HACID - Deliverable

Demonstration in medical diagnostics

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101070588. UK Research and Innovation (UKRI) funds the Nesta and Met Office contributions to the HACID project.

Deliverable number:	D6.2
Due date:	28.02.2026
Nature[1]:	DEM
Dissemination Level[2]:	SEN
Work Package:	WP6
Lead Beneficiary:	Human Dx EU
Contributing Beneficiaries:	CNR, MPG

[1] The following codes are admitted:
- R: Document, report (excluding the periodic and final reports)
- DEM: Demonstrator, pilot, prototype, plan designs
- DEC: Websites, patents filing, press & media actions, videos, etc.
- DATA: Data sets, microdata, etc.
- DMP: Data management plan
- ETHICS: Deliverables related to ethics issues.
- SECURITY: Deliverables related to security issues
- OTHER: Software, technical diagram, algorithms, models, etc.

[2] The following codes are admitted:
- PU – Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)
- SEN – Sensitive, limited under the conditions of the Grant Agreement
- Classified R-UE/EU-R – EU RESTRICTED under the Commission Decision No2015/444
- Classified C-UE/EU-C – EU CONFIDENTIAL under the Commission Decision No2015/444
- Classified S-UE/EU-S – EU SECRET under the Commission Decision No2015/444

Document History

Version	Date	Description	Author	Partner
V1	09/01/2026	Created document draft	Vito Trianni	CNR
V2	02/02/2026	First draft of backend methods	Alessandro Russo	CNR
V3	05/02/2026	First draft of frontend methods	Vito Trianni	CNR
V4	11/02/2026	First draft of RAG-based aggregation	Weilai Xu	CNR
V5	12/02/2026	First draft of DKG-based aggregation methods	Chiara D'Onofrio	CNR
V6	19/02/2026	Revision and quality control	Gioele Barabucci	HDX
V7	26/02/2026	Final Version	Chiara D'Onofrio, Vito Trianni	CNR

Table of content

Document History **2**
Table of content **3**
1. Introduction **4**
2. Integration of backend technologies **4**
2.1. Reference scenario 5
2.2. General framework 6
2.2.1. Triplification and Case Knowledge Graph construction 8
2.2.2. Concept matching and linking to Domain Knowledge Graph 8
2.2.3. Collective aggregation and synthesis 9
2.3. Aggregation methods 9
2.3.1. DKG-based aggregation 10
2.3.2. RAG-based aggregation 12
3. Integration of frontend interaction elements **17**
3.1. Metacognitive reasoning 18
3.1.1. Subjective confidence estimation 18
3.1.2. Popularity of diagnoses 18
3.2. Exposure to diagnoses generated by diagnostic teams 19
3.3. Integrated chat with dedicated chatbots 20
4. Conclusions **21**

1. Introduction

Hybrid collective intelligence, combining human expertise with artificial intelligence, has emerged as a promising paradigm for addressing complex, open-ended decision-making problems. In the domain of medical diagnostics, hybrid human–AI approaches have demonstrated significant potential to substantially improve diagnostic accuracy. By aggregating and structuring independent diagnostic opinions from multiple clinicians and augmenting them with advanced computational methods, collective intelligence systems can support and improve clinical decision-making.

Within the general medical diagnostics domain, the HACID project has focused on the design, development, and validation of novel automated methods for aggregating independent diagnoses provided by human experts, with the goal of producing high-quality collective diagnoses. This approach aims at enhancing clinical reasoning by systematically leveraging diversity of human expertise and combining it with machine-supported reasoning mechanisms. To this end, HACID has designed and implemented a set of complementary methods and techniques that enable the representation, analysis, and aggregation of diagnostic knowledge, including information about the reasoning process performed by diagnosticians. These methods, described in detail in Deliverable D4.1, are grounded in several core components developed or consolidated within the project, including domain and case knowledge graphs integrating the SNOMED CT clinical terminology, aggregation algorithms leveraging term- and concept-level embeddings, graph-based reasoning and specialised retrieval-augmented generation (RAG) approaches powered by large language models (LLMs), and front-end features to elicit provision of metacognitive information.

Additionally, HACID has tested different means to leverage collective intelligence by providing feedback to expert users during the decision-making process. The experimentation devised for testing the value of integrating such social knowledge into the diagnostic process are described in detail in Deliverable D4.2. These experiments are grounded on frontend software features that enable the provision of social feedback, either in passive or in interactive modes.

The software demonstrator developed specifically for the HACID project showcases how these approaches could be or have been integrated with the HumanDx medical crowdsourcing platform. Through this integration, novel collective intelligence features are made available directly to human experts accessing the HumanDx platform via its mobile application, enabling seamless interaction between clinician-generated and AI-supported diagnostic processes.

2. Integration of backend technologies

This section is intended to support the understanding and future exploitation of the HACID backend technologies developed to illustrate hybrid collective intelligence in medical diagnostics. Specifically, Section 2.1 presents the reference demo scenario, while Section 2.2 provides a technical description of the scope and functionality of the demonstrator, outlines the end-to-end automated pipeline enabling collective diagnosis generation with

HACID technologies, and describes the integration and core functioning of the backend components involved. Finally, Section 2.3 focuses on the aggregation methods that have been implemented and integrated within the demonstrator framework.

2.1. Reference scenario

To understand, explain, and demonstrate how HACID technologies have been integrated with the Human Dx platform, we first provide an overview of the reference application scenario. The goal is to describe how the Human Dx platform works in general, outlining a typical user interaction flow, as well as the role of the platform in supporting collaborative diagnostics, and the role of collective intelligence capabilities in our demo scenario.

The Human Dx platform provides a collaborative diagnostic environment, accessible primarily via a mobile application, in which medical professionals can provide diagnostic support to their colleagues or patients, while improving their diagnostic and clinical reasoning skills. The platform supports a global community of users who can register, submit clinical cases and contribute their own diagnostic assessments. Through the Human Dx platform, users can propose clinical patient cases by specifying a set of findings, including symptoms, medical history, physical examination results, imaging, laboratory values, and other relevant clinical information. To ensure clinical quality and educational value, submitted cases are reviewed and published only after approval by an editorial board of licensed medical professionals. Once a case is approved and made public on the platform, it becomes available to other users, who can then attempt to solve it by submitting their own diagnoses.

Each clinical case is presented to users in the app as a so-called “vignette”, a concise, structured clinical case description designed to mimic the information flow encountered in real-world medical practice (see Figure 1). The vignette contains patient information and clinical findings as provided by the case creator, such as presenting symptoms, prior medical records, and results of diagnostic tests. A typical usage pattern involves incremental disclosure of case findings. As the user progresses through a case, clinical information is revealed step by step, simulating the gradual availability of new knowledge during patient assessment. After each step the user can submit their diagnostic solution.

When solving a case, users can submit either a single diagnosis or a ranked list of possible diagnoses, commonly referred to as a *differential diagnosis*. Diagnoses can be entered as free text or selected from a structured medical taxonomy supported by an auto-complete feature that assists users as they type. After each incremental disclosure of new clinical findings, users have the opportunity to revise or resubmit their differential diagnosis by adding new hypotheses, removing unlikely options, or re-ranking diagnoses based on the newly available evidence. This iterative process continues until all findings have been revealed and the user submits a final diagnostic solution for the case.

