RESEARCH

A sample article title

Jane E. Doe[1*] and John R.S. Smith[1,2]

[*]Correspondence:
jane.e.doe@cambridge.co.uk
[1]Department of Science, University of Cambridge, London, UK
Full list of author information is available at the end of the article

Abstract

First part title: Text for this section.

Second part title: Text for this section.

Keywords: sample; article; author

Introduction

Data and Information modeling in the healthcare domain have witnessed significant improvements in the last decade owing to advances in the development of state-of-the-art information and communication technologies (ICT) and formalization of storage and messaging standards. Subsequently, the scope of Healthcare Management Information Systems (HMIS), medical ontologies, and Clinical Decision Support Systems (CDSS) has broadened, beyond the operational capabilities of traditional rule based systems. One of the major reasons behind this limitation is due to the numerous heterogeneities in healthcare at data, knowledge, and process level. Thus, healthcare interoperability which aims to provide a solution to this problem, can be compartmentalized into data interoperability, process interoperability, and knowledge interoperability. Data interoperability resolves the heterogeneity between data artificats, to enable, seamless and interpretable communcation among source and target organizations, while preserving the data's original intention during storage, communication, and usage (as defined by IEEE 610.12 [1], Health Level Seven International HL7, and Healthcare Information and Management Systems Society HIMSS [2]). On the other hand, process interoperability regulates the communication among organizational processes to provide compatability between process artifacts within and seamless transformations across different organizations[3]. Lastly, knowledge interoperability provides a sharing mechanism for reusing interpretable medical knowledge, acquired through expert intervention and other mechanisms, across decision support systems [4]. In more tangible terms, healthcare interoperability at data, process, and knowledge level can be exemplified within the healthcare constraints experienced due to the emergence of Covid 19. The operational capabilities of the current healthcare service delivery infrastructure has gone under tremendous stress due to Covid 19. World over, large primary healthcare units have managed to create separate units for managing patients, suffering from extreme cases of the novel coronavirus. For secondary and tertiary care units, government involvement has become necessary to filter coronavirus patients and adhering to a national pandemic response policy. These complex circumstances have enhanced the need for sharing patient data and state-of-the-art medical knowledge in real-time, to provide the medical experts with a tool to

make accurate and timely decisions. Data interoperability can enable the front line medical workers to fetch, understand, and use patient data, especially comorbidities, across organizational and physical boundaries, without suffering from societal taboos that may prevent the patient from sharing their complete and accurate medical histories. Knowledge interoperability can improve the knowledge acquisition and sharing protocols to provide the medical experts such as epidieomologists and vaccinologist, with latest information on affected population trends, disease diagnosis, treatment, and followup procedures, and interpretable decisions leading to positive or negative outcomes. Process interoperability can help reduce and in some cases remove the operational redundancies between health centers. In this way, successive healthcare treatments can take benefit from earlier diagnosis, treatment, and followup procedures, thereby reducing the stress on healthcare experts and systems. Standards such as Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR), and openEHR provide the foundations for storing and communicating medical data, through the use of well defined protocols. While systematized nomenclature of medicine—clinical terms (Snomed-CT) [5] and logical observation identifiers names and codes (LOINC) [6] provide a standard definition for clinical terminologies and laboratory tests, respectively. Similarly Medical Logic Module (MLM) provides a standardized way for expressing medical knowledge. However, the plethora of standards, necessitates the creation of bridging standards, that can resolve the heterogeniety between the medical standards. Substantial effort has gone into this endeavor with the Clinical Information Modeling Initiative (CIMI) [7] taking the lead in bridging the gap between HL7v3 and openEHR. Similarly, SNOMED CT and LOINC are working to resolve the redundancies between the two terminological standards since 2013. This healthcare interoperability solution follows a formal, albiet long process, which is greatly dependent on the human factor. However, the current healthcare scenario, requires a quick solution to create a scaffolding of an interoperable bridge between various healthcare providers. It is also important to ensure that this scaffolding should be able to support the formal standardization processes of the future. In [8]we have presented the Ubiquitous Health Platform (UHP), which provides semantic reconciliation-on-read based data curation for resolving data interoperability between various schema. This methodology is based on the creation and management of schema maps, that can provide the framework for transforming a source schema into a target schema. In the current manuscript, we will present our research work to build and manage the schema map knowledge base. Overall, our methodology has two major portions, firstly we apply a novel schema matching technique to create a transformation function σ for the participating legacy schema, and secondly, we have used the Ripple Down Rules (RDR) to manage our knwoledgebase, which will be presented in some detail, focusing primarily on the search and evolution services. In particular,

- Section 2 contains the details of our methodology, where we aim to present a reproducible theoretical framework.
- Section 3 provides the experimental setup
- Section 4 presents the results
- Section presents the related work
- Section 6 concludes the paper.

