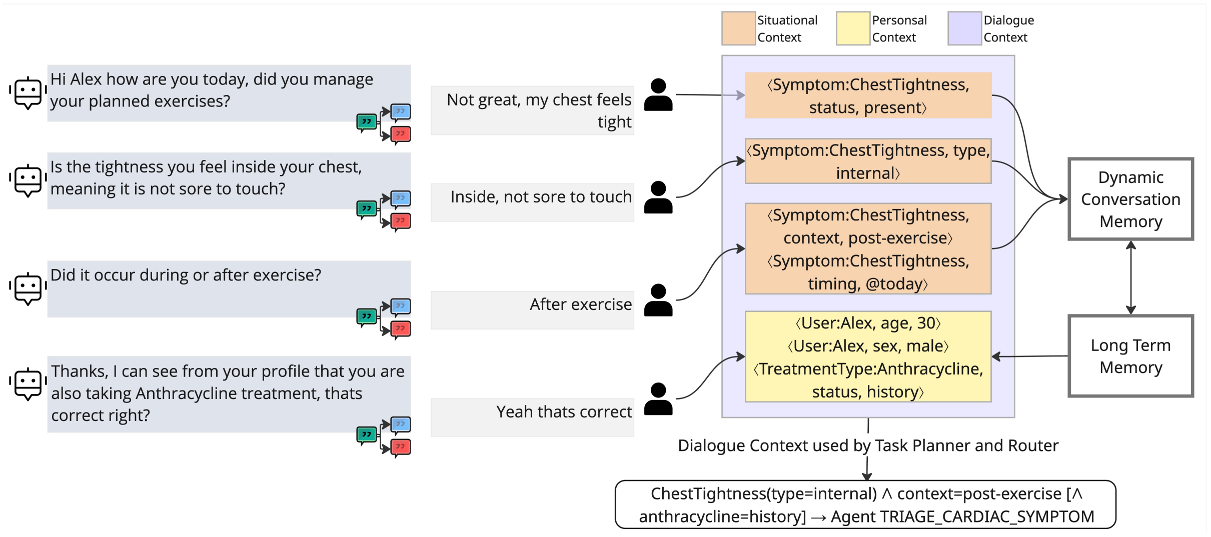
Situational Context
Personsal Context
Dialogue Context
Hi Alex how are you today, did you manage your planned exercises?
Not great, my chest feels tight
Is the tightness you feel inside your chest, meaning it is not sore to touch?
Inside, not sore to touch
Did it occur during or after exercise?
After exercise
Thanks, I can see from your profile that you are also taking Anthracycline treatment, thats correct right?
Yeah thats correct
⟨Symptom:ChestTightness, status, present⟩
⟨Symptom:ChestTightness, type, internal⟩
⟨Symptom:ChestTightness, context, post-exercise⟩
⟨Symptom:ChestTightness, timing, @today⟩
⟨User:Alex, age, 30⟩
⟨User:Alex, sex, male⟩
⟨TreatmentType:Anthracycline, status, history⟩
Dynamic Conversation Memory
Long Term Memory
Dialogue Context used by Task Planner and Router
ChestTightness(type=internal) ∧ context=post-exercise [∧ anthracycline=history] → Agent TRIAGE_CARDIAC_SYMPTOM

Figure 3: Ontology guided disambiguation of user utterances. Both short and long-term memory are used in this example, and extracted triples are colour-coded by context (situational, personal, dialogue). The resulting fully clarified intents are ready for use by the Task Planner. Ontological properties of ‘Symptom’ such as ‘type’ and ‘status’ help to guide the clarification turns.