Once a user submits her final solution, the platform provides feedback depending on the type of case (see Figure 1). For training cases, where the correct diagnosis is known because it has been specified by the case creator, the system presents feedback showing how the user's differential diagnosis compares with the reference solution. In addition, the platform can combine the user’s submitted solution with the solutions provided by other users who have previously solved the same case. Using collective intelligence methods, these individual diagnostic opinions are synthesised into a single cohesive result, referred to as a *collective synthesis* or *aggregated solution*. This collective synthesis represents the platform’s collective diagnostic assessment for the case.

The key enabler for the integration of HACID technologies in the Human Dx platform is the extension of the platform to support additional external collective intelligence services. In the demonstration scenario, collective synthesis capabilities are provided by a backend platform powered by HACID components, as explained in the following sections.

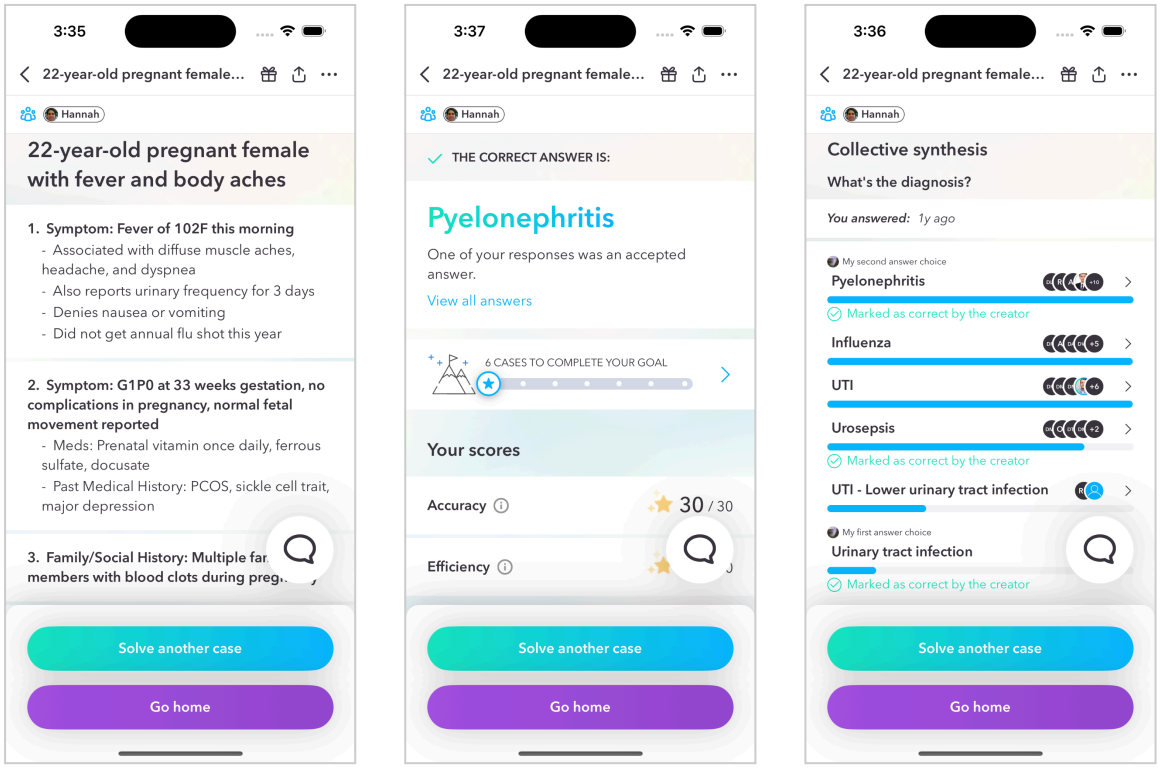

Figure 1. Example of a case displayed by the Human Dx platform. Left: presentation of information about the case, where title and case findings are visible. Center: feedback about the correct answer available for the given case. Right: visualization of the collective synthesis, aggregating the most frequent diagnoses provided by the community, ranked in order of popularity.

2.2. General framework

To better understand how HACID technologies are integrated with the Human Dx platform, it is useful to briefly examine the underlying data flow and processing steps.

Figure 2 provides a representation of the data flow between Human Dx platform and HACID technologies. While a user is solving a case, the mobile application sends intermediate differential diagnoses to the Human Dx backend platform, where they are stored for later analysis. When the user submits her final solution, the platform triggers a series of backend computations. The most relevant of these is the computation of an updated collective synthesis that incorporates not only previously submitted solutions but also the newly provided diagnosis. In addition, several other metrics and derived indicators may be computed by the platform.

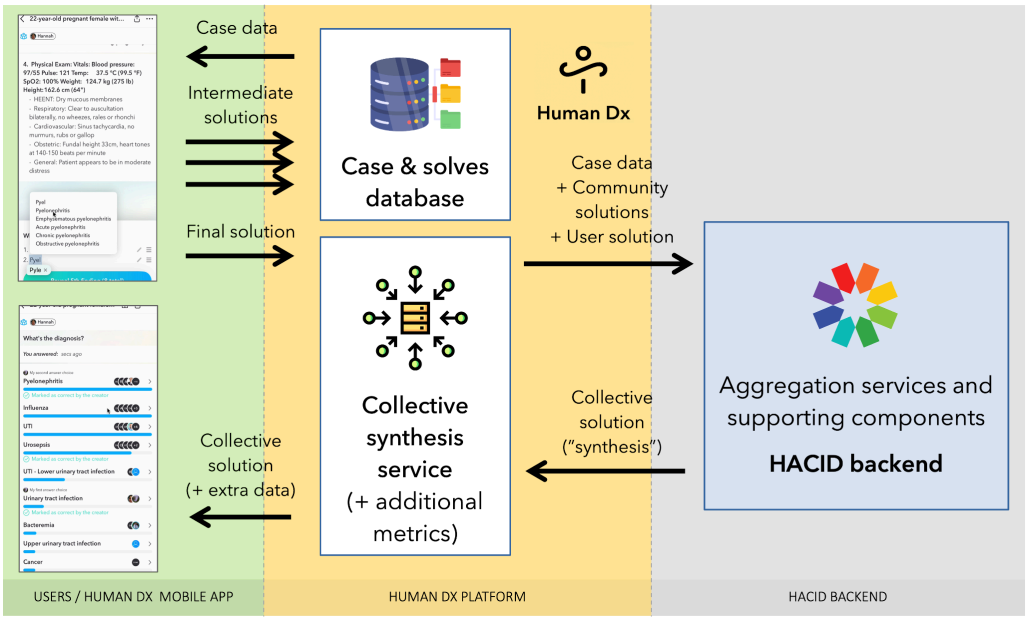

Figure 2. Schema of the framework to integrate HACID technologies in the Human Dx platform.

To perform collective synthesis, the Human Dx platform relies on an internal dedicated collective synthesis service. This service can be configured to delegate the computation of the aggregated collective solution to an external system (for example, the one developed by HACID in Work Packages 2, 3, and 4, as then described below). Such an external service receives the relevant case data and the set of user-provided solutions and returns an integrated collective diagnosis generated using its internal collective intelligence and aggregation methods. The results produced by this service are then received by the Human Dx platform, combined with other computed metrics, and presented back to the user in the mobile application in a transparent and easily understandable manner.

In our demo scenario, when a user submits her final solution to a case, the Human Dx platform forwards the case data together with the diagnostic solutions submitted so far to a backend service powered by HACID technologies. The HACID backend is thus responsible for processing this input, integrating it with existing domain knowledge, and computing an updated collective diagnostic solution using the hybrid collective intelligence methods developed in the project. Specifically, the collective synthesis is computed using HACID components and technologies, including:

- the Domain Knowledge Graph[3] (DKG) built on SNOMED CT;
- the Case Knowledge Graph[4] (CKG) built for the specific case; and
- a set of automated collective intelligence and aggregation algorithms.

