HISaaS: A Cloud-Based Framework for Semantic Enrichment of Legacy Health Databases Using OpenAI Embeddings

Toufik Salmi, Abdeslem Dennai
Department of Mathematics and Computer Science,
Faculty of Exact Sciences, Smart Grids & Renewable Energies Laboratory,
Tahri Mohamed University, Bechar, Algeria

***Abstract*—Legacy Information Systems (LIS) confront significant interoperability challenges, particularly at the database level. If existing solutions focus on data exchange, they fail to address the fundamental structural incompatibilities of Legacy Databases (LDBs). In this article, we present Health Information System as a Service (HISaaS), an innovative dual solution based on AI and Cloud Computing, designed either to update a LIS with a new modern HIS (Health Information System); or to completely propose a new IS. This new HIS is the result of the interoperability of several modern HIS. HISaaS uses Cloud computing as an interoperability environment, Natural Language Processing (NLP), and semantic similarity to analyze and enrich the already existing structure of the LDBs with new tables and/or new attributes from modern and interoperable external databases. This solution uses a vector-based approach relying on contextual embeddings from OpenAI models and cosine similarity measurement to accurately match schema elements. In the end, an evaluation based on real healthcare scenarios demonstrates the effectiveness of this HISaaS solution.**

***Keywords*—Information systems; interoperability; database; cosine measurement; cloud computing; OpenAI; embeddings; NLP**

I. INTRODUCTION

The success of large companies such as Alibaba, Google, and Amazon has been achieved through the use of reliable and intelligent information systems. Information systems (IS) play an essential role in any business, enabling it to stay ahead of the competition. The main components of an IS are software, hardware, databases, communications, and human resources. A database is at the heart of an IS; it must be well designed, regularly updated, and secure. Its data is essential for making the right decisions and improving a company's performance.

Interoperability between ISs in the same field has become a crucial requirement for organizations and individuals in order to ensure optimal performance and high-quality service delivery. IS interoperability is the ability of two or more ISs to exchange and use data. [1] defines interoperability as the ability of a computer system to work with other existing computer products or systems, without restrictions on access or implementation. As illustrated in Fig. 1, interoperability operates at several levels:

At the technical level, basic connectivity is established using common protocols and transport mechanisms to ensure secure data flow between different platforms. At the syntactic level, the focus is on ensuring that the exchanged data between systems preserves a consistent structure to facilitate accurate data parsing and processing. Semantic interoperability ensures that the meaning of the exchanged data is preserved and unambiguous among diverse systems by employing shared vocabularies, ontologies, and coding standards. Finally, organizational interoperability enables seamless, cross-institutional collaboration and coordinated use of shared information by aligning governance, policies, and business processes [2].

Despite these structured levels, interoperability challenges persist due to the heterogeneity of information systems. The need to integrate data from multiple disparate systems and collaborate with them, systems that use different communication protocols, data formats, and query languages, leads to the emergence of conflicts at technical, syntactical, and semantic levels. While existing approaches often address these challenges with integration tools and standardized interfaces, a critical issue remains: the lack of focus on database interoperability. Most solutions prioritize data exchange over fundamentally modernizing the legacy database structure itself, leaving core data silos and structural incompatibilities unaddressed.

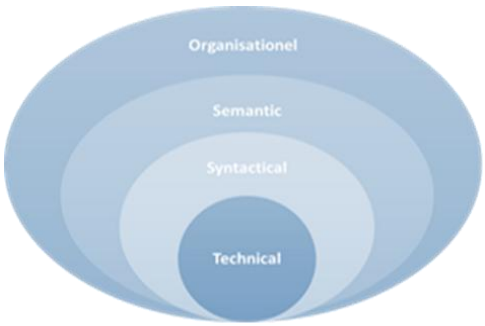

Fig. 1. Levels of interoperability.

Legacy information systems are known to be difficult to integrate with modern platforms due to significant interoperability issues, including the complexity of data migration and integration, dependence on obsolete data formats and structures, and the inability to communicate seamlessly with Cloud Computing services and APIs. To overcome these challenges, this article presents Health Information System as a Service (HISaaS), an innovative solution based on semantic AI

and Cloud Computing. With the scalability and flexibility of Cloud Computing, as well as advanced Natural Language Processing (NLP) techniques and semantic similarity measures, this HISaaS solution goes beyond simple data transfer by actively modernizing the very structure of the legacy database. By strategically adding new tables and/or attributes from external databases, HISaaS enriches and upgrades the existing system, making its data more useful and easily scalable without complete replacement, effectively bridging the gap between the existing infrastructure and the requirements of a modern, interoperable system.

The remainder of the study is organized as follows: Section II explores the context of this research, including Cloud Computing concepts and various techniques for measuring semantic similarity. Section III reviews related work in the field of IS interoperability. The architecture and detailed features of the proposed HISaaS solution are described in Section IV. Finally, Section V presents a comprehensive evaluation through a simulation based on the Cloud environment, detailing the performance of HISaaS in several real-world healthcare scenarios and demonstrating its effectiveness in structurally enriching a database, before concluding and proposing areas for future work in Section VI.

II. Background

This section covers two foundational pillars that our HISaaS solution builds upon. First, we look at Cloud Computing and explain why it serves as the ideal delivery platform for our service. Second, we explore semantic similarity techniques, particularly the vector-based approach using OpenAI embeddings that power the intelligent matching of database schemas across different healthcare systems. Understanding these concepts is essential before diving into how HISaaS actually performs the structural enrichment of legacy databases.

A. Cloud Computing

It is an environment for accessing shared computing resources servers, storage, databases, and software on demand via the Internet, rather than managing them locally. It transforms information systems by offering flexibility, scalability, and reduced infrastructure costs, with service models such as Infrastructure as a Service (IaaS), which offers fundamental compute, storage, and networking resources, Platform as a Service (PaaS), which provides complete development and deployment environment, and Software as a service (SaaS), which delivers fully managed, ready to use applications to end users [3][4] (see Fig. 2).

Fig. 2. Cloud computing basic delivery models.

Cloud Computing has become a pillar of modern information systems, enabling agility and digital innovation by offering agility and scalability through instant increase or reduction of resources according to real needs, mobility as the IS becomes accessible from anywhere and at any time, facilitating teleworking and real-time collaboration, cost optimization via the transition from heavy investment to subscription-based operating costs, and security and maintenance where updates and basic cybersecurity are ensured by the provider, freeing up IT teams for strategic projects [5].

In this work, Cloud Computing serves as the foundational delivery framework for the proposed HISaaS solution. It provides the scalable infrastructure and service-oriented model necessary to aggregate database schemas from multiple providers and offer new IS or database modernization as an on-demand service. Our HISaaS service uses the Cloud environment to provide seamless, secure connectivity to legacy systems and facilitates the structural enrichment of their databases without requiring significant on-premise resources, thereby directly handling interoperability issues.