Listing 1: RAG prompt skeleton over the disambiguated text

```
Role: Task Planner in a multi-agent dialogue system.

Input (disambiguated utterance):
triples:
 User:u123 | hasBiomarker | TotalCholesterol
 Biomarker:TotalCholesterol | last_value | 6.7 mmol/L
     @2025-10-10
 Conversation:Goal | ask | risk_explanation
 Conversation:Goal | ask | exercise_plan_start?

Ontology intent labels (retrieval top-k):
 CHECK_BIOMARKER, ASSESS_RISK, ASK_EXERCISE_PLAN,
     SET_REMINDER, ...

Instructions:
 1) Split the user request into plan steps.
 2) Find best match intent id from list, else "OTHER".
 3) Return JSON with fields: step_id, intent_id, span,
      args, depends_on, confidence.
```

Listing 2: Worked example output

```
Utterance (disambiguated):
"Last cholesterol? Should I worry?
    Start the exercise plan?"

Output JSON:
[
 {"step_id":"s1",
  "intent_ids":[
  "CHECK_BIOMARKER","ASSESS_RISK"
  ],
  "args":{"biomarker":"
      TotalCholesterol","value":
      "6.7 mmol/L","when":
      "2025-10-10"}}

  {"step_id":"s2","intent_id": "
      ASK_EXERCISE_PLAN",
  "depends_on":["ASSESS_RISK"]}
]
```

Figure 4: Planning operates on the disambiguated frame (triples) produced by ontology-guided clarification for "Hi! What's my last cholesterol reading? Should I worry? Would it be OK to start the recommended exercise plan?". The RAG step retrieves candidate intents and returns structured items with dependencies for the planner. The routing to agents happens after this.

2.3. Planning Pipeline: Decomposing with Task Planer

In *i*CARE, a retrieval-augmented LLM first embeds the disambiguated user utterance, and retrieves the top-k ontology intent labels. It segments the request and returns a JSON array of plan-step candidates with fields (e.g., step_id, intent_id, args, depends_on, similarity). See Figure 4 and Listings 1–2, for an example involving the utterance "Hi! What's my last cholesterol reading? Should I worry? Would it be OK to start the recommended exercise plan?" (The JSON shown is illustrative, not exhaustive). The **Task Planner** then organises these candidate intents into a mini-plan, capturing dependencies, such as needing the biomarker values for risk assessment, and marks any parallelisable steps or follow-ups.

Next, the Intent Router binds each step to an appropriate agent. It first matches the disambiguated frame against intent schemas to identify candidate handlers. When a single intent in a step is uniquely handled by an agent, the mapping is obtained directly from the ontology's `handledBy` relation. When multiple candidate agents exist, a learned router (few-shot LLM-as-Router or ProtoKNN) selects the best match based on the step context. Compound steps containing multiple intents are similarly mapped to an agent capable of performing all included actions. Low-confidence cases can trigger a human-in-the-loop recommendation.

In the given example in Figure 4, the router maps the intents in the two steps to agents: (CHECK_BIOMARKER, ASSESS_RISK) $\rightarrow$ RiskPredictionAgent, and ASK_EXERCISE_PLAN $\rightarrow$ ExerciseCoach. The planner enforces the intent dependency order CHECK_BIOMARKER $\rightarrow$ ASSESS_RISK $\rightarrow$ ASK_EXERCISE_PLAN and fuses the agents' outputs into a single coherent reply.

2.4. Planning Pipeline: Intent Router

We explore several approaches to intent routing using both LLMs and more resource efficient matching methods. With the **LLM-as-Router** setup, a prompt (see Figure 5) with few-shot in-context exemplars per agent is used to predict the agent given a disambiguated dialogue context frame. LLMs are strong zero-/few-shot routers: a single prompt with a handful of exemplars per agent can yield competitive routing without task-specific training. Because they encode broad world knowledge (approximateretriever behaviour from large-scale pretraining), they cope well with open phrasing, rare synonyms, and long-tail constructions. This makes an LLM-as-Router attractive for rapid onboarding of new intents and agents, early-stage pilots with scarce exemplars, and domains where ontology cues (units, dates, history) can be injected post-disambiguation directly into the prompt.

```
You are the intent router for the iCare healthcare robot framework. Route the (already disambiguated) user
(U) request to the correct internal task agent (A).

   Disambiguated User Input:
   - A clarified user utterance.
   - Semantic triples capturing entities, relations, and temporal or contextual tags (e.g., @now, @2025-10-10).

   Agent Categories (excerpt): BiomarkerAgent, ExerciseCoach, ... ,TriageAgent
   Return ONLY the agent's name (exactly). No other text.

   FEW-SHOT (1-shot exemplars; frames include triples):

   U: "BP was 140/90 just now, so can I still do an easy run?"
   Triples: [Biomarker:SystolicBP | value | 140 mmHg @now,
             Biomarker:DiastolicBP | value | 90 mmHg @now,
             Conversation:ExerciseContext | planned | Run 20 min easy]
   A: ExerciseCoach

   U: "What's my last cholesterol reading?"
     Triples: [Biomarker:TotalCholesterol | last_value | 6.7 mmol/L @2025-10-10]
   A: BiomarkerAgent
   ...
```