[3] The **Domain Knowledge Graph** (DKG) is the relatively static, domain-level knowledge graph that encodes shared, structured medical knowledge, primarily standardised concepts and their semantic relations as defined in SNOMED CT. It serves as the foundational reference knowledge source from which relevant concepts and relationships are drawn to support case analysis, reasoning, and aggregation.

[4] The **Case Knowledge Graph** (CKG) is a dynamically generated, case-specific knowledge graph constructed from the available case data (including its description, associated clinical findings, and the diagnostic attempts and differential diagnoses provided by users) and further enriched by selecting, linking, and contextualizing relevant concepts from the DKG. It provides a formal representation of the knowledge considered pertinent to the specific case and supports automated reasoning and the aggregation of diagnostic solutions.

The resulting collective diagnosis is then returned to the Human Dx platform and made available to users through the standard application interface.

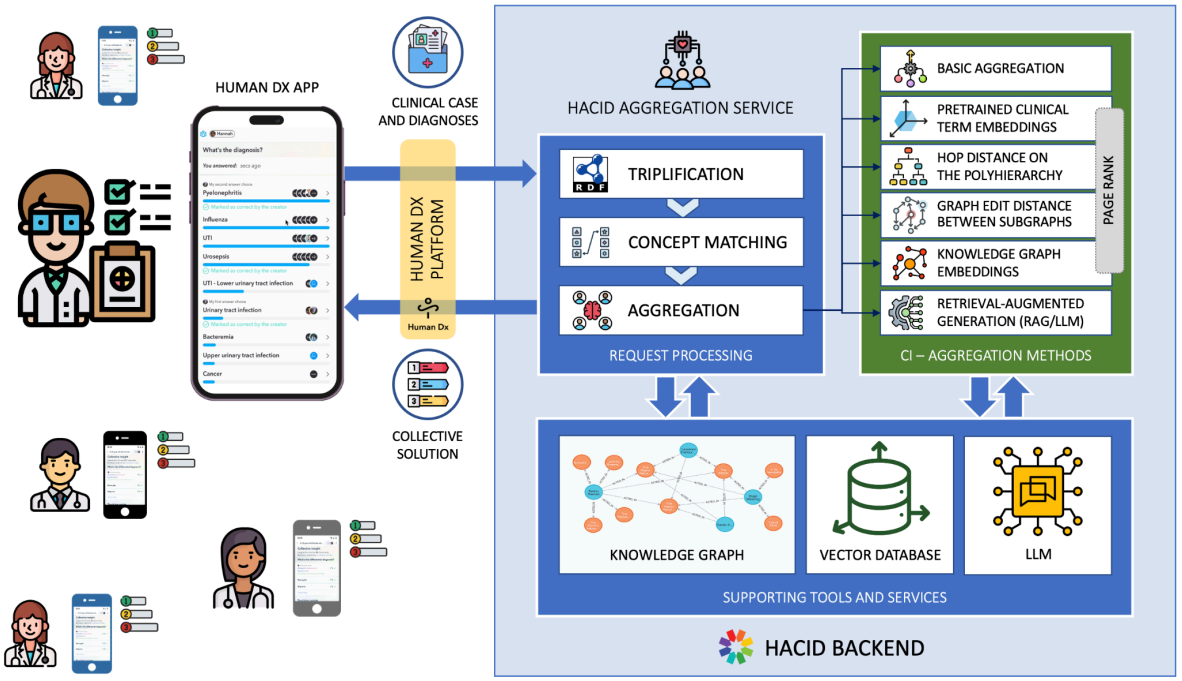

Figure 3. Graphical representation of the main HACID technologies integrated in the general framework.

From a high-level perspective, the request received from the Human Dx platform is processed through a sequence of coordinated steps that together constitute the collective synthesis pipeline, as described in the following sections (see also Figure 3).

2.2.1. Triplification and Case Knowledge Graph construction

The first responsibility of the HACID backend is to represent the incoming information in a structured and semantically grounded form. Case data provided by the Human Dx platform, together with the ranked lists of diagnoses submitted by the users who solved this case in the past and by the current user, are transformed into a set of RDF triples according to the reference ontologies defined within the project (and presented in detail in Deliverables D2.1 and D2.2) for representing clinical case information and diagnostic attempts.
The RDF triples produced by this triplification step are used to build or incrementally enrich a Case Knowledge Graph that essentially captures the clinical case and its associated findings (as described in the clinical vignette), as well as the individual diagnostic attempts submitted by users, with the ranked structure of differential diagnoses.

2.2.2. Concept matching and linking to Domain Knowledge Graph

During the construction or extension of the Case Knowledge Graph, the diagnoses provided by users (i.e., the entries in their ranked differential diagnosis lists) are semantically aligned with standardized medical concepts. Specifically, each diagnosis is matched, when possible, to corresponding concepts in the SNOMED CT medical ontology, which forms a core

component of the HACID Domain Knowledge Graph, as detailed in Deliverables D2.1 and D2.2.
The entity linking step follows the approach outlined in Deliverables D2.1 (cf. Section 3.3) and D4.1 (cf. Section 2.1). In brief, diagnostic labels are first normalized through standard text preprocessing procedures for clinical terminology (e.g., lowercasing, removal of non-essential variations, etc.) to ensure consistent comparison. The normalised labels are then matched against the normalised labels associated with concepts in SNOMED CT. In its basic form, this procedure relies on exact lexical matching: a link is established when an exact correspondence is identified, operationalised through a Jaccard similarity measure indicating full overlap between the compared label representations. When such a matching fails, we exploit clinical term embeddings stored in a vector database, which return the closest concept in the embedding space according to the cosine similarity.
This semantic matching step establishes explicit links between case-specific diagnostic hypotheses and domain-level medical knowledge. As a result, diagnostic entries expressed in different textual forms by different users can be normalized to shared clinical concepts, enabling consistent aggregation and comparison. Moreover, the linkage between the CKG and the DKG allows HACID aggregation algorithms to exploit hierarchical relations, semantic similarity, and other medically meaningful relationships encoded in SNOMED CT and captured in the corresponding knowledge graph.

2.2.3. Collective aggregation and synthesis

Once the CKG has been constructed or updated and appropriately linked to the DKG, the HACID backend proceeds with the computation of the collective diagnostic solution. Depending on the configuration of the backend service, one of the available aggregation approaches developed within HACID is selected and triggered. These aggregation approaches operate on the structured knowledge graph to retrieve case data and users' differential diagnoses, and then exploit different combinations of graph-based metrics, embedding-based similarity measures, and knowledge-driven retrieval and generation mechanisms.
Depending on their internal logic, the aggregation methods may further access the DKG to retrieve specific contextual or semantic information and may rely on additional supporting services and tools, ranging from vector databases for embedding-based similarity evaluation to Large Language Models for retrieval-augmented generation (RAG)–based aggregation strategies; additional details on these mechanisms are provided in the next section.
The output of this aggregation step is a collective solution, expressed as a ranked list of diagnostic concepts representing the synthesised diagnostic assessment of the users, which is then returned to the Human Dx platform and finally presented to the user through the standard application interface.