B. Semantic Similarity

Semantic similarity refers to the degree of similarity between two texts (words, expressions, sentences, or documents) in terms of meaning. It is a key concept in computational linguistics and natural language processing (NLP), whose goal is to calculate the degree of similarity between two texts expressing identical or similar ideas, even if they use different words or structures [6]. Semantic similarity goes beyond simple text matching. It is based on the context or meaning of the text. “Medicine” and “Drug,” although not the same words, are semantically similar.

Semantic similarity methods can be systematically categorized based on the type of knowledge representation and computational strategy [7]. Multiple methods for semantic similarity exist, such as:

- Lexical methods: based on synonym matching (e.g., WordNet) [8].
- Vector-based methods: represent sentences or words as vectors in high-dimensional space (e.g., Word2Vec, Glove, FastText) [9] [10][11]. Here, the similarity is measured using cosine similarity or Euclidean distance.
- Contextual Embeddings: capture context-aware representations of text by using advanced models like BERT[12], GPT[13], or other transformer-based models.
- Knowledge-based Methods: based on hierarchical relationships, these methods measure similarity by using ontologies or semantic networks (e.g., WordNet).

Techniques for measuring similarity can be classified into multiple distinct approaches [14]; each one offers a unique perspective on assessing how closely related two concepts or objects are. The techniques can be categorized as follows:

- Arc-based: use the hierarchical structure of an ontology to assess semantic similarity by measuring the distances

between objects. The principle of this method is that the similarity between two concepts can be measured by the shortest path connecting them via a common ancestor. Wu and Palmer's metric [15] calculates similarity by assessing the distances from each concept to their least common ancestor, thereby providing a quantitative assessment of their semantic proximity.

- Nodes-based: analyses the information content of each concept to assess how similar they are. The main idea here is that two concepts are more similar when they share more information. Resnik's measure[16] is a well-known example. It uses the Information Content (IC) of the most specific concept that includes both concepts to compute their similarity. This approach focuses on the informational connections between concepts rather than just their structural connections within the ontology.
- Hybrid techniques: represent a synthesis of based-arcs and nodes approaches; these methods combine distance measures with information content, allowing for a nuanced assessment that captures both structural and semantic relationships. Jiang and Conrath's measure [17] is an illustrative example, which merges entropy (representing information content) with the distance between concepts. The combination of these approaches aims to enhance the accuracy of similarity evaluations, addressing the limitations found in using either technique in isolation.
- Vector space technique: these techniques are widely adopted in the field of information retrieval. These approaches represent objects as vectors within a multi-dimensional space, using various mathematical methods to calculate similarity. Cosine similarity[18], [19], for example, calculates the cosine of the angle between two vectors, indicating their similarity based on internal content. Additionally, Euclidean distance [20] calculates the straight-line distance between two points in the vector space; often adjusted to improve similarity assessments. The Jaccard index [21] further contributes to this domain by measuring the similarity between finite sample sets, defined as the size of their intersection divided by the size of their union. Each of these vector space techniques has unique strengths, making them applicable in various contexts depending on the nature of the data being analyzed.

Table I illustrates semantic similarity methods, categorizing different approaches for measuring how similar two pieces of text are in meaning. It organizes these methods into families based on their underlying principles and techniques.

Semantic similarity serves as the intelligent core for our proposed HISaaS that enables database enrichment from disparate databases. It is used to power two critical functions: synonym creation and embedding creation to match database schema elements (table names, attributes). Our method employs a vector-based approach to semantic similarity, using contextual embeddings generated by OpenAI's models to represent database schema elements as high-dimensional vectors. The similarity between these elements is then computed using *Cosine Similarity* on their respective vector representations. This approach allows our system to perform sophisticated semantic matching of database elements, identifying that "patient_tbl" and "client_records" are semantically related despite naming conventions, thereby enabling accurate structural enrichment of legacy databases.

TABLE I. SEMANTIC SIMILARITY METHODS

Method Family	Underlying Knowledge Representation	Example Techniques	Representative Models/Measures
Lexical Methods	Lexical databases (WordNet)	Synonym matching, string similarity	Levenshtein, Jaccard, WordNet Synsets
Knowledge-Based Methods	Ontologies / Semantic networks	Path-based, IC-based, Hybrid	Wu–Palmer, Resnik, Jiang–Conrath
Vector-Based Methods	Distributional semantics	Cosine, Euclidean, Jaccard	Word2Vec, GloVe, FastText
Contextual Embeddings	Transformer models	Contextualized embeddings	BERT, GPT, Sentence-BERT
Hybrid Methods	Combined symbolic + statistical	Ontology + embeddings fusion	BERT + WordNet, ConceptNet + Word2Vec

III. RELATED WORKS

This section reviews existing solutions related to information system interoperability and database schema integration. The review is organized around three themes directly relevant to our proposed HISaaS framework: health information system interoperability, cloud-based integration approaches, and AI-driven schema matching and structural enrichment. For each group, we discuss what the prior work contributes and where it falls short specifically regarding the structural modernization of legacy databases, which is the core gap our work addresses.

A. Health Information System Interoperability

A substantial body of work targets semantic interoperability in the healthcare domain through standard terminologies, ontologies, and data exchange protocols. Pellison et al. [22] proposed a semantic Web layer for tuberculosis data exchange in Brazil, using ontologies built on the Basic Formal Ontology (BFO) and SPARQL queries to map entities across heterogeneous Health Information Systems. The approach achieved structured, semantically consistent data exchange but remained tightly coupled to the existing system infrastructure and hard to extend to other diseases. Roehrs et al. [23] presented the OmniPHR model for personal health record interoperability, combining HL7 FHIR, OpenEHR, and NLP-driven automation to construct a unified view of patient data across disparate systems. While the model demonstrated integration potential, its NLP component needed extensive retraining for diverse medical terminologies and lacked real-world validation at scale.

More recent systematic reviews have taken stock of the state of semantic interoperability in EHRs (Electronic Health Records). Palojoki et al. [24] conducted a systematic review of approaches for enhancing semantic interoperability in EHR systems, identifying key elements such as shared terminologies (SNOMED CT, ICD, LOINC), standard models (FHIR,

OpenEHR), and cross-border data exchange mechanisms. Their review confirmed that achieving interoperability without loss of meaning remains unsolved and that most real-world deployments require considerable manual effort. Amar et al. [25] performed a systematic mapping review of 70 FHIR-based studies from 2012 to 2022, classifying semantic approaches into six categories including mapping, ML/NLP, ontology, and terminology services. They found that NLP and machine learning hold promise for unstructured EHR data but that automation of annotation and ontology comparison still needs significant human intervention. Critically, neither of these studies addresses structural enrichment of the legacy database itself; they focus on how data is exchanged, not on how the underlying schema is modernized.

B. Cloud-Based Interoperability Frameworks

Cloud computing has opened new paths for interoperability, particularly around scalability and service-oriented data management. Mohammed Sha and Rahamathulla [26] proposed a cloud-based healthcare data management framework that combines HL7, FHIR, and DICOM formats with a centralized data lake and API integration to streamline cross-system data sharing. The framework produced a scalable cloud system but faced persistent issues with legacy system compatibility, data ownership, and vendor lock-in.