Figure 5: LLM-as-Router prompt extract, with few shots stating U:Utterance; A: Agent; and clarified Triples

The alternative is the non-parametric ProtoKNN router, a lazy learner using a Class–Prototype k-Nearest-Neighbour based router strategy, inspired by Prototypical Networks [11]. To construct prototypes, for every agent $a \in \mathcal{A}$ we keep a small support set (like the few-shots in the LLM-as-Router method), $\mathcal{S}_a = \{(\mathbf{z}_i, y_i)\}_{i=1}^{N_a}$. Here $\mathbf{z}_i$ is the embedding of the disambiguated intents (i.e., resolved dialogue context plus ontology-derived cues) and y the agent label. The class prototype for agent a is the mean of its support embeddings; $\mathbf{p}_a = \frac{1}{N_a} \sum_{(\mathbf{z}_i, y_i) \in \mathcal{S}_a} \mathbf{z}_i$. Using these prototypes, the model selects the relevant agent based on pairwise comparisons of the intent (plus disambiguated context) embeddings and prototype embeddings. Given one or more intent embeddings $\mathbf{q}$ from the planner's mini-plan step, we compute its distance to each prototype, and assign it to the agent with the nearest prototype: $a^\star = \arg\min_{a \in \mathcal{A}} \|\mathbf{q} - \mathbf{p}_a\|_2$, optionally, a confidence margin Δ may be enforced. If matching confidence is low or ambiguous, control may need to be handed over to a catch-all General agent for clarification.

ProtoKNN's advantages are fourfold. First, it is few-shot friendly: centroids can be calculated with as few as five to ten labelled examples per agent [12], whereas keyword rules [13] require exhaustive vocabularies. Second, it is tolerant of noise: averaging the support embeddings smooths over minor wording differences, so small lexical variations are unlikely to cause misclassification. Third, it remains interpretable: at inference the router can reveal the k-nearest prototype sentences (for example, "matched *remind-me* agent's prototype with similarity 0.92"), thereby providing human-readable justification for routing decisions. Fourth, comparison of fixed embeddings makes routing both cheaper and deterministic, unlike LLM-as-Router methods that vary with wording. However the ProtoKNN does require representative coverage across intents; performance may lag if the support set omits common phrasings or context variants and the embedding model is not appropriate for the domain. This is unlike the LLM-as-Router which does have access to broad world knowledge.

2.5. Response Generation in iCARE Framework

In *i*CARE, agents produce outputs that must be presented coherently to the user. Some responses are straightforward results, while others are explanatory, designed to educate the user about their health, for example, through a question–answer dialogue. To ensure factual reliability, each response is grounded in external knowledge rather than relying solely on the LLM's parametric memory [6].

A conventional, vector-based **RAG** module embeds the user's disambiguated utterance frame, together with its candidate intents, as a single vector and returns the closest passages from a knowledge store that contains both **general** medical literature and **domain**-specific documents (e.g., ESC guidelines on cardio-oncology) and any **case** knowledge (e.g., patient instances). This approach works well for simple, self-contained questions such as "*What is dyslipidaemia?*" However, complex health queries can bundle several clinically important attributes:

> "I finished anthracycline chemotherapy last year, my cholesterol is still high, and I'd like an exercise plan that minimises cardiac risk – any recommendations?"