2.3. Aggregation methods

Aggregation methods constitute the core functional components of the HACID backend and are responsible for producing collective diagnostic solutions from the set of individual diagnoses submitted by users for a given clinical case. Each aggregation method implements a specific strategy for combining, weighting, and synthesizing users' diagnostic inputs into a single collective outcome, with the objective of leveraging both the diversity of human expertise and the structured medical knowledge available in the system.

The HACID backend provides a library of aggregation algorithms, each reflecting a distinct collective intelligence approach. These approaches differ in terms of the representations they operate on and the degree to which they rely on structured domain knowledge, embeddings, or generative reasoning. As described in the reference demonstration scenario, the HACID backend is configured to use a specific aggregation method when producing collective solutions for a case. This configuration enables the demonstrator to showcase different aggregation strategies while preserving a consistent integration with the Human Dx platform and ensuring transparency for end users.
The various aggregation approaches developed within HACID are described in detail in Deliverable D4.1. In addition, we provide here also a description of the RAG-based aggregation method that was not included in D4.1 (see Section 2.3.2). For the sake of completeness and understandability, each aggregation method is summarised in the next subsections.

2.3.1. DKG-based aggregation

For open-ended questions, the most intuitive way to combine individual judgments into a single collective solution is to select the most frequently proposed diagnostic judgment—a procedure known as *plurality voting*. We explored a novel method for improving collective outcomes beyond simple vote aggregation: weighting each judgment based on both its position in a participant's list and its similarity to other proposed judgments.
In this section we first describe a baseline aggregation method, where the DKG is used solely for detecting concept equivalence, without leveraging the richer semantic relationships embedded within the graph structure—such as hierarchical links or contextual similarities—which could further enhance the generation of a collective solution. We then extend this baseline by incorporating these conceptual relationships from the DKG into the scoring mechanism used to rank collective solutions. Specifically, we show various methods to compute the semantic similarity between diagnoses. The resulting similarity measure contributes to the overall score assigned to each nominated concept by factoring in its semantic proximity to other candidate concepts.
We refer to Deliverable D4.1 for detailed descriptions of the specific models and techniques used for knowledge graph embeddings and clinical term embeddings, as well as for precise definitions of the similarity functions.

Baseline aggregation

To compare different scoring functions based on item ranks, the similarity weight can be defined as a binary function, returning 1 when comparing identical concepts and 0 otherwise. In this setting, a concept's score increases only when it is explicitly nominated by multiple users, rather than when similar concepts are nominated.
Setting the rank weight to 1 corresponds to simple plurality voting, where all nominated concepts contribute equally to the collective solution regardless of their rank in the differential diagnosis. We also tested rank-biased scoring rules, showing that applying a reciprocal rank significantly improves diagnostic performance compared to equal weighting. This aligns with the intuition that physicians rank diagnoses according to their estimated diagnostic probability.

Similarity-based aggregation

Building on the baseline aggregation algorithm described above, we exploit semantic similarity to rank the diagnoses provided by different experts. To incorporate semantic relationships between nominated concepts into the aggregation process, specifically within the scoring mechanism that actually generates the collective solution, we proceed as follows: (i) for each pair of nominated concepts, we compute their semantic distance using a selected metric, (ii) we transform these distances into semantic similarities through a kernel function (e.g., a Gaussian kernel), (iii) we integrate these similarity values into the final score assigned to each concept by weighting them according to both their similarity to other nominated concepts and their position within the differential diagnosis in which they were mentioned (for more details, see Deliverable D4.1). This approach allows semantically related diagnoses to reinforce one another, even when expressed using different but closely related concepts.

We have tested different distance metrics in support of the similarity-based aggregation approach, that directly use the DKG structure or exploit vectorial embeddings, possibly trained on the DKG.

- **Hop distance on the polyhierarchy** — Our DKG encodes a wide range of semantic relationships between medical concepts, including those derived from SNOMED CT. One of the most frequent and significant is the “is-a” relationship, represented by the “broader” property in our ontology model and DKG. This relationship forms a polyhierarchy that organizes clinical concepts by specificity, linking more granular concepts to broader categories through subject–relation–object triples. A straightforward way to measure distance within this polyhierarchy is to consider the number of hops along the shortest path between two concept nodes. Concepts that are directly connected or separated by only a few intermediate nodes are considered semantically closer than those that are farther apart in the hierarchy.
- **Tree edit distance between subgraphs** — In this section, we describe a distance metric based on structured subgraphs derived from the DKG. For each solution diagnosis, we extract a subgraph centered on the diagnosis concept, including its immediate neighbors (1-hop) and their connections (2-hop). This subgraph captures key semantic properties such as: (a) *isDescribedBy*, linking the concept to its description nodes, (b) relationships such as *hasInterpretation*, *findingSite*, *associatedMorphology*, *pathologicalProcess*, *causativeAgent*, connecting descriptions to relevant concepts, (c) *type*, indicating semantic categories (e.g., disorder, clinical finding, organism, morphologically abnormal structure), (d) *broader*, linking the concept to parent nodes in the SNOMED CT polyhierarchy. To quantify the distance between two diagnoses, we represent their subgraphs as trees—with the central concept as the root, 1-hop neighbors as children, and 2-hop distant concepts as leaves. We then compute the Tree Edit Distance (TED), which measures the minimum number of edit operations (node substitutions, insertions, deletions) needed to transform one tree into the other.
- **Vectorial distance between knowledge graph embeddings** — In addition to the “broader” relationship that constitutes the backbone of the polyhierarchy, the DKG includes a variety of other relationship types that can be used to further enrich our analyses. To capture these additional semantic links, we use Knowledge Graph Embeddings (KGEs). KGEs project entities and relations from the knowledge graph into a continuous vector space while preserving structural and semantic relationships. Each entity (e.g., a specific disease or body part) and each relation (e.g., “broader” or “finding

site") is represented as a vector, allowing embedding algorithms to learn latent patterns such as hierarchy or transitivity. These learned representations can then be used for downstream tasks such as measuring semantic distance or clustering related concepts, while retaining the original graph's informative structure.

- **Vectorial distance between pre-trained clinical term embeddings** — Another approach to calculating distances between medical concepts, in addition to the knowledge-graph-based approaches just described, are pretrained sentence transformer embeddings. Pretrained sentence transformer embeddings can assess the semantic distance between diagnostic terms and enhance the aggregation process. Clinical term embeddings encode medical concepts as numerical vectors, capturing both semantic meaning and contextual relationships. In this vector space, semantically related terms are positioned closer together. We therefore exploit the distance between embedding vectors as a basis for the aggregation mechanism.

PageRank aggregation

As an alternative to directly using the similarity coefficients to compute relevance scores, a graph of nominated concepts can be built, where nodes represent concepts and edge weights are provided by similarity metrics. The PageRank algorithm, and specifically its personalized variant, is then used to identify the most important nodes in the graph (and extract a ranking). *Personalized PageRank* allows weighting specific nodes higher via a personalization vector, providing a ranking of concepts that incorporates both nomination frequency and semantic relatedness.

2.3.2. RAG-based aggregation

The RAG-based aggregation method implements a retrieval-augmented generation approach that combines large language models with structured knowledge retrieval from the SNOMED CT Domain Knowledge Graph. RAG augments the LLM with task-specific evidence retrieved at inference time, so the model can generate diagnoses that are explicitly grounded in the provided clinical knowledge rather than relying only on its internal parameters pre-trained on the universal tasks. Unlike the aggregation methods described above, this approach enriches the collective diagnostic process enabling the LLM to both aggregate the diagnoses proposed by the experts, and by generating new diagnoses grounded onto the available medical knowledge for the case.