Related Work

Althubait et al. [9] proposed an ontology expansion methodology that identifies and extracts new class from text articles using word embedding and machine learning techniques. The authors identified the similarity of tokens and phrases of the text articles with the exiting classes of the ontology. The target ontology is expanded with classes from text articles having greater similarity with that of already added classes. A similar word embedding technique was also used by Nozaki et al. [10], where the authors used instance based schema matching technique to identify the semantic similarity between two instances. The results of the study showed the possibility of detecting similar string attributes of different schemas. Yousfi et al. [11] also utilized semantic base techniques and proposed xMatcher XML schemas matching approach. xMatcher transforms schemas into a set of words, followed by measuring words context, and relatedness score using WordNet. The terms from different schemas having similarities greater or equal to 0.8 are considered similar. Bylygin et al. [12] devised an ontology and schema matching approach by combining lexical and semantic similarity with machine learning approaches. The authors used lexical and semantic measures as features and trained various machine learning algorithms including Naive Bayes, logistic regression, and gradient boosted tree. The result achieved showed that the combination of algorithms outperformed the single modal.

Martono et al. [13] provided overview of previous studies related to linguistic approaches used for schema matching. Linguistic methods focused on finding strings and evaluate there similarity in different schemas. The string are normally normalized before to align both the strings before similarity comparison. The normalized strings are categories based on the information relatedness and element with similar category are compared using various similarity measure including Jaro-distance, Lavenstein (edit-distance), and many more. Alwan et al. [14] have summarized the techniques used in the literature for schemas and instances based schema matching. The information used for schema matching is categories into schema information, instance and auxiliary information. Most of the searchers have used syntactic techniques (including n-gram, and regular expression), semantic techniques (including Latent Semantic Analysis, WordNet/Thesaurus and Google Similarity) for schema level and instance level matching to achieve the final goal of data/information interoperability. Kersloot et al. [15] performed a comprehensive systematic review to evaluate natural language processing (NLP) algorithms used for clinical text mapping onto ontological concepts. The findings of the studies were evaluated with respect to five categories; use of NLP algorithms, data used, validation and evaluation performed, result presentation, and generalization of results. The authors revealed that over one-fourth of the NLP algorithms used were not evaluated and have no validation. The systems that claimed generalization, were self evaluated and having no external validation.

Methodology

Healthcare interoperability, with a focus on non-standard compliant medical schema, is dependent on the generation and validation of schema maps, as discussed above. To this end, the creation of a cohesive workflow is of utmost importance.

In our earlier work [8] we used maximum sequence identification and matching using suffix trees for syntactic matching of two distinctly sources data schemas. This was followed by semantic concept enrichment and subsequently concept matching, for creating rules in the form of schema maps. The simplified mapping functions, thereby created, provided a simple methodology for converting semi-structured medical data, into a non-persisted, interpretable, model form. In our current methodology we have utilized state-of-the-art machine learning techniques for converting medical schemas into semi-structured form, which is then used to create embedding vectors. These vectors are then used for modeling and creating interpretable rules for applying semantic reconciliation in the form of schema matching and/or transformation. These rules are managed by the RDR which allows fast inference using the relevant knowledge sub-tree, and can well manage knowledge evolution to scale out (by incorporating new schema) and up (by incorporating changes and additions to existing schema) In essence, the aim here is to define a uniqueness property ($\mathbb{P}$), as shown in Equation (1), whereby the semantic and syntactic uniqueness of each attribute as a disjoint union, is used.

$$range(\mathbb{P}) = \tau_{syntatic} \uplus \tau_{semantic} \tag{1}$$

The syntactic uniqueness as represented in Equation (2), is also a disjoint union of the measured attribute type, corresponding to one of integer, float, boolean, or string, and it valid values. The elements of the attribute type set have been determined based on the common primitive types available to the applications at an abstract level. While database schema used by EMR systems, correspond to the SQL datatypes[16], the application of semantic reconciliation-on-read strategy, instead necessitates the usage of common programming primitive types. On the other hand, "validValues" are determined based on what is contained in some sampled portion of the database. It is represented by the set of all possible categorical values (such as values of the attribute with type string which can be used to represent names, address, and others), the minimum and maximum values enclosed within open intervals ("( )") or closed intervals ("[ ]") to represent a range of numerical values (such as values of attributes with type float or integer), and boolean values (such as values represented by 1/0 or True/False).