Treating the whole disambiguated utterance as one vector returns generic exercise advice and ignores treatment history. Instead, we plan to adopt **CBR-RAG** [14], which represents the structured intent as attribute–value pairs (treatment = anthracycline, biomarker = cholesterol, goal = exercise, plus persona data), and matches locally against stored cases. Because situational and personal facts live in the ontology, CBR-RAG can retrieve the case where "Exercise PhyxioPlanA" was personalised for the treatment (e.g., anthracycline), yielding a far more relevant recommendation. At run-time the ontology-organised intents guide the retriever to use basic RAG when no fine-grained attributes are present, and to fall back to CBR-RAG whenever attribute-level matching is required. The Retriever also has access to *i*CARE's short and long-term memories.

2.6. Response Generation and Refinement

The Response Generator combines the agents' outputs into a single, fluent reply. This stage has three layers (illustrated as three stacked boxes in Figure. 1).

Response Templates & Prompts: For every dialogue-act/intent pair, our framework provides a small set of pre-defined prompt templates such as Explain-Risk, Recommend-Intervention, Confirm-Recommendation, and Explain-Change. The generator fills the template with slot variables such as *{evidence_snippet}*, *{biomarker_value}*, *{guideline_ref}*, then passes the result to the LLM for surface realisation. Drafting responses as slot-filled templates improves factual accuracy and ensures clinically-approved wording. Coupling the template with a refinement prompt yields responses that are more natural and less 'robotic' than strict slot-filled output. Integrating knowledge of a template taxonomy within the prompt then facilitates flexibility to resolve sensitive attributes [15].

Personalisation: A light post-processor adapts the draft response to the user's persona and the current situational context, both supplied by the ontology. Some examples are to:

- convert units or language variants *("mg/dL" → "mmol/L")*;
- reference recent episodic events *("Last week you completed Exercise Plan A.")*;
- adjust reading level *(≤8th-grade Flesch–Kincaid for cardiovascular health management of cancer survivors)*;
- adapt wording to accommodate accessibility needs *(replacing colour-based cues with shape or text descriptors for colour-blind users and simplifying visual references for low-vision readers)*;
- format the output for multimodal delivery *(automatic-speech-recognition (ASR) and text-to-speech pipelines so the same content can be voiced naturally when the user prefers audio interaction).*

Ethics & Compliance: Personalised drafts are finally assessed for classes of violations, such as: 1) Medical safety where every recommendation is cross-checked against the 2022 ESC Cardio-Oncology Guideline with missing evidence downgrades the advice to a cautionary statement [16]; 2) Hallucination filter to ensure all factual claims must map to a retrieved snippet or semantic triples, otherwise the sentence is removed; and 3) Privacy & tone where the draft is screened for Protected Health Information leakage, rude or biased language, and for lack of empathy (i.e., poor bedside manner). Responses must maintain a supportive, respectful tone in line with national AI-Act risk categories and patient-communication guidelines [17]. If any rule fails, the message is either redacted or returned to the LLM with a corrective instruction. Only text that passes the review stage is delivered to the user.

3. Evaluation

Having introduced the *i*CARE framework (in Figure 1), we focus this workshop paper's evaluation on a single question: *do ontology-derived features improve intent routing to specialised agents?*

For our experiments, we created a parallel corpus. Each example contains (i) a *raw* utterance spanning a set of intents; and (ii) a *disambiguated* counterpart forming the fully clarified intent with some ontology annotation cues (classes, relations, units, temporal tags). Typically, the fully formed intent utterances are gathered through the disambiguation planning step, which would interactively elicit missing slots (in reference to the ontology) via brief clarifying turns before passing on the clarified utterance for routing. In this study, we simulate those turns offline, by providing a disambiguated counterpart whose triples encode exactly the information that short clarification turns would have supplied based on the ontology. A few examples of the initial utterance (raw text), and its fully clarified intent (disambiguated text) obtained from the turn simulation process appear in Table 2.