Gohar et al. [27] introduced a five-layered patient-centric framework combining blockchain, cloud storage, and IoT for secure and semantically interoperable data exchange. Using SNOMED-CT, HL7 FHIR, and ICD standards, the framework outperformed existing blockchain-based approaches in security and exchange efficiency. Still, it remained at the prototype stage and lacked large-scale real-world testing.

Sony et al. [28] proposed an Edge Fog-based architecture integrating IoT, cloud, and the UMLS ontology to improve semantic interoperability in healthcare systems. The work showed reductions in computation overhead, latency, and network usage. However, the study relied entirely on simulations and raised concerns about inter-fog communication and data security. These cloud-based approaches share a common pattern: they create interoperability layers that facilitate data movement and exchange, but do not address the structural incompatibilities of the legacy database itself. They assume the source schema is adequate and focus on translation rather than modernization

C. AI-Driven Schema Matching and Structural Enrichment

The most directly related strand of recent work involves the use of language models for schema matching, the task of identifying semantic correspondences between elements of different relational schemas. Parciak et al. [29] ran an experimental study on using off-the-shelf LLMs for schema matching in the health domain, varying the amount of context information passed to the model through different "task scopes". Their results showed that LLMs can meaningfully bootstrap the matching process using only element names and descriptions, but that quality suffers when too much or too little context is provided. The study did not address multi-source schema enrichment or attribute-level gap analysis.

Sheetrit et al. [30] proposed ReMatch, a retrieval-augmented schema matching system that uses embedding-based candidate retrieval to narrow the search space before passing candidates to GPT-4 for ranking. The approach achieved strong results but showed significant performance degradation with smaller models and was not tested in healthcare scenarios involving legacy database upgrades. Liu et al. [31] developed Magneto, a hybrid pipeline that combines small and large language models for schema matching. A two-stage process, first narrowing candidates via contrastive column embeddings, then scoring with an LLM, outperformed single-model approaches and reduced the cost of labeled training data. That said, Magneto operates at the column level and does not attempt to identify missing attributes for structural enrichment. Ma et al. [32] introduced a knowledge graph-based retrieval-augmented generation approach for schema matching, using entity-type hierarchies to improve disambiguation for unseen schema elements. The method showed robustness under zero-shot conditions, but requires a pre-built knowledge graph and has not been evaluated in multi-provider legacy database settings.

Across all three research threads, a critical gap is consistent: existing solutions create interoperability layers to facilitate data exchange, but neglect the structural modernization of legacy databases. Schema matching work focuses on finding correspondences, not on using those correspondences to enrich a schema with missing tables and attributes. Health interoperability work relies on exchange standards and ontologies, not on semantic enrichment of the source database structure. The proposed HISaaS framework is designed specifically to fill this gap. It moves beyond data translation to actively identify missing schema elements (tables and attributes) from multiple cloud-hosted providers, using contextual embeddings and cosine similarity, and offers them to the user for structural integration.

IV. PROPOSED SYSTEM

HISaaS Health Information System as a Service is a cloud-delivered framework that gives organizations two distinct paths: they can build a completely new database from tables sourced across multiple cloud providers, or they can upgrade an existing legacy database by enriching it with new tables and attributes drawn from those same sources. The framework does not force a total system replacement. Instead, it works with what the client already has, adding structure where it is missing and leaving the rest intact.

The cloud platform is central to how HISaaS delivers both flexibility and scale. Because all schema catalogs are hosted and served from cloud infrastructure, clients can explore, compare, and compose database structures without requiring significant on-premise resources. The service adapts to organizations of different sizes, from a small clinic building its first schema to a hospital network looking to modernize a decades-old system.

A. General Model

The HISaaS workflow begins when a client selects a health domain from a catalog of available options. Once a domain is chosen, the system presents a list of information systems

offered by registered cloud providers that match that domain. For each selected system, a database schema is displayed, showing its tables and the attributes within them.

The client then chooses which tables to incorporate into their own database. This selection step is iterative: a client can work through multiple provider systems in sequence, adding tables from each one until the composite schema fully meets their requirements. The output of this process is either a new database ready to serve a new information system, or a set of enrichment operations to be applied to an existing one. Fig. 3 illustrates the full general workflow.

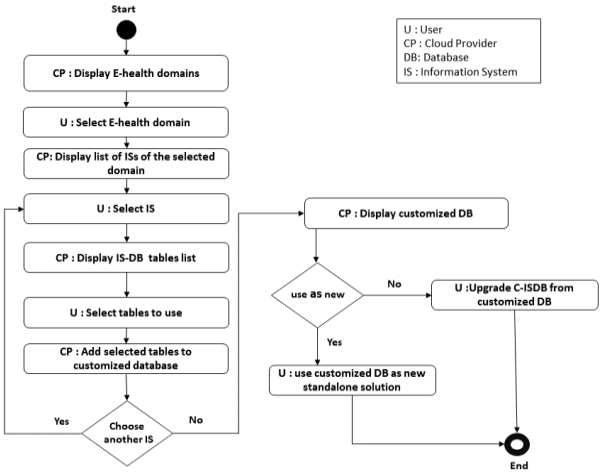

Fig. 3. HISaaS general model workflow.

B. 1st Scenario: Deployment of a New Custom Database

If the client needs to use a new customized database as a solution, he can either use it as it is or further customize it to better meet their needs. This customization allows for selecting the tables to include in the composite database and modifying them by choosing only the attributes necessary to meet the client's requirements (see Fig. 4).

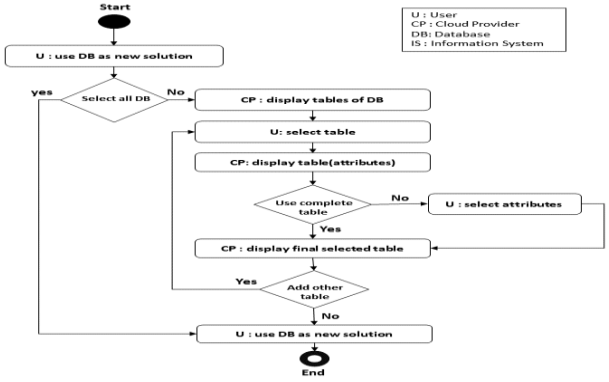

Fig. 4. Workflow for constructing a new custom database.

C. 2nd Scenario: Upgrading User Database

The scenario for updating the user database begins with uploading their database to the HISaaS platform. During the upload, HISaaS extracts and analyzes the metadata of the database, particularly the names of all the tables and their corresponding attributes, as illustrated in Fig. 5. The client has two options to improve their existing database:

1) Integrate a new table: The client can select a table from the custom database and integrate it into their existing IS database. He can use it as it is or customize it with selected attributes.