The aggregation workflow proceeds through four sequential stages: 1) concept retrieval, 2) triple retrieval, 3) diagnostic aggregation, and 4) entity linking (see Figure 4). Each stage builds upon the outputs of previous stages, progressively transforming raw clinical case data and expert opinions into a structured, knowledge-grounded collective diagnosis.

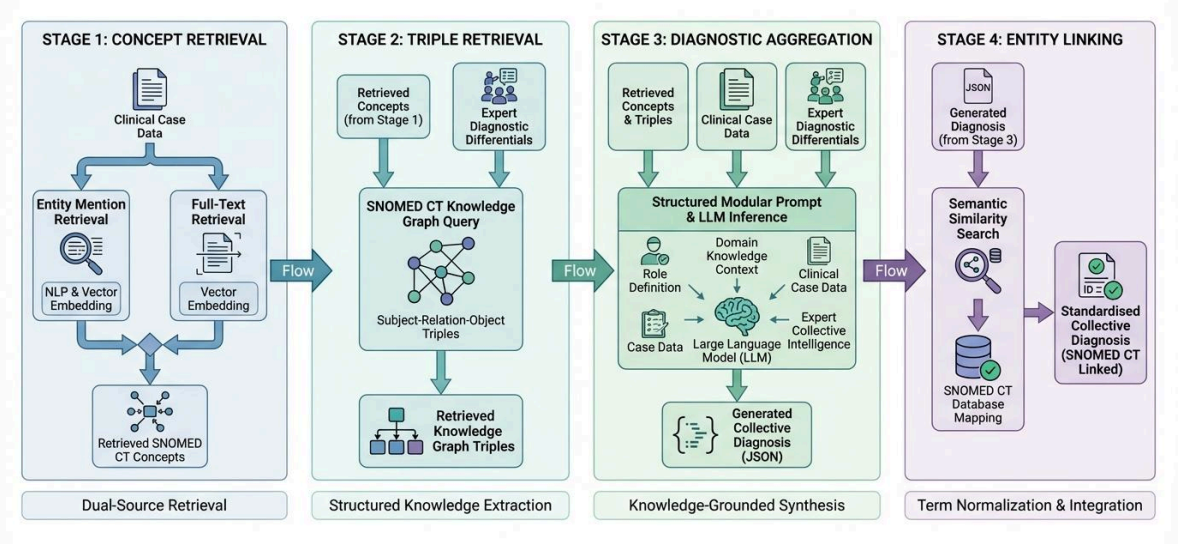

Figure 4. The workflow of the RAG-based aggregation approach.

Stage 1: Concept Retrieval

The first stage retrieves relevant SNOMED CT concepts from the clinical case description by using the so-called “entity mention” and “full text” methods, described in the rest of this section. This dual-source retrieval strategy combines fine-grained entity-level precision with broader case-level contextual coverage, ensuring that both explicitly mentioned clinical entities and implicit semantic patterns of the case are captured.

- **Entity Mention Retrieval**: For the clinical case description, the system first performs biomedical named entity recognition to identify explicit medical entities mentioned in the text. A light, specialised natural language processing model trained on biomedical literature extracts entity spans such as symptoms, anatomical locations, procedures, and conditions. To ensure high-quality extraction, the system filters entities to retain only those containing substantive medical nouns while excluding overly generic terms like "disorder" or "finding" that provide limited discriminative value.

  Each extracted entity is converted into a numerical vector representation using the same embedding model that is used for building the vector database. The embedding model generates 768-dimensional vectors, which are then compared against a vector database containing embeddings of all SNOMED CT concepts, retrieving the single most similar concept for each entity based on cosine similarity.

  This entity-level retrieval strategy directly links mentioned clinical phenomena to standardised medical concepts, providing precise grounding for explicitly stated case information. By retrieving only the top match for each entity, the system maintains high precision while avoiding noise from weakly related concepts.
- **Full-Text Retrieval**: Complementing entity-level retrieval, the system also performs full-text retrieval by encoding the entire case description as a single semantic unit. This captures implicit medical context, symptom patterns, and clinical narratives that may not be explicitly identified as discrete entities but nonetheless carry diagnostic significance.

  The complete case text is processed through the same biomedical embedding model to generate a single 768-dimensional vector representing the overall clinical presentation. This vector is compared against the entire SNOMED CT concept

database, retrieving the top five most semantically similar concepts regardless of whether they were explicitly mentioned in the case. Full-text retrieval typically surfaces broader diagnostic categories, clinical contexts, or associated conditions that provide valuable background knowledge for reasoning about the case.

The system combines entity-level and full-text retrieval results into a unified set of retrieved concepts, with explicit entity matches given higher priority than full-text matches to ensure that directly stated clinical information takes precedence over inferred context. Each retrieved concept consists of its fully specified name in SNOMED CT terminology and its unique concept identifier, both of which are used in subsequent retrieval stages.

Stage 2: Triple Retrieval

With relevant SNOMED CT concepts identified, the second stage retrieves structured relationship triples from the Knowledge Graph to provide explicit medical knowledge about these concepts. Triple retrieval occurs from two distinct sources: concepts retrieved from Stage 1, and concepts present in expert diagnostic differentials. The latter are already matched to SNOMED CT concepts, and therefore can be directly used to retrieve relevant triples.

- **Retrieval from Case Concepts**: For each concept retrieved from the case description in Stage 1, the system queries the DKG to extract relationship triples that describe medical properties, hierarchical classifications, and semantic associations of that concept. These queries retrieve triples in the form of subject-relation-object statements, where both subject and object are SNOMED CT concept terms.
  Retrieved triples are filtered and prioritised based on their relevance to the retrieved concept set. Triples where both the subject and object match retrieved concepts receive highest priority, as they represent explicit relationships between elements of the clinical case. Triples where only one endpoint matches a retrieved concept are included up to a configured limit, providing extended context without overwhelming the system with loosely related information. This filtering strategy ensures that the knowledge provided to the language model remains focused on the specific clinical scenario while maintaining sufficient breadth to support comprehensive reasoning.
- **Retrieval from Diagnostic Differentials**: In addition to case-derived knowledge, the system retrieves triples for concepts appearing in expert diagnostic differentials. This retrieval provides structured medical knowledge about diagnoses that experts have proposed, enabling the language model to evaluate these proposals against established clinical relationships and case evidence.
  For each unique concept identifier present in the diagnostic differentials, the system performs targeted knowledge graph queries similar to those used for case concepts. However, rather than prioritising triples based on overlap with case concepts, differential-based retrieval focuses on capturing comprehensive knowledge about each proposed diagnosis up to a configured maximum per diagnosis. This ensures that the system has sufficient information to assess the validity of expert suggestions without excessive context length that could impair language model performance.

The triples retrieved from case concepts and diagnostic differentials are combined into a unified knowledge base, with duplicate triples removed to avoid redundancy. The system maintains separate tracking of triple sources (case-derived versus differential-derived) to enable transparency about knowledge provenance and to support prompt engineering strategies that distinguish evidence-based knowledge from hypothesis-specific knowledge.