The parallel corpus was created using a two-step prompting procedure. First, for each likely intent (aligned with the skeletal ontology in Figure 2), we asked an LLM (GPT-5) to generate 2–3 raw utterances that, owing to their generality, could span multiple intents and therefore correspond to multiple `:handledBy` ontology relations. Next, we re-prompted with the same raw text, the candidate intents, and the skeletal ontology (classes, relations) to produce a disambiguated counterpart containing minimal clarifying cues and triple annotations mapped to a single handling agent. For this study, we focused on 9 such agent classes (Exercise, Biomarker, Risk, Dietitian, Logger, Triage, Medication, Appointments, Privacy). The resulting parallel corpus was written to CSV with fields id, intent_id, agent, raw_text, disambig_text, and annotations. For simplicity, we assume a one-intent one-step planner, where each step is handled by a single agent. An intent router here classifies the given step with the utterance (raw or disambiguated) to one of the 9 agents. Extending this to multiple-intent plan steps and agent routing is left for future work. We assess and report dataset quality through inter-annotator agreement.

We evaluate the intent router under two conditions: (a) **Raw** (*raw_text* only); and (b) **Ontology-enriched** (*disambig_text* with ontology-derived cues). We compare three routing methods: **LLM-as-Router**; **ProtoKNN**; and a **KNN** variant. In our KNN variant, instead of aggregating same-class shots into a single prototype, the sampled shots themselves serve as candidate prototypes for similarity comparison; effectively a constrained KNN that ranks similarities only within the sampled shots rather than the full training set. This restricted variant is used primarily to study the potential of forming prototypes from randomly sampled shots per class. Both ProtoKNN and KNN use PubMedBERT embeddings to represent the *raw_text*, and *disambig_text* for similarity computations. For the LLM-as-Router, we evaluate GPT-4o (mini), Gemini 2.5 Flash, and LLaMA 8B Instruct. Evaluation results report routing accuracy on the parallel corpus (raw vs. ontology-enriched) using 4-fold cross-validation. In each fold, the held-out fold serves as the test set of utterances for classification, while the remaining folds are used to sample exemplar shots per class. Because the dataset provides only limited variation across classes (i.e., low class coverage), we restricted our experiments to 1-, 2-, and 3-shot settings. We do not assess downstream agent execution or response quality.

3.1. Analysing the quality of the parallel corpus

We used three independent annotators (blinded to the data generation prompt) to assess each row of the parallel corpus via two questions:

Q1 Disambiguation correctness (Yes/No): "Is the value in `Disambig_Text` the correct disambiguated version of the `Raw` utterance?"

Q2 Ontology coverage (Yes/No + Ontology figure): "Are all the concepts mentioned in `Disambig_Text` present in the ontology figure? If *No*, list the missing concepts."

Three assessors independently evaluated each binary (yes/no) item. Overall agreement, measured using Fleiss' κ, was **0.10**, indicating only slight agreement beyond chance. Given the low reliability, an independent meta-assessor reviewed all instances where at least one assessor disagreed (approximately

Table 2
Raw vs ontology-enriched question utterances (examples).

Raw (user text)	Ontology-enriched Disambiguated text
Is 6.7 bad?	**Text:** Is Total Cholesterol = 6.7 mmol/L high for me (21-year-old female)? **Triples:** Biomarker:TotalCholesterol | value | 6.7 mmol/L | | User:Maria | age | 21
where would 140 reading place me?	**Text:** With repeated elevated BP readings this month and past anthracycline exposure, what is my short term CV risk? **Triples:** Biomarker:SystolicBP | trend | elevated @month | | User: Maria | hasTreatmentType | Anthracycline
My chest feels tight.	**Text:** I am experiencing post-session muscular tightness after push-up exercises **Triples:** Symptom:ChestTightness | context | post-exercise | | Triage:red-flags | status | history

50% of the LLM-generated examples) and adjudicated the final fully clarified utterance with assistance from an LLM (ChatGPT-5) used solely for normalisation.