2) Update an existing table: The client can also select a table from the custom database and ensure that HISaaS finds a corresponding table in the client database. If a match is found, HISaaS identifies the attributes present in the selected table but absent in the client's table, and those can be integrated to enrich the existing data structure. This option enhances the data while preserving the core structure and familiarity of the existing dataset.

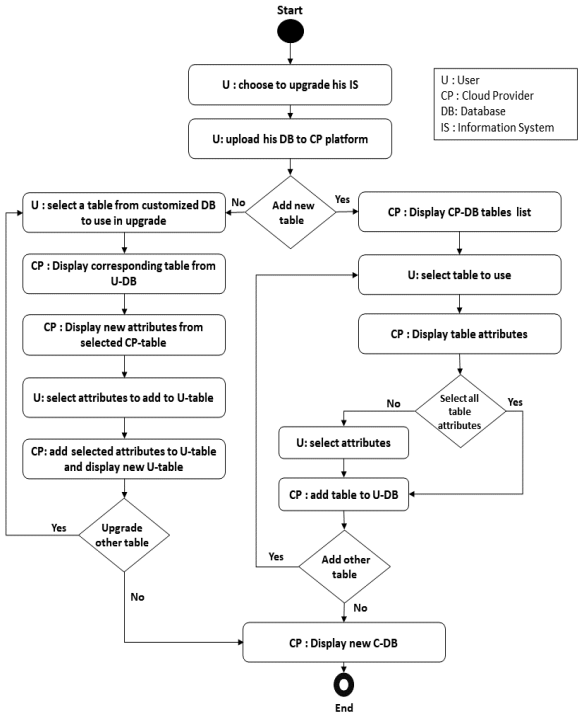

Fig. 5. Workflow for upgrading an existing legacy database.

D. HISaaS Database Enrichment Algorithms

Our proposed system is powered by two main algorithms that utilize OpenAI's advanced natural language processing (NLP) capabilities. First, identification of the most semantically relevant table for an enrichment request, and then the search for new attributes in the selected table from the customized database that are absent in the identified table in the client's database. These two algorithms facilitate the semantic matching of database schema elements (tables, attributes) by relying on the power of contextual embeddings and semantic similarity measures to overcome the heterogeneity and naming inconsistencies inherent in disparate information systems.

Semantic similarity between any two schema elements, whether table names or attribute names, is quantified by computing the cosine similarity between their respective embedding vectors. Given two dense vectors A and B produced by the embedding model, cosine similarity is defined as:

$$\text{cosine\_sim}(\boldsymbol{A}, \boldsymbol{B}) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\| \cdot \|\boldsymbol{B}\|} \quad (1)$$

where, A and B are the high-dimensional vector representations of two schema elements, A • B is their dot product, and ‖ A ‖ and ‖ B ‖ are their respective Euclidean norms. The resulting score lies in [0, 1] for text embeddings: a score of 1 indicates identical semantic direction, and 0 indicates complete orthogonality. Cosine similarity is particularly well-suited to this task because it is invariant to vector magnitude; two schema element names of very different character lengths are compared purely based on their semantic direction in the embedding space, not their raw size.

The embedding vectors are produced by OpenAI's *text-embedding-3-small* model, which maps each schema element name to a 1536-dimensional dense vector. These vectors encode contextual meaning rather than surface form, enabling the system to capture semantic relationships that character-level or keyword-based matching would miss entirely.

We select a similarity threshold of t = 0.80, validated both by established literature and our own experimental tuning. Externally, ontology matching adopts t = 0.80 for Language-agnostic BERT Sentence Embedding (LaBSE)[33]; BERT-based extraction achieves 88% accuracy at ⩾0.80 [34]. Our systematic sweeps independently confirmed 0.80 as the optimal precision–recall trade-off for HISaaS schema matching: lower values introduce spurious links (e.g., "patient_id" ↔ "physician_id"), while higher values miss genuine synonyms expressed through different naming conventions. Hence, t = 0.80 is applied throughout.

1) Procedure for finding the corresponding table: In this scenario, once the client selects a table in the custom database for the upgrade process, the system searches for a corresponding table in the client's database. The search process is carried out as follows:

a) Phase 1 Syntactic search: HISaaS starts by performing a syntactic search, seeking an exact match between the table name chosen by the user and table names present in the user's database. This is the fastest and most efficient method, but is often unsuccessful due to variations in naming conventions (e.g., "patient", "patient_tbl", "client_records").

b) Phase 2 Semantic search (contextual matching): If no syntactic match is found, HISaaS moves on to a more advanced search using OpenAI's advanced natural language processing capabilities to identify the semantics (meaning) of the table name.

This semantic search allows discovering tables that do not have an exact syntactic match with the name but are semantically related to the table the client wishes to update. This ensures that HISaaS can identify the most relevant table in the client's database, if it exists, even when the naming conventions or terminology used by the client differ from those in the database.

c) Algorithm 1: Finding the Corresponding Table

Algorithm 1 Explanation: The algorithm uses two components of the OpenAI Python library. Synonym generation is handled by ChatCompletion.create with GPT-4. Embedding generation is handled by embedding.create with the text-embedding-3-small model, chosen for its strong performance on semantic search benchmarks and its substantial improvement over the previous text-embedding-ada-002 generation at lower operational cost.

Algorithm 1: Find_Corresponding_Table

Input:

```
CP_DB     : list of table names in the customized database
Cu_DB     : list of table names in the user's database
Upg_table : table selected by client from CP_DB for upgrade
sim_thresh : minimum cosine similarity score (t = 0.80)
max_syn   : maximum number of synonyms to generate
```

Output:

```
CR_table  : matched table from Cu_DB, or None if not found
```

Phase 1 — Syntactic Search:

```
CR_table ← None
FOR each table IN Cu_DB DO
    IF table == Upg_table THEN
        CR_table ← table
        RETURN CR_table
    END IF
END FOR
```

Phase 2 — Semantic Search Preparation:

```
IF CR_table == None THEN
    // Generate synonyms for the upgrade table name via GPT-4
    synonyms ← Generate_SYNONYMS(Upg_table, max_syn)
    synonyms ← synonyms + [Upg_table]   // include original term

    syn_embs ← []
    FOR each syn IN synonyms DO
        syn_embs.append(GET_EMBEDDINGS(syn))
    END FOR

    cu_embs ← {}
    FOR each table IN Cu_DB DO
        cu_embs[table] ← GET_EMBEDDINGS(table)
    END FOR
```

Phase 3 — Semantic Matching:

```
best_match ← None
highest_sim ← 0
```

```
    FOR each (syn, syn_emb) IN zip(synonyms, syn_embs) DO
        FOR each (table, cu_emb) IN cu_embs DO
            // Apply Equation (1)
            sim ← COSINE_SIM(syn_emb, cu_emb)
            IF sim > highest_sim AND sim >= sim_thresh THEN
                highest_sim ← sim
                best_match  ← table
            END IF
        END FOR
    END FOR

    CR_table ← best_match IF highest_sim >= sim_thresh ELSE None
    END IF

RETURN CR_table
```

- Synonym Generation: The GENERATE_SYNONYMS function submits the upgrade table name to GPT-4 with a tailored prompt and receives a list of semantically related terms. These synonyms broaden the search space so that the system can recognise equivalent concepts even when they are expressed with different vocabulary across systems.
- Embedding Generation: GET_EMBEDDINGS transforms each synonym, the original table name, and all user table names into 1536-dimensional dense vectors encoding contextual meaning rather than surface form, enabling similarity computation to capture relationships that character-level matching would miss.
- Semantic Similarity: Cosine similarity [Eq. (1)] measures directional alignment between embedding vectors. Two conditions must hold simultaneously for a match to be accepted:
    - The similarity score must reach or exceed the threshold t = 0.80.
    - The score must be strictly higher than the best score recorded for that client table against any previously processed synonym.