Stage 3: Diagnostic Aggregation

The third stage constitutes the core synthesis process, where the system generates a collective diagnostic solution. This is achieved by feeding the gathered information into an LLM via a structured, modular prompt. The design of this stage emphasises the integration of clinical evidence, expert opinions, and retrieved domain knowledge to guide the model towards a highly accurate and consistent differential diagnosis.

```
You are a medical expert providing a differential diagnosis for a clinical case.
You will be provided with:
— a detailed description of the case
— a list of differential diagnoses provided by medical experts. Each differential
is a list of diagnoses sorted by probability of being the correct one, most
probable first.
— relevant SNOMED concepts and triples to help ground your diagnosis.

Your task is to:
1. Reflect upon the case description, meticulously evaluating every piece of
evidence available. Ensure each piece of evidence is assessed correctly, offering
a balanced view of its implications.
2. Reflect upon the differential diagnoses provided by medical experts,
evaluating how they link to the evidence available from the case description.
3. Formulate a differential diagnosis as a list of hypotheses.


Retrieved SNOMED concepts:
{retrieved_concepts}

Retrieved SNOMED triples:
{retrieved_triples}

Take your time to think through each piece of evidence step-by-step. Consider all
aspects of the case description, and consider all the suggestions provided by
medical experts as a starting point.
Check that each diagnosis in your answer is consistent with each finding in the
case description.
Be as concise as possible, no need to be polite. Provide only the most probable
differential diagnosis, no explanation, no rationale, no recapitulation of the
case information or task.
Give {top_n} diagnoses, sorted by probability of being the correct diagnosis,
most probable first.
In your answer, provide only the appropriate SNOMED CT fully specified name (FSN)
of each diagnosis, no id.

Return STRICT JSON:
{output_format}

Case description:
{case_description}

{diagnostic_differentials}

IMPORTANT: Do NOT only rely on the suggestions from diagnostic differentials, but
use them to inform your own independent analysis of the case. Make sure to
evaluate the case description thoroughly and critically, and to check that all
{top_n} diagnoses in your answer are consistent with each finding in the case
description.
```

Figure 5: Prompt schema exploited for the RAG-based aggregation

Prompt Construction

The system constructs a comprehensive prompt designed to act as a "virtual medical expert" (see Figure 5). This prompt is structured into four distinct functional blocks, each serving a specific purpose in guiding the LLM's reasoning process:

1. **Role Definition and Task Instruction:** The prompt begins by establishing the persona of the model as a "medical expert" and clearly defining the objective: to formulate a differential diagnosis based on provided evidence. It explicitly instructs the model to perform a multi-step reasoning process: first, to meticulously evaluate every piece of evidence in the case description; second, to critically reflect on the diagnoses provided by human experts; and finally, to synthesise these inputs into a ranked list of hypotheses.
2. **Domain Knowledge Context (Retrieved Concepts and Triples):** This section injects the structured knowledge retrieved in the previous stages. It presents relevant SNOMED CT concepts and knowledge graph triples (Subject-Relation-Object). By determining the relationships between symptoms and conditions (e.g., finding site, associated morphology), this block grounds the model's generation in verified medical facts, reducing hallucinations and ensuring terminological precision.
3. **Clinical Case Data:** The full clinical vignette is embedded directly into the prompt. This includes the patient's history, physical exam findings, and test results. The placement of this data ensures that the model has access to the raw clinical evidence necessary to validate or refute potential diagnoses.
4. **Expert Collective Intelligence (Diagnostic Differentials):** The prompt incorporates the list of differential diagnoses submitted by human experts from the Human Dx platform. Crucially, the prompt frames these not as definitive answers, but as "suggestions" or "starting points" to be evaluated. A specific directive warns the model to leverage the "wisdom of the crowd" while maintaining the autonomy to correct errors or identify overlooked possibilities.

Modular Design for Experimental Flexibility

A key feature of this prompt design is its modularity. The system architecture allows for dynamic inclusion or exclusion of specific blocks—such as the retrieved knowledge graph triples or the expert differentials—without altering the core instruction structure. This flexibility enables the execution of ablation studies (e.g., comparing performance with vs. without RAG support, or independent AI vs. hybrid Human-AI aggregation) to rigorously evaluate the contribution of each information source to the final diagnostic accuracy.

Language Model Inference and Response Parsing

Once constructed, the prompt is submitted to an instruction-tuned Large Language Model[5]. The model operates as a black-box reasoning engine, processing the complex interplay between the unstructured clinical text, the structured expert rankings, and the symbolic domain knowledge.

The inference process focuses on generating a structured output that strictly adheres to a predefined JSON schema. The prompt explicitly commands the model to be concise and to provide only the requested number of top diagnoses (e.g., 5), sorted by probability. It further enforces the use of SNOMED CT Fully Specified Names (FSN) for all generated diagnoses.

[5] In the demonstration, we use mistralai/Ministral-3-14B-Instruct-2512 as the core LLM, offering a practical trade-off between performance and inference latency.

Upon receiving the raw text response from the LLM, the system employs a robust parsing mechanism to extract the JSON object. This parser handles potential formatting irregularities in the model's output, ensuring that the generated diagnoses are correctly interpreted as a structured list of ranked candidates, ready for the final entity linking stage. This streamlined inference and parsing pipeline ensures that the rich, free-form reasoning of the LLM is converted into a standardised format interoperable with the downstream medical platform.

Stage 4: Entity Linking

The final stage performs entity linking to map the language model's generated diagnosis terms to canonical SNOMED CT concepts. While the prompt instructs the model to use SNOMED CT fully specified names, language models may produce variant terminology, colloquial expressions, or closely related terms that require normalisation for interoperability with the HumanDX platform.

For each diagnosis in the LLM output, the system extracts the diagnosis text and performs a semantic similarity search against the SNOMED CT concept database. The diagnosis text is encoded using the same biomedical embedding model employed in Stage 1. This vector is compared against embeddings of all SNOMED CT concepts to identify the single most similar standardised term.

The entity linking process retrieves not only the matched concept's fully specified name but also its unique SNOMED CT identifier and concept type (such as disorder, procedure, or finding). These linked identifiers enable seamless integration with the HumanDX platform's data model and ensure consistency with expert-provided differentials that use the same standardisation approach.

The system constructs the final aggregation result by combining each diagnosis's original language model text with its linked SNOMED CT term and identifier. This dual representation preserves both the model's reasoning output and its standardised interpretation, supporting transparency and enabling quality assessment. Additional metadata including confidence scores, reasoning summaries when available, and information about retrieved concepts and triples are included to provide complete provenance of the aggregation process.

This structured result is returned to the HACID backend aggregation service for transmission to the HumanDX platform, where it is presented to users as the collective synthesis of expert opinions augmented by artificial intelligence and structured medical knowledge.

3. Integration of frontend interaction elements

The exploitation of hybrid collective intelligence requires methodologies to maximise information retrieval from users, as well as to provide feedback to human experts during the decision making process. Beyond differential diagnoses, users can provide information useful to evaluate the diagnostic process, known as metacognitive reasoning. Feedback can come from either other humans—as in standard collective intelligence approaches—or from AI systems, and can take different forms, from simple advice to more structured deliberation methods. Within WP4, we have studied how additional information and feedback should be integrated into the decision making process with different experimental studies, carried out within the Human Dx platform, as described in Deliverables D4.1 and D4.2. To make such studies possible and to enable social feedback during the diagnostic process carried on the Human Dx platform, several frontend features have been developed and tested. These new

frontend features, their scientific foundations, and how they integrate with the user experience of Human Dx, are presented in the rest of this section.