The LLM reformulated the contested fully clarified utterances to ensure they met four reproducibility criteria: (1) no advice or resolutions were included, (2) clarifications were added only to make the utterance interpretable, (3) the clarified utterance is mapped to a single target agent, and (4) the phrasing followed the corpus style of user statements with additional minimal ontology tags. The meta-reviewer then reviewed the LLM-normalised version and made the final accept directly, or with revision, decisions. The adjudicated utterances, together with their ontology annotations (minimal tags), were then used as the corrected gold standard in our experiments.

For Q2, when items were marked "No", annotators list concepts present in the disambiguated text but absent from the ontology figure. We normalise these strings (lowercase, punctuation removal, unit standardisation) and collapse common variants. We tally the canonicalised terms to produce a short list of out-of-ontology concepts. All of which we found to be useful suggestions and should serve as potential candidates for ontology expansion.

4. Results and Discussion

Across all results, ontology clarified utterances consistently led to superior performance. Further increasing number of support examples (i.e., the number of shots from 1 to 3) results in substantial improvement across the board, with approximately 15-20% accuracy gains observed for the disambiguated text results. For Raw texts, performance improvements with additional shots are modest and often inconsistent, indicating that the limited semantic clarity of the original utterances (in the absence of the ontology) constrains the models' ability to form well-separated routing class representations.

Comparing ProtoKNN with the KNN baseline (see Tables 4 and 3), we find the use of prototypes provides comparable accuracy results. With regards to latency, ProtoKNN achieved lower latency than KNN, with improvements growing from 0.3% in 1-shot to 6.1% in 3-shot settings. On average, ProtoKNN reduced latency by 3.4%. Note in our implementation both ProtoKNN and KNN baseline use only the randomly selected shots for local neighbourhood computation. With larger support sets and improved class coverage, prototype-based methods will offer efficiency gains by reducing similarity computations while maintaining or enhancing accuracy relative to KNN. Among the distance metrics, Mahalanobis performs best, especially with ProtoKNN, likely because in large, non-normalised vector spaces it better manages noisy feature directions (variance over prototype aggregation) and decorrelates dimensions. With KNN both Mahalanobis and Euclidean are comparable.

Table 5 presents the LLM-as-Router results, which clearly show performance gains of over 20% compared to ProtoKNN. Among the evaluated models, GPT-4o mini demonstrates strongest performance, achieving 92.2% accuracy with just 3-shots. The LLM-as-Router setup also allows us to explore Zero-Shot prompting, where no exemplars from the dataset are provided, offering insight into the models' in-built world knowledge relevant to the *i*CARE routing task. Notably, even in the Zero-Shot setting, GPT-4o mini achieves 76.56% accuracy, and this increases to 92.2% with the three shots. This clearly shows how additional domain evidence (i.e., ontology-disambiguated, fully clarified intents) enhances performance. Gemini 2.5 Flash's results are also noteworthy, achieving only about 3% lower accuracy than GPT-4o mini (with 3-shots), despite having no connection to the LLM family used for dataset generation or validation (in this case, we had used ChatGPT 5). These results also suggest that the use of an ensemble [18, 19] of LLM-as-Routers could be an interesting direction to explore, provided that latency costs are kept low. For instance, we found that compared to ProtoKNN, the LLM variants had latency increases of approximately 85–92%, with GPT-4o-mini, Gemini-2.5-flash, and LLaMA-3-8B all showing substantially higher inference times across all shot settings. This demonstrates the efficiency advantage of ProtoKNN for low-latency prediction. These results suggest the potential for a hierarchical hybrid approach, where ProtoKNN routing is invoked in regions with good evidential coverage, while the more computationally costly LLM router is employed otherwise.