Together, these conditions ensure that the returned match is the most semantically coherent candidate across the full synonym set, correctly linking, for example, “medications” to “drug_inventory” or “billing” to “invoice” despite the absence of any shared characters.

2) Procedure for identifying missing attributes: Once a corresponding client table is found, the system moves to the attribute level. The objective here differs from table matching: rather than finding a single best match, the system determines which attributes of the upgrade table are genuinely absent from the client table. Attributes that already exist by exact name or by semantic equivalence are filtered out. Only attributes that survive both filtering passes are surfaced as enrichment candidates.

a) Phase 1 Syntactic search: The algorithm checks each attribute in the upgrade table against the client table’s attribute list. Any attribute found by exact name is skipped; unmatched attributes are forwarded to the semantic phase.

b) Phase 2 Semantic search: For each unmatched attribute, GPT-4 generates synonyms. Both the synonyms and the client attributes are embedded. Cosine similarity [Eq. (1)] is computed across all synonym–client attribute pairs. If any pair exceeds t = 0.80, the attribute is considered present under a different name and excluded from the output. Only attributes for which no synonym achieves the threshold are added to the missing-attributes list.

c) Algorithm 2: Identifying Missing Attributes

Algorithm 2 Explanation:

- Syntactic Search: The algorithm iterates through CP_attrs. Attributes found verbatim in Cu_attrs are skipped; unmatched attributes are forwarded immediately to the semantic phase.
- Semantic Search: For each unmatched attribute, the system generates synonyms (e.g., for “user_id”: [“client_id”, “customer_id”, “cust_id”]), embeds them together with all Cu_attrs, and runs cosine similarity comparisons using Eq. (1). As soon as any synonym–attribute pair crosses t = 0.80, the attribute is flagged as present and the inner loop exits early. This early-exit design keeps computational cost proportional to the difficulty of the matching problem.
- Result: Attributes for which neither phase finds a match are appended to new_attrs. This list is presented to the client, who decides which attributes to add to their table. The human decision step is intentional: the system identifies enrichment candidates but does not automatically modify the schema, preserving full user control over the final database structure.

Algorithm 2: Upgrade_Table

Input:

```
CP_table   :  source table from the customized database
Cu_table   :  target table in user's database (from Alg. 1)
CP_attrs   :  list of attribute names in CP_table
Cu_attrs   :  list of attribute names in Cu_table
sim_thresh :  minimum cosine similarity score (t = 0.80)
max_syn    :  maximum synonyms to generate per attribute
```

Output:

```
new_attrs  :  attributes in CP_table
```

```
    absent in Cu_table
```

Phase 1 — Syntactic Search:

```
new_attrs ← []
FOR each attr IN CP_attrs DO
    IF attr IN Cu_attrs THEN
        CONTINUE   // attribute already exists — skip
    ELSE
        GOTO Phase 2 with attr
    END IF
END FOR
```

Phase 2 — Semantic Search:

```
FOR each unmatched attr IN CP_attrs DO
    synonyms ← Generate_SYNONYMS(attr, max_syn)
    synonyms.append(attr)
    syn_embs ← [GET_EMBEDDINGS(s) for s in synonyms]
    cu_embs  ← [GET_EMBEDDINGS(a) for a in Cu_attrs]

    is_found ← False
    FOR each syn_emb IN syn_embs DO
        FOR each cu_emb IN cu_embs DO
            // Apply Equation (1)
            sim ← COSINE_SIM(syn_emb, cu_emb)
            IF sim >= sim_thresh THEN
                is_found ← True
                BREAK
            END IF
        END FOR
        IF is_found THEN BREAK
    END FOR

    IF NOT is_found THEN
        new_attrs.append(attr)
    END IF
END FOR
```

RETURN new_attrs

V. Results and Discussion

To evaluate HISaaS proposed in a real-world setting, we created a Cloud environment using the OpenNebula Cloud platform. The infrastructure was deployed on a physical host machine with the following specifications: an Intel Core i7 10th Gen processor (6 Cores @ 2.59 GHz), 64 GB of RAM, and 1 TB of NVMe storage. The host machine ran Ubuntu Desktop 22.04 LTS. Within this environment, Oracle VirtualBox 7.0 was used to create a virtualized instance, which also ran Ubuntu Desktop 22.04 LTS. On this guest operating system, the OpenNebula 6.8 cloud management platform was deployed to orchestrate the creation and management of the virtual machines (VMs) required for the simulation.

The environment consisted of eight virtual machines (VMs) representing the cloud providers and user organizations. As shown in Fig. 6, three provider VMs (P1, P2, P3) with 2vCPUs, 12 GB RAM,100 GB storage, and running Ubuntu Server 22.04 LTS hosted real-world healthcare datasets: P1 hosting MIMIC-III (Medical Information Mart for Intensive Care) [35]Clinical Database V1.4, P2 hosting eICU (electronic intensive care unit) Collaborative Research Database [36], and P3 hosting a private hospital database. Four user VMs with 2vCPUs, 8 GB RAM,40 GB storage, and running Ubuntu Server 22.04 LTS represented clinics with legacy systems seeking to upgrade their schemas or build a new one. The HISaaS VM with 3vCPUs, 16 GB RAM,200 GB storage, and running Ubuntu Server 22.04 LTS hosted the core HISaaS service, responsible for coordinating semantic matching, schema analysis, and the enrichment processes across all provider and user nodes.

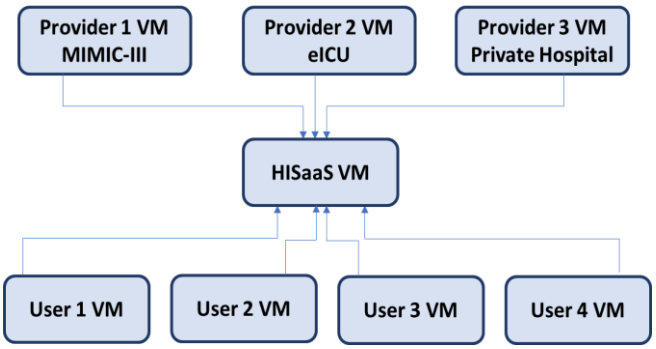

Fig. 6. Deployment architecture of the HISaaS cloud-based evaluation environment.