$$\tau_{syntatic} = \{t \mid t\epsilon \{\{integer, float, boolean, string\} \uplus validValues\}\} \tag{2}$$

The semantic type is represented by the Equation (3), which is the set of all concepts corresponding to the leaf nodes of the suffix tree, generated from the name of the attribute. Suffix trees are used to divide a string into components, which is a very useful base for quickly identifying the longest common subsequence between a pair of strings and data compression. In our case, these trees provide a manageable list of words, contained with the name of attribute. For each substring, represented by a path of the tree, we then identify the corresponding concept, if it exists, thereby producing a set of concepts which may represent the attribute.

$$\tau_{semantic} = \{s \mid s\epsilon\{concept_{i,j} \mid i \leq width(suffixTree(s)) \wedge j\epsilon\{substring_i\}\}\} \quad (3)$$

Finally, the set AA of Amplified Attribute (AA), holds all unique attributes of participating schema, as shown in Equation (4). For each element of this set, the attribute is unique if its properties from Equation (1), are not similar to any other element of this set. The similarity function "$\sim$" is a threshold based loose bound, on the disjoint elements of "$\mathbb{P}$".

$$\exists a\epsilon AA \mid (\mathbb{P}(a) \wedge \forall b\epsilon AA \mid \mathbb{P}(b) \rightarrow a \sim b) \quad (4)$$

As a possible realization of this theoretical set notation form, and from a practical point of view, our methodology is presented in the following subsections.

Schema acquisition

In the first step of our semantic reconciliation methodology, we collect medical schema from five distinct EMR storage systems (S). These include patient reports from OpenEMR (s_1), 100,000 patient records from EMRBOTS (s_2) [17], custom database design by Pan et. al(s_3) for supporting regional clinics and health care centers in China [18], clinical knowledge discovery tool MedTAKMI-CDI (s_4) [19], and our custom implementation (s_5). These five schema follow relational database design, with logical entities, such as demographics, diagnosis, medicine or others, placed into tables which can be further linked to one or more tables. Each of these schema, fulfills the need of their respective information processing applications, however the lack of interoperability is very much visible here. Here, heterogeneity is caused by various factors, including the lack of standard terminologies and different normalization level. While SNOMED-CT provides a mechanism for identifying the standard codes for clinical terms and LOINC can be used for laboratory related terms, most attribute names are created based on the gut feeling of the database designer. While this behavior is not wrong, it does create syntactic differences in the participating schema. Consider the terms "name" and "patientName", which refer to the same attribute of the patient entity, however there is no standard way to represent it, with both being correct. As a result, s_1 and s_3 use the former representation, while s_2 and s_5 use the latter. Differences in normalization cause semantic differences, due to which some data could be available in one schema but absent in others, such as OpenEMR demographics identifying the patient's residential location using specific attributes like "Address", "City", "State", "Postal Code", "Country", and others. Similarly, "EncounterDate" from s_5 is semantically similar to "BeginDate" of "openemr_MedicalProblems" table in s_1, "AdmissionStartDate" of "LabsCorePopulatedTable" in s_2, "time" in "Diagnosis" table of s_3, and "dateOfAdmission" in "Diagnosis" and "CareHistory" tables of s_4. Finally, s_1 and s_3 have separate tables containing the medicinal prescription details, however the same details are unavailable in s_2, s_4, s_5. Once again, this is not an incorrect behavior since this information, might not be a part of the context or the requirements

for these EMR storage systems. In fact, the change in context of the EMR storage system from the initial time of development to a later stage of collaborative processing systems, is the main cause of heterogeneity. In order to provide an interoperable solution, it is therefore necessary to provide syntactic and semantic mappings, while taking the data instances (or tuples) into account. Data instances are used to enrich the semantic context of an attribute (a_i) by establishing its probable data type and possible value set. In our earlier work [8] s_1, s_2, and s_5 were used to generate over 115 million patient records, which are converted into a semi-structured form and stored in Hadoop Distributed File System (HDFS). We extended the same setup to **further include**