Table 3
KNN Accuracy for Agent Classification (9 Classes) using PubMedBERT embeddings for vector representation

	Raw Text			Disambig_Text		
Shot	**Euclidean**	**Cosine**	**Mahalanobis**	**Euclidean**	**Cosine**	**Mahalanobis**
1-Shot	39.06	37.50	39.06	54.69	53.12	**56.25**
2-Shot	35.94	35.94	43.75	**59.38**	57.81	57.81
3-Shot	32.81	31.25	42.19	**70.31**	68.75	**70.31**

Table 4
ProtoKNN Accuracy for Agent Classification (9 Classes) using PubMedBERT embeddings for vector representation

	Raw Text			Disambig_Text		
Shot	**Euclidean**	**Cosine**	**Mahalanobis**	**Euclidean**	**Cosine**	**Mahalanobis**
1-Shot	39.06	37.50	39.06	54.69	53.12	**56.25**
2-Shot	39.06	35.94	37.50	60.94	59.38	**64.06**
3-Shot	43.75	43.75	37.50	64.06	64.06	**73.44**

Table 5
LLM Accuracy for Agent Classification (9 Classes)

	Raw Text			Disambig_Text		
Shot	**GPT 4o mini**	**Gemini 2.5 flash**	**LLama 8b Instruct**	**GPT 4o mini**	**Gemini 2.5 flash**	**LLama 8b Instruct**
Zero-Shot	64.06	67.19	65.62	**76.56**	67.19	73.44
1-Shot	70.31	67.19	65.62	79.69	**85.94**	78.12
2-Shot	71.88	70.31	62.50	**89.06**	84.38	78.12
3-Shot	73.44	70.31	60.94	**92.19**	89.06	73.44

5. Conclusion and Future Work

*i*CARE integrates lightweight knowledge to precisely capture intents, enabling LLMs to keep dialogues fluent yet factual within an agentic setup. The ProtoKNN approach offers a lightweight, easily deployable intent router that can operate offline, an important advantage for in-home care environments with limited connectivity. In contrast, the LLM-as-Router variants demonstrated higher accuracy in our simulated experiments, despite the limited dataset size. This contrast highlights the complementary strengths of symbolic and generative approaches and the potential for hybrids. A major challenge in this research area remains the scarcity of domain-specific data. Our experience indicates that even state-of-the-art LLMs do not produce fully reliable data without human oversight, although they serve as a valuable foundation for dataset generation. In our simulated corpus, each intent maps to a single agent handler, allowing a clear test of disambiguation effects on intent routing; extending this to many-to-many mappings is left for future work. Effective task decomposition will be essential to ensure that generative models follow accurate, well-structured routes. Future work could explore case-based planning for managing multiple intents and hybrid routing strategies [20] that combine low-latency, lazy-learning methods such as ProtoKNN with LLM-as-Router mechanisms when exemplar coverage is sparse.

Acknowledgments

This work was funded by the European Union's Horizon Europe programme under grant agreement No. 101213323. The views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Health and Digital Executive Agency or the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Declaration on Generative AI

The authors employed language technologies only for editorial assistance and dataset bootstrapping under human oversight: (i) ChatGPT (GPT-5) was used to generate a synthetic parallel corpus that was subsequently curated and evaluated via a human study; (ii) Grammarly in Overleaf for paraphrasing, re-wording, and grammar/spelling checks; and (iii) ChatGPT for minor improvements to writing style and clarity. No generative tool was used to create research content, analyses, figures, or citations. All text was reviewed and approved by the authors who take full responsibility for the publication's content.

References

[1] OpenAI, Introducing chatgpt, 2022. URL: https://openai.com/index/chatgpt/, accessed: 2025-07-23.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[3] G. Bansal, V. Chamola, A. Hussain, M. Guizani, D. Niyato, Transforming conversations with ai: A comprehensive study of chatgpt, Cognitive Computation 16 (2024) 2487–2510.

[4] P. Sahoo, P. Meharia, A. Ghosh, S. Saha, V. Jain, A. Chadha, A comprehensive survey of hallucination in large language, image, video and audio foundation models, in: EMNLP 2024, Association for Computational Linguistics, 2024, pp. 11709–11724.

[5] K. Hatalis, D. Christou, J. Myers, S. Jones, K. Lambert, A. Amos-Binks, Z. Dannenhauer, D. Dannenhauer, Memory matters: The need to improve long-term memory in LLM-agents, in: Proceedings of the AAAI Symposium Series, volume 2, 2023, pp. 277–280.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in neural information processing systems 33 (2020) 9459–9474.