A. Use Case Scenario and Outcomes

Four users participated in the simulation, each performing schema enrichment operations. The outcomes are detailed below.

1) User 1 Targeted schema upgrade: The first user started with a database of 7 tables and 45 attributes: patients table with 8 attributes, doctors table with 7 attributes, appointments table with 6 attributes, medical records table with 7 attributes, billing table with 6 attributes, medication table with 5 attributes, and prescriptions table with 6 attributes.

The user created a customized database of 29 tables. Added four tables: doctors_details and Patient_medical_info from private hospital, icu_stays from private hospital, and prescriptions from MIMIC-III.

Upgraded four Existing tables:

- Patients (8 attributes) using patient_tbl (21 attributes) from private hospital; the matching table was found via semantic search with a cosine similarity of 0.83. Syntactic search identifies 21 new attributes, while

semantic search identifies 13 new attributes. The user chose 10 attributes to add to his Patients table

- billing (6 attributes) using invoice (11 attributes) from private hospital; the matching table was found via semantic search with cosine similarity of 0.88. Syntactic search identifies 10 new attributes, while semantic search identifies 7 new attributes. The user chose 7 attributes to add to his billing table.
- doctors (7 attributes) using doctors_tbl (17 attributes) from a private hospital; the matching table was found via semantic search with a cosine similarity of 0.82. Syntactic search shows 17 new attributes, while semantic search shows 15 new attributes. The user chose 12 attributes to add to his doctor's table
- medications (5 attributes) using medication (15 attributes) from eICU; the matching table was found via semantic search with a cosine similarity of 0.96. Syntactic search shows 13 new attributes, while semantic search also shows 13 new attributes. The user chose 4 attributes to add to his medications table.

The upgrade process was both precise and selective: the user did not adopt all new attributes suggested by the system (e.g., selecting only 10 of 13 semantically suggested for the "Patients" table). This reflects the effectiveness and the flexibility of our system. The final database schema grew to 11 tables (+57.1%) and 139 attributes (+208.9%), representing substantial structural enrichment. Fig. 7 illustrates this enrichment.

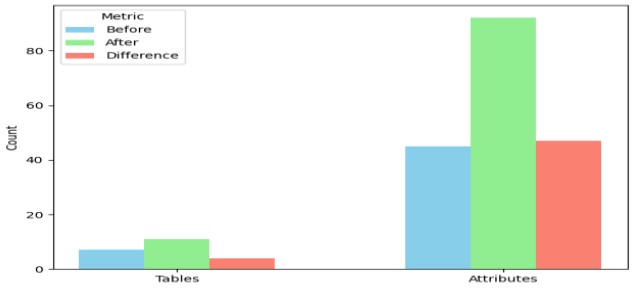

Fig. 7. User 1 database enrichment statistics.

2) User 2 Large-scale schema expansion: The second user began with a database of 5 tables and 31 attributes: patient_inf (7 attributes), visits (6 attributes), departments (3 attributes), health_records(6 attributes), and billing_records (6 attributes).

The user created a customized database of 26 tables. They performed a large-scale integration, added 14 tables: doctors_detail, doctor_tbl, email_config, email_delivery, email_inf, email_schedul, email_template, insurance, payment_account, payment_table, and user_tbl from private hospital. nurceAssissment, nurceCare, and nurceChart from eICU.

Upgraded three existing tables: "billing_records" using "invoice" from private hospital with 0.91 similarity, "health_records" using "patient_medical_info" from private hospital with 0.88 similarity, and "patient_inf" using "patient_tbl" from private hospital with 0.92 similarity.

All table matches were discovered using semantic search with similarities above 0.88, ensuring high confidence in the results. The outcome was a transformation to 19 tables (+280%) and 154 attributes (+396.8%), as illustrated in Fig. 8. This case highlights the capacity of HISaaS to support rapid schema expansion.

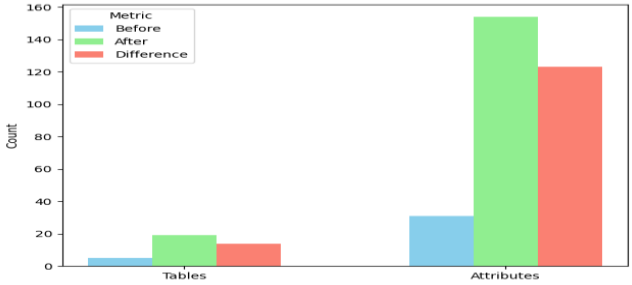

Fig. 8. User 2 database enrichment statistics.

3) User 3 Selective Enrichment: The third user starts with a database of 11 tables and 63 attributes: accounts (5 attributes), patients (7 attributes), medical_providers (6 attributes), patient_clinical_history (7 attributes), consultations (6 attributes), consultation_notes (4 attributes), prescriptions (6 attributes), transactions (6 attributes), support_tickets (6 attributes), chat_transcripts (5 attributes), system_logs (5 attributes). Created a customized_dB of 12 tables, chose to enrich their schema by adding 5 new tables from MIMIC-III and the private hospital database: payment_account_setup (4 attributes), payment_table (9 attributes), and schedul_table (8 attributes) from the private hospital; ICUSTAYS (12 attributes) and TRANSFERS(13 attributes) from MIMIC-III.

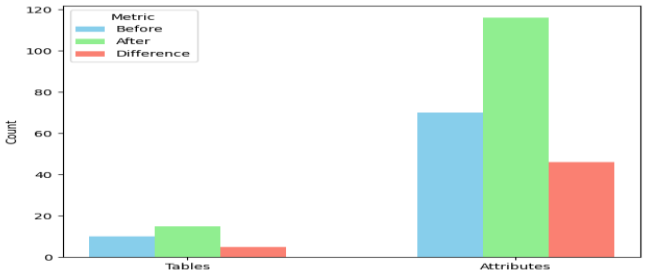

Fig. 9. User 3 database enrichment statistics.

User 3 kept the majority of their original database structure (11 initial tables), but enriched their schema by adding 5 new tables, leading to a final 16 tables. There were no table upgrades made. But the number of attributes increased by 53, from 63 to 116 (+84.13%) (See Fig. 9).

4) User 4 New schema construction: User 4 represented a new clinic without an existing schema. They used HISaaS to build a new customized database from scratch, composing it of 14 tables and 169 attributes with 14 tables and 169 attributes, representing a 100% new construction, sourced from all three providers: user_tbl (11 attributes), doctor_tbl (17 attributes), doctor_details (6 attributes), invoice (11 attributes), invoice_product (7 attributes), patient_tbl (21 attributes) and patient_medical_info (22 attributes) from

private hospital; nurseAssessment (8 attributes), nurseCare (8 attributes) and nurseCharting (8 attributes) from eICU; ICUSTAYS (12 attributes), PRESCRIPTIONS (19 attributes), SERVICES (6 attributes) and TRANSFERS (13 attributes) from MIMIC-III.

This use case demonstrates the system's utility for greenfield projects, enabling the rapid construction of a comprehensive, modern database.