Schema processing

metadata In order to process the EMR schema set S, we use the methodology shown in **figAlgorithm**. For each data representation s_i, we first serialize its tables into Comma Separated Value (CSV) formatted flat files. Each table has a separate file, which contains the attribute names in the header and the data instances (following the same format and CSV struture), in subsequent lines. Each file is loaded in memory and converted into an algorithmic iterable structure. For each attribute, we then generate the suffix tree, which provides all possible substring representations contained within the attribute name. Then for each node of this suffix tree, if its length is greater than 2 (to avoid lookups for alphabets), we then query UMLS with approximate search strategy.Using Lexical Variant Generation (LVG) this strategy retrieves an expanded list of associated semantic concepts. If atleast one semantic concept is found for the substring, we replace the substring with the most relevant concept, based on the highest value of the confidence measure, creating the Suffix Concept Tree (SCT). The SCT is structurally smaller than or equal to the suffix tree, since all suffix which are smaller than 2 and those without a qualifying semantic concept, are discarded. Finally for each node of the SCT, the substring from the suffix tree and its corresponding concept is placed into a sentence, which is converted into an embedding vector using **BioBERT**. This embedding vector represents the domain qualified semantics of the source attribute.

We continue this process, until all substrings have been processed. This is followed by syntactic type check of the attribute, which is based on its data. Here we determine, if the attribute values correspond to one of Integer, Float, Boolean, or String types. Additionally, we collect the values of each corresponding data cell, and determine the set of possible values in each case. This semantic enrichment from UMLS concepts and syntactic enrichment from attribute types, is then saved into persistent storage for further processing.

Meanwhile the process continues for the next attribute, then the next table, and finally the next system, till no further processing is possible. The flat enriched schema, following the design shown in **figEnrichedSchema** provides the input to our knowledge base. The logically amplified structure of the attribute, AA, is represented in **figAttributeRepresentation**. Here, the elements of AA are categorized into three parts. “AttributeContext” contains the metadata conforming to an instance of the attribute’s existence, with its name, container table name and schema name, acting as a pointer, and schema version, source, and recorded date providing

the version control features. In this manner, the attribute's fully qualified reference, along with its existential context is identified. "AttributeType" then encapsulates the data type features of this attribute's instances, with its primitive data type, and values providing a representation of the attribute's instance. Finally, "Attribute Semantics" holds, the semantic information of this attribute, in the form of its suffix tree, SCT and Tree Embedding. A pair of AA's with distinct references, are then used as an input to the schema map generation process, which is explained in the next step.

Schema Map generation

Schema Maps, provide an interoperable bridge between two medical systems $(s_i \wedge s_j)$, by identifying the links between their participating attributes. This identification is based on the schema matching process, shown in Algorithm 1, which operates on a pair of amplified attributes, $(aa_i, aa_j) \mid aa_i \epsilon s_i \wedge aa_j \epsilon s_j$, and calculates the similarity score S.

Algorithm 1 Attributes similarity identifier

Input: AmplifiedAttributes aa_i, aa_j
Output: Similarity S

Similarity $S = 0$;
if $aa_i.AttributeContext == aa_j.AttributeContext$ **then**:
    $S = 1$;
else
    Syntactic Similarity $SynSim = 0$
    $at_i = aa_i.AttributeType$
    $at_j = aa_j.AttributeType$
    if $(at_i.DataType == at_j.DataType) || (at_i.DataType \Leftrightarrow at_j.DataType)$ **then**:
        $SynSim = 1$
    end if
    Semantic Similarity $SemSim = 0$
    $a\vec{e}_i = aa_i.AttributeSemantics.TreeEmbedding$
    $a\vec{e}_j = aa_j.AttributeSemantics.TreeEmbedding$
    $SemSim = \frac{a\vec{e}_i \cdot a\vec{e}_j}{\|a\vec{e}_i\| \|a\vec{e}_j\|}$
    $S = (0.5 * SynSim) + (0.5 * SemSim)$
end if
return S