[7] K. Bach, R. Bergmann, F. Brand, M. Caro-Martínez, V. Eisenstadt, M. W. Floyd, L. Jayawardena, D. Leake, M. Lenz, L. Malburg, D. H. Ménager, M. Minor, B. Schack, I. Watson, K. Wilkerson, N. Wiratunga, Case-Based Reasoning Meets Large Language Models: A Research Manifesto For Open Challenges and Research Directions, 2025. URL: https://hal.science/hal-05006761, working paper or preprint.

[8] K. Jin, H. H. Zhuo, Integrating ai planning with natural language processing: A combination of explicit and tacit knowledge, ACM Transactions on Intelligent Systems and Technology 16 (2025) 1–37.

[9] Y. Cheng, W. Liu, J. Wang, C. T. Leong, Y. Ouyang, W. Li, X. Wu, Y. Zheng, Cooper: Coordinating specialized agents towards a complex dialogue goal, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 17853–17861.

[10] F. Hengst, R. Wolter, P. Altmeyer, A. Kaygan, Conformal intent classification and clarification for fast and accurate intent recognition, in: Findings of the Association for Computational Linguistics: NAACL 2024, 2024, pp. 2412–2432.

[11] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, Advances in neural information processing systems 30 (2017).

[12] Y. Ukai, T. Hirakawa, T. Yamashita, H. Fujiyoshi, This looks like it rather than that: Protoknn for similarity-based classifiers, in: The Eleventh International Conference on Learning Representations, 2022.

[13] D. M. Manias, A. Chouman, A. Shami, Semantic routing for enhanced performance of LLM-assisted intent-based 5g core network management and orchestration, in: GLOBECOM 2024-2024 IEEE Global Communications Conference, IEEE, 2024, pp. 2924–2929.

[14] N. Wiratunga, R. Abeyratne, L. Jayawardena, K. Martin, S. Massie, I. Nkisi-Orji, R. Weerasinghe, A. Liret, B. Fleisch, CBR-RAG: case-based reasoning for retrieval augmented generation in LLMs for legal question answering, in: International Conference on Case-Based Reasoning, Springer, 2024, pp. 445–460.

[15] P. Salimi, N. Wiratunga, D. Corsar, A. Wijekoon, Towards feasible counterfactual explanations: A taxonomy guided template-based nlg method, in: ECAI 2023, IOS Press, 2023, pp. 2057–2064.

[16] A. R. Lyon, T. Lopez-Fernandez, L. S. Couch, R. Asteggiano, M. C. Aznar, J. Bergler-Klein, G. Boriani, D. Cardinale, R. Cordoba, B. Cosyns, et al., 2022 esc guidelines on cardio-oncology developed in collaboration with the european hematology association (eha), the european society for therapeutic radiology and oncology (estro) and the international cardio-oncology society (ic-os) developed by the task force on cardio-oncology of the european society of cardiology (esc), European Heart Journal-Cardiovascular Imaging 23 (2022) e333–e465.

[17] K. G. van Leeuwen, L. Doorn, E. Gelderblom, The ai act: Responsibilities and obligations for healthcare professionals and organizations, Diagnostic and Interventional Radiology (2025).

[18] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging LLM-as-a-judge with mt-bench and chatbot arena, in: 37th NIPS, NIPS '23, Curran Associates Inc., Red Hook, NY, USA, 2023.

[19] S. Es, J. James, L. Espinosa Anke, S. Schockaert, RAGAs: Automated evaluation of retrieval augmented generation, in: N. Aletras, O. De Clercq (Eds.), 18th EACL: System Demonstrations, ACL, St. Julians, Malta, 2024, pp. 150–158. URL: https://aclanthology.org/2024.eacl-demo.16/.

[20] D. Ding, A. Mallick, C. Wang, R. Sim, S. Mukherjee, V. Ruhle, L. V. Lakshmanan, A. H. Awadallah, Hybrid LLM: Cost-efficient and quality-aware query routing, arXiv preprint arXiv:2404.14618 (2024).