B. System Performance Evaluation

The four user scenarios showcased distinct patterns of use, from targeted upgrades to full-scale schema construction, reflecting the HISaaS service flexibility and its adaptation to varying interoperability challenges and modernization needs across different organizational contexts. A summary of the enrichment across all users is provided in Table II.

TABLE II. ENRICHMENT STATISTICS ACROSS ALL USERS

User	Tables before	Tables After	Tables gain %	Attributes before	Attributes after	Attributes gain %	Tables added	Tables upgraded
User1	7	11	+57.1%	45	139	+208.9%	4 (from MIMIC-III and Private hospital)	4 (patients, billing, doctors, medication)
User2	5	19	+280%	31	154	+396.8%	14 (mixed sources)	3 (billing_record, health_records, patient_inf)
User3	11	16	+45.45%	63	116	+84.13%	5 (from MIMIC-III and private hospital)	0
User4	0	14	New (100%)	0	169	New (100%)	14 (mixed sources)	0

The results demonstrate substantial structural enrichment across all scenarios. User 2 achieved the highest relative growth in both tables (+280%) and attributes (+396.8%), reflecting a large-scale expansion of a previously minimal schema. User 1, while more selective, still realized significant attribute-level enrichment (+208.9%), underscoring the system's ability to perform deep, targeted upgrades. User 3 opted for a conservative approach, focusing on table additions rather than upgrades, resulting in a balanced increase in both tables (+45.45%) and attributes (+65.7%). User 4, building a schema from scratch, successfully constructed a fully modernized database with 14 tables and 169 attributes, demonstrating the system's utility for greenfield deployments.

To evaluate the system performance, we calculated the F1-score, precision, and recall for the applicable users (Users 1 and 2). The system achieved perfect performance in table matching for both users. The global precision, recall, and F1-score for table matching were all 1.00, indicating that the system successfully identified all relevant tables and avoided any false matches. The system also performed well in attribute-level enrichment. The global new attribute identification precision was 0.97, with a recall of 1.00, resulting in an F1-score of 0.98. The only recorded error occurred in User1's scenario, where the system recommended an attribute that already existed in the user's original table (doctors table upgrade), leading to a single false positive. This outcome highlights the overall accuracy and reliability of the HISaaS system in both structural and semantic enrichment tasks.

A key component of the system's efficacy is its use of semantic similarity, powered by OpenAI's advanced language models. This capability drives two critical functions within HISaaS: synonym generation and embedding creation, which are used for identifying corresponding tables and finding new attributes during the schema upgrade. By defining a cosine similarity threshold of 0.80, the system efficiently eliminated weak matches while maintaining recall, ensuring that most relevant structures were presented to the user. The HISaaS service proved particularly useful in real-world datasets where naming conventions and structural variations were significant across multiple providers. For example, HISaaS was able to match patients with patient_tbl, and billing with invoice, not based on name overlap, but on underlying semantic equivalence inferred from their embeddings.

The results show that HISaaS is well-suited for both schema upgrade and new schema construction tasks in heterogeneous healthcare environments. Its high precision reduces the need for manual correction, while its high recall ensures complete enrichment of the target schema. Using semantic similarity, the HISaaS effectively addresses the limitations caused by non-standardized naming conventions, an important challenge in healthcare data integration. These findings establish the HISaaS as a robust, scalable, and accurate approach for practical schema enrichment.

VI. CONCLUSION AND FUTURE WORK

This work presented a new approach to achieving interoperability in information systems based on a Cloud-VM environment by proposing a Health Information System as a Service (HISaaS) framework. The study conducted in this work highlighted the limitations of traditional interoperability solutions that primarily focus on data exchange and fail to address the fundamental structural issues of legacy databases. By using cloud computing and advanced AI-driven semantic similarity measurement techniques, the HISaaS solution offers a scalable and intelligent method to modernize legacy systems. HISaaS uses OpenAI's linguistic models to accurately identify semantically related tables and attributes to overcome naming conventions, terminologies, and schema design across different heterogeneous Cloud providers.

The evaluation was conducted in a simulated Cloud environment with real data from the healthcare sector. The results demonstrated that the proposed framework can significantly improve database structures, making them more complete and more interoperable without requiring a costly and large-scale replacement. HISaaS achieved perfect precision and recall in table matching with nearly perfect performance in attribute-level enrichment across various scenarios, including

targeted upgrades, large-scale expansions, selective enrichment, and complete schema construction.

In conclusion, the proposed solution addressed a critical issue in the creation and enrichment of IS by providing interoperability-based means in a Cloud-VM environment to improve the fundamental structure of clients' legacy databases.

Although the results presented in this work are promising, it is still necessary to make modifications to the HISaaS framework to improve its capabilities. These modifications will focus on further supporting the subdomains of healthcare, namely care, pharmacy, medical analyses, ... Moreover, it is important to extend the enrichment of the database schema beyond tables and attributes to include relationships, constraints, and metadata for a complete relational modernization. Moreover, automating the database enrichment process will minimize the need for manual interventions, thereby making the solution more accessible to a wider range of users and organizations.

Given that the HISaaS framework involves accessing and modifying sensitive data in the healthcare field (particularly patient data), future improvements should focus on investigating and strengthening security and privacy mechanisms in the Cloud environment.

REFERENCES

[1] G. da Silva Serapião Leal, W. Guédria, and H. Panetto, "Interoperability assessment: A systematic literature review," Comput. Ind., vol. 106, pp. 111–132, Apr. 2019, doi: 10.1016/j.compind.2019.01.002.

[2] K. S. S. Santos, L. B. L. Pinheiro, and R. S. P. Maciel, "Interoperability Types Classifications: A Tertiary Study," in XVII Brazilian Symposium on Information Systems, New York, NY, USA: ACM, Jun. 2021, pp. 1–8. doi: 10.1145/3466933.3466952.

[3] R. B. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "NIST Cloud Computing Reference Architecture," in 2011 IEEE World Congress on Services, IEEE, Jul. 2011, pp. 594–596. doi: 10.1109/SERVICES.2011.105.

[4] F. Wulf, T. Lindner, M. Westner, and S. Strahringer, "IaaS, PaaS, or SaaS? The Why of Cloud Computing Delivery Model Selection – Vignettes on the Post-Adoption of Cloud Computing," 2021. doi: 10.24251/HICSS.2021.758.

[5] I. Odun-Ayo, R. Goddy-Worlu, V. Samuel, and V. Geteloma, "Cloud designs and deployment models: A systematic mapping study," BMC Res. Notes, vol. 12, no. 1, Jul. 2019, doi: 10.1186/s13104-019-4474-y.

[6] W. H.Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," Int. J. Comput. Appl., vol. 68, no. 13, pp. 13–18, Apr. 2013, doi: 10.5120/11638-7118.

[7] D. Chandrasekaran and V. Mago, "Evolution of Semantic Similarity -- A Survey," Jan. 2021, doi: 10.1145/3440755.