3.1. Metacognitive reasoning

Metacognitive reasoning refers to the capacity of individuals to reflect on, monitor, and regulate their own cognitive processes. In diagnostic contexts, it encompasses awareness not only of *what* conclusion is reached, but also of *how* that conclusion was formed and how reliable it may be. This second-order reasoning is critical in medical decision-making, where uncertainty is inherent and where explicit representations of reasoning quality can be leveraged to improve both individual and collective performance. Within HACID, we focused on two different metacognitive abilities: (i) the estimation of the confidence of the proposed diagnosis, a necessary information for confidence-slating approaches in collective intelligence,[6] and (ii) the estimation of the popularity of diagnoses, to be used for the implementation of the *surprisingly popular* voting method.[7]

3.1.1. Subjective confidence estimation

Subjective confidence estimation is a central component of metacognitive reasoning in medical diagnostics. It denotes a diagnostician's ability to assess and report how likely their diagnosis is to be correct. Such confidence judgments provide information beyond the diagnostic label itself, enabling approaches such as confidence-slating, in which diagnoses are weighted, selected, or combined based on expressed certainty. Accurate confidence calibration—where higher confidence corresponds to higher accuracy—is especially important, as miscalibration can undermine the benefits of confidence-aware aggregation.
To enable subjective confidence estimation within the Human Dx platform, the UX has been modified to include a specific question about confidence after the diagnosis has been completed. We have decided to (i) completely separate the submission of differential diagnoses from the confidence question, and (ii) to limit the subjective confidence estimate to a single question about the whole differential, instead of a question for each diagnosis in the provided differential. In this way, we obtain a simple but still informative estimation. The responses are organised as a 4-level Likert scale. The left panel in Figure 6 shows a screenshot of the confidence estimation question in the Human Dx platform.

3.1.2. Popularity of diagnoses

The estimation of the popularity of diagnoses represents another distinct metacognitive skill, namely the ability to judge how common one's own diagnostic opinion is among peers. This form of social metacognition is a key prerequisite for the "surprisingly popular" methodology, which exploits systematic differences between what is correct and what is widely believed. By eliciting both a diagnostician's own diagnosis and their estimate about what diagnosis would be the most popular among others, it becomes possible to identify responses that are less common than expected yet more likely to be correct, thereby improving collective diagnostic accuracy.

---

[6] Asher Koriat ,When Are Two Heads Better than One and Why?.Science336,360-362(2012).
[7] Prelec, D., Seung, H. & McCoy, J. A solution to the single-question crowd wisdom problem. Nature 541, 532–535 (2017).

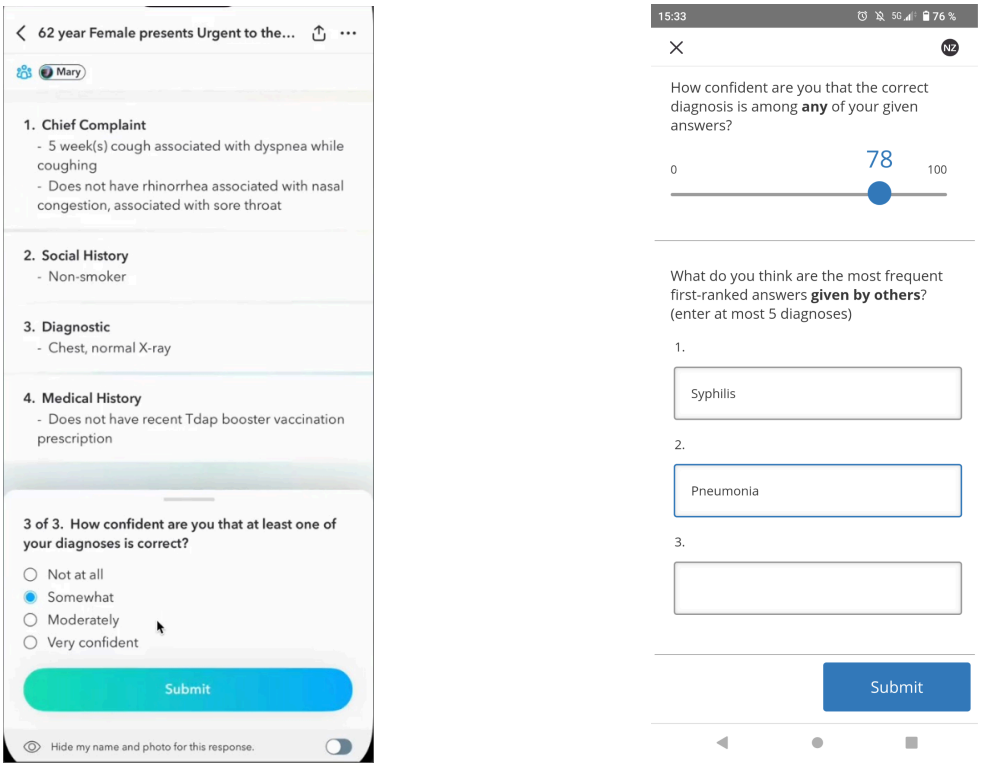

Figure 6. Left: Interface for the elicitation of subjective confidence estimation, with a 4-level Likert scale. Right: Interface for the elicitation of the popularity of diagnoses, where at most 5 different diagnoses are requested.

The surprisingly popular methodology has been applied mostly to closed-ended questions, but within HACID it must be adapted to open-ended ones, spanning all possible diagnoses for a given case. There are multiple ways to elicit popularity of diagnoses in an open-ended setting. For instance, one could propose the user to estimate the popularity of the diagnoses provided in their own differential, but that would be limiting. Alternatively, a set of diagnoses could be proposed, sampled from the ones that have been already suggested by others. Both these approaches limit the popularity estimation to a closed set. We decided to also keep the question about popularity open, asking users to indicate what diagnosis would be the most popular among the other respondents.

This feature has been implemented within the Human Dx platform by means of an in-app popup page that redirects the user to a survey where the popularity question has been provided together with the confidence estimation (see the right panel in Figure 6). This choice has been made to limit the number of steps the user had to undergo within the app to provide multiple pieces of information beyond the differential.

3.2. Exposure to diagnoses generated by diagnostic teams

Experimental studies within HACID have demonstrated that the collective diagnosis obtained by aggregating differentials from both humans and LLMs is the most accurate one, on average.[8] This finding has prompted other experiments to understand how diagnostic advice

[8] N. Zöller, J. Berger, I. Lin, N. Fu, J. Komarneni, G. Barabucci, K. Laskowski, V. Shia, B. Harack, E.A. Chu, V. Trianni, R.H.J.M. Kurvers, & S.M. Herzog, Human–AI collectives most accurately diagnose clinical vignettes, Proc. Natl. Acad. Sci. U.S.A. 122 (24) e2426153122, (2025).

received by homogeneous or hybrid teams gets integrated by physicians when it is received at different points during the diagnostic process (see Deliverable D4.2), for example before seeing the case detail (early-advice setting) or after having formed their initial hypothesis (late-advice setting).