[8] G. A. Miller, "WORDNET: A Lexical Database for English," in Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992, 1992, pp. 417–419.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in Advances in Neural Information Processing Systems (NIPS), 2013, pp. 3111–3119.

[10] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Trans. Assoc. Comput. Linguist., vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[13] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018.

[14] A. DENNAI and S. M. BENSLIMANE, "A New Measure of the Calculation of Semantic Distance between Ontology Concepts," International Journal of Information Technology and Computer Science, vol. 7, no. 7, pp. 48–56, Jun. 2015, doi: 10.5815/ijitcs.2015.07.06.

[15] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection," in 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, USA: Association for Computational Linguistics, Jun. 1994, pp. 133–138. doi: 10.3115/981732.981751.

[16] Y. Bin, L. Xiao-Ran, L. Ning, and Y. Yue-Song, "Using Information Content to Evaluate Semantic Similarity on HowNet," in 2012 Eighth International Conference on Computational Intelligence and Security, IEEE, Nov. 2012, pp. 142–145. doi: 10.1109/CIS.2012.39.

[17] L. Jiang, J. G. González, E. N. Navas-Conyedo, and Y. Chen, "Methods of Community Detection in Hybrid Network Applied to the Academic Network Analysis," Revista Cubana de Ciencias Informáticas, vol. 15, no. 5, pp. 130–161, 2021.

[18] M. Belabed and A. Dennai, "A Double Metan-Semantic Search Model Based on Ontology and Semantic Similarity: Asthma Disease," Journal of Information & Knowledge Management, vol. 22, no. 02, p. 2250082, 2023, doi: 10.1142/S0219649222500824.

[19] A. Dennai and S. M. Benslimane, "Toward an Update of a Similarity Measurement for a Better Calculation of the Semantic Distance between Ontology Concepts," 2013, pp. 197–207.

[20] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean Distance Matrices: Essential theory, algorithms, and applications," IEEE Signal Process. Mag., vol. 32, no. 6, pp. 12–30, Nov. 2015, doi: 10.1109/MSP.2015.2398954.

[21] V. A. Gonzalez-Huitron, A. E. Rodriguez-mata, L. E. Amabilis-Sosa, R. Baray-Arana, I. Robledo-Vega, and G. Valencia-Palomo, "Jaccard distance as similarity measure for disparity map estimation," IEEE Latin America Transactions, vol. 21, no. 5, pp. 690–698, May 2023, doi: 10.1109/TLA.2023.10130841.

[22] F. C. Pellison et al., "Data integration in the brazilian public health system for tuberculosis: Use of the semantic web to establish interoperability," JMIR Med. Inform., vol. 8, no. 7, 2020, doi: 10.2196/17176.

[23] A. Roehrs, C. A. Da Costa, R. D. R. Righi, S. J. Rigo, and M. H. Wichman, "Toward a Model for Personal Health Record Interoperability," IEEE J. Biomed. Health Inform., vol. 23, no. 2, pp. 867–873, Mar. 2019, doi: 10.1109/JBHI.2018.2836138.

[24] S. Palojoki, L. Lehtonen, and R. Vuokko, "Semantic Interoperability of Electronic Health Records: Systematic Review of Alternative Approaches for Enhancing Patient Information Availability," 2024, JMIR Publications Inc. doi: 10.2196/53535.

[25] F. Amar, A. April, and A. Abran, "Electronic Health Record and Semantic Issues Using Fast Healthcare Interoperability Resources: Systematic Mapping Review," Jan. 01, 2024, JMIR Publications Inc. doi: 10.2196/45209.

[26] M. Mohemmed Sha and M. P. Rahamathulla, "Cloud-based Healthcare data management Framework," KSII Transactions on Internet and Information Systems, vol. 14, no. 3, pp. 1014–1025, 2020, doi: 10.3837/tiis.2020.03.006.

[27] A. N. Gohar, S. A. Abdelmawgoud, and M. S. Farhan, "A Patient-Centric Healthcare Framework Reference Architecture for Better Semantic Interoperability Based on Blockchain, Cloud, and IoT," IEEE Access, vol. 10, pp. 92137–92157, 2022, doi: 10.1109/ACCESS.2022.3202902.

[28] P. Sony, G. Siva Shanmugam, and S. Nagarajan, "Enhancement of Semantic Interoperability in Healthcare Systems Using IFCIoT Architecture," KSII Transactions on Internet and Information Systems, vol. 18, no. 4, pp. 881–902, Apr. 2024, doi: 10.3837/tiis.2024.04.004.

[29] M. Parciak, B. Vandevoort, F. Neven, L. M. Peeters, and S. Vansummeren, "Schema Matching with Large Language Models: an Experimental Study," in Proceedings of the 2nd International Workshop on Tabular Data Analysis (TaDA24), Guangzhou, China, Aug. 2024. doi: 10.48550/arXiv.2407.11852.

[30] E. Sheetrit, M. Brief, M. Mishaeli, and O. Elisha, "ReMatch: Retrieval Enhanced Schema Matching with LLMs," arXiv preprint arXiv:2403.01567, 2024, doi: 10.48550/arXiv.2403.01567.

[31] Y. Liu, E. H. M. Pena, A. Santos, E. Wu, and J. Freire, "Magneto: Combining Small and Large Language Models for Schema Matching," in Proceedings of the VLDB Endowment, VLDB Endowment, 2025, pp. 2681–2694. doi: 10.14778/3742728.3742757.

[32] C. Ma, S. Chakrabarti, A. Khan, and B. Molnár, "Knowledge Graph-based Retrieval-Augmented Generation for Schema Matching," arXiv preprint arXiv:2501.08686, 2025, doi: 10.48550/arXiv.2501.08686.

[33] S. Banerjee, B. R. Chakravarthi, and J. P. McCrae, "Cross-Lingual Ontology Matching using Structural and Semantic Similarity," in Proceedings of the 9th Workshop on Linked Data in Linguistics @ LREC-COLING 2024, C. Chiarcos, K. Gkirtzou, M. Ionov, F. Khan, J. P. McCrae, E. M. Ponsoda, and P. M. Chozas, Eds., Torino, Italia: ELRA and ICCL, May 2024, pp. 11–21. [Online]. Available: https://aclanthology.org/2024.ldl-1.2/

[34] E. Herrmann, L. U. Von Rohr, F. Kotta Parambil, S. M Joy, and Y. Chittazhathu Kurian Kuruvilla, "Validating a semantic similarity approach for automated data extraction in phase II oncology trials.," Journal of Clinical Oncology, vol. 43, no. 16_suppl, Jun. 2025, doi: 10.1200/JCO.2025.43.16_suppl.e13666.

[35] A. Johnson, T. Pollard, and R. Mark, "MIMIC-III Clinical Database," PhysioNet, Sep. 2016, doi: 10.13026/C2XW26.

[36] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eICU collaborative research database, a freely available multi-center database for critical care research," Sci. Data, vol. 5, Sep. 2018, doi: 10.1038/SDATA.2018.178.