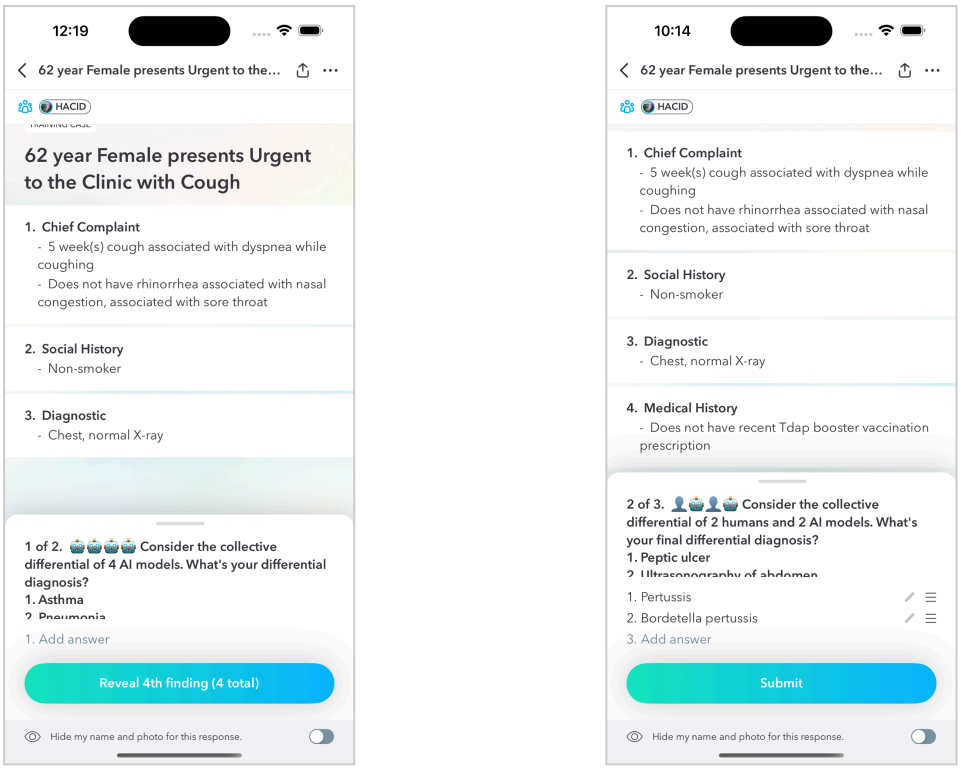

Figure 7. Left: Early-advice condition showing a collective formed from LLMs. Right: Late-advice condition showing a hybrid collective formed from 2 humans and 2 LLMs.

In the early-advice setting, the normal diagnostic UX has been adapted by inserting the advice directly into the diagnostic question, which is visible as soon as the case is accessed and only a short description of the case has been provided. In this way, the participant can see the advice while collecting information about the case and pondering about the possible diagnoses (see left panel in Figure 7). In the late-advice setting, the UX has been adapted by inserting a second question after the user has provided a first differential, which contains the advice in a similar format as in the early advice setting (see right panel in Figure 7). The late-advice setting has been implemented in this way to evaluate how the advice influences the diagnostic reasoning, hence forcing a first deliberation without advice and then requiring a second deliberation. In the future, late-advice could also be provided as a case-information element similar to the case findings, hence before the first deliberation is submitted.

3.3. Integrated chat with dedicated chatbots

Going beyond simple advice, feedback can be provided to users within a conversation with other agents, human or AI. In Deliverable D4.2, an experiment is described to evaluate the accuracy of medical deliberation teams (MDT), where expert physicians are paired either with another human physician or with AI chatbots. In order to implement deliberation rooms exploiting the UX on the Human Dx platform, a chatroom is generated and associated with a

case under investigation, enabling users to interact either in human-only dyads, or in hybrid human-AI dyads. The latter functionality is the one that best fits the Human Dx platform UX, because it allows interaction following user requests, and does not need arbitrary pairing with other users; the human-only dyads have been implemented mainly for experimental purposes (see Deliverable D4.2 for details). The deliberation rooms with AI systems exploit chatbots implemented with the OpenAI GPT-4o model, which is prompted to act as (i) a coach inviting the users to deliberate reflection (LLM-C), (ii) an expert evaluator, providing feedback about the pros and cons of the user differential diagnosis (LLM-E), and (iii) as a peer, discussing difference and similarity between the users diagnosis and the AI-generated one (LLM-X). The different deliberation rooms are displayed in Figure 8, and enable the user to have an open confrontation on the medical case under investigation, in order to improve the reasoning and arrive at a better diagnosis.

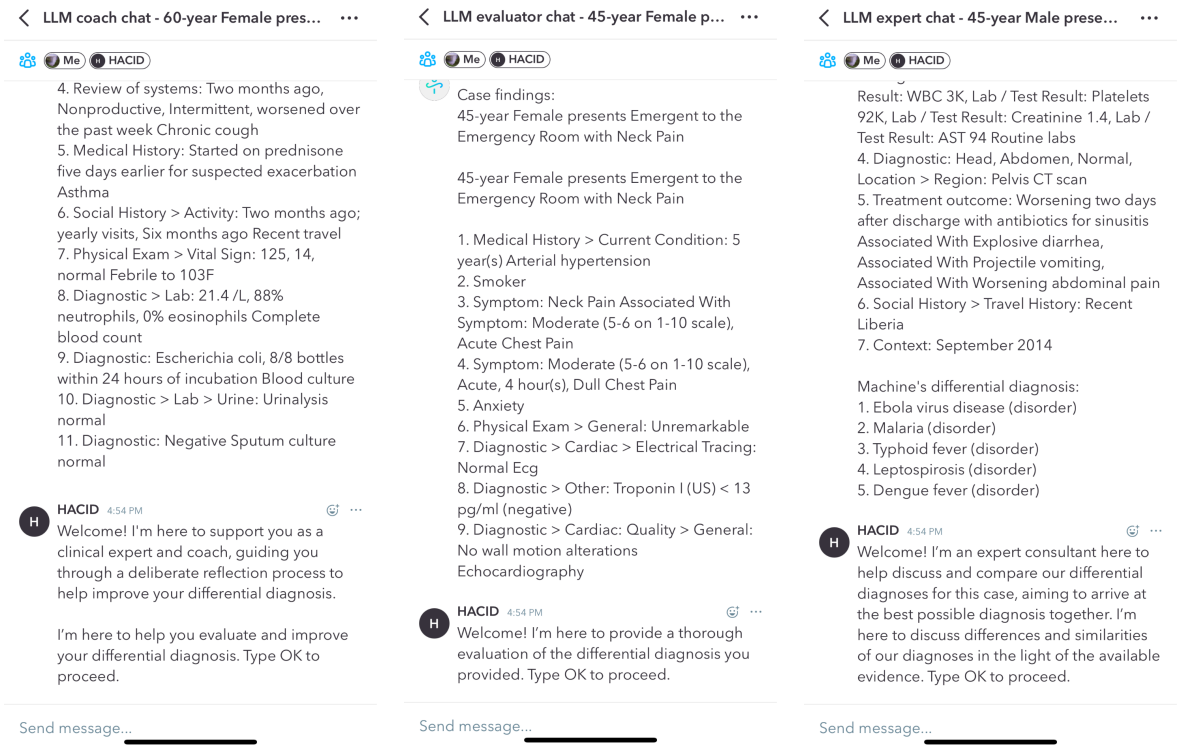

Figure 8: example of deliberation rooms with the LLM prompted as a coach (left), as an evaluator (centre) and as an expert (right).

4. Conclusions

In this deliverable, we have provided details about the different technologies that are integrated with the Human Dx platform in support of the demonstration of the studies performed within the HACID project. The different backend and frontend technologies have been presented in detail, providing a description on how they can support decision making by physicians engaged in the solution of diagnostic problems. The implemented technologies could be integrated in the Human Dx platform in production following a large-scale experimentation to provide evidence about the usefulness and usability by the Human Dx community. This step is necessary to support exploitation of the HACID technologies, and is described in detail in Deliverable D8.8.