This algorithmic process starts by comparing the metadata context of the two amplified attributes. The schema name, table name, attribute name, and version are used to establish the context of each attribute, which are then evaluated based on naive string matching of corresponding elements. If the pair refer to the same instance of the amplified attribute the process simply returns 1 as the similarity score. If however, the amplified attribute refer to separate instances, we then apply syntactic and semantic similarity on the pair. Firstly we compare the datatypes of the pair, to determine if they either have the same datatype or the datatypes are convertible. In our current approach *float* and *integer* datatypes are considered convertible, however this step is very much implementation specific and can be extended when the set of types is enhanced with newer data types, such as *short* is convertible to *integer* and vice versa, *bit* is convertible to *boolean* and vice versa, and so on. This test is used to set the "SynSim" score to "1", if the datatypes are equal or convertible, and "0" otherwise. Secondly, we compare the semantic concepts of the two amplified attributes, by applying **cosine** similarity between their embedding vectors. Since the embedding vector is based on the amalgamation

of the suffix strings and their corresponding suffix concepts, **cosine similarity can give a good measure of the direction of these vectors**. "SemSim", thereby obtained is then used in conjuction with previously obtained "SynSim" to calculate the similarity of the two amplified attributes. Using equal weights for syntactic and semantic similarity, we then re-scale their individual values to finally provide a similarity score between "0" and "1".

Schema Map evolution

Experimental Setup

Results

Results goes here ....

Acknowledgements
Text for this section...

Funding
Text for this section...

Abbreviations
Text for this section...

Availability of data and materials
Text for this section...

Ethics approval and consent to participate
Text for this section...

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Text for this section...

Authors' contributions
Text for this section ...

Authors' information
Text for this section...

Author details
[1]Department of Science, University of Cambridge, London, UK. [2]Institute of Biology, National University of Sciences, Kiel, Germany.

References

1. Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H., Springsteel, F.: IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries. IEEE Press, ??? (1991)
2. Healthcare Information and Management Systems Society: Definition of Interoperability (2013). http://www.himss.org/library/interoperability-standards/what-is
3. Khan, W.A., Hussain, M., Latif, K., Afzal, M., Ahmad, F., Lee, S.: Process interoperability in healthcare systems with dynamic semantic web services. Computing **95**(9), 837–862 (2013)
4. Ali, T., Hussain, M., Khan, W.A., Afzal, M., Hussain, J., Ali, R., Hassan, W., Jamshed, A., Kang, B.H., Lee, S.: Multi-model-based interactive authoring environment for creating shareable medical knowledge. Computer Methods and Programs in Biomedicine **150**, 41–72 (2017)
5. SNOMED Clinical Terminologies. http://www.snomed.org/snomed-ct/five-step-briefing Accessed 2020-03-19
6. LOINC. https://loinc.org/ Accessed 2019-03-19
7. CIMI: Clinical Information Modeling Initiative (CIMI) (2015). http://www.opencimi.org/ Accessed 2020-03-19
8. Satti, F.A., Ali, T., Hussain, J., Khan, W.A., Khattak, A.M., Lee, S.: Ubiquitous Health Profile (UHPr): a big data curation platform for supporting health data interoperability. Computing (2020). doi:10.1007/s00607-020-00837-2
9. Althubaiti, S., Kafkas, Ş., Abdelhakim, M., Hoehndorf, R.: Combining lexical and context features for automatic ontology extension. Journal of biomedical semantics **11**(1), 1–13 (2020)
10. Nozaki, K., Hochin, T., Nomiya, H.: Semantic schema matching for string attribute with word vectors. In: 2019 6th International Conference on Computational Science/Intelligence and Applied Informatics (CSII), pp. 25–30 (2019). IEEE
11. Yousfi, A., El Yazidi, M.H., Zellou, A.: xmatcher: Matching extensible markup language schemas using semantic-based techniques. International Journal of Advanced Computer Science and Applications **11**(8), 655–665 (2020)

12. Bulygin, L.: Combining lexical and semantic similarity measures with machine learning approach for ontology and schema matching problem. In: Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains"(DAMDID/RCDL'2018), pp. 245–249 (2018)
13. Martono, G.H., Azhari, S.: Review implementation of linguistic approach in schema matching. International Journal of Advances in Intelligent Informatics **3**(1), 1–9 (2017)
14. Alwan, A.A., Nordin, A., Alzeber, M., Abualkishik, A.Z.: A survey of schema matching research using database schemas and instances. International Journal of Advanced Computer Science and Applications **8**(10) (2017)
15. Kersloot, M.G., van Putten, F.J., Abu-Hanna, A., Cornet, R., Arts, D.L.: Natural language processing algorithms for mapping clinical text fragments onto ontology concepts: a systematic review and recommendations for future studies. Journal of biomedical semantics **11**(1), 1–21 (2020)
16. SQL Data Types. https://www.w3resource.com/sql/data-type.php. Accessed: 2021-01-14
17. Kartoun, U.: A methodology to generate virtual patient repositories. arXiv preprint arXiv:1608.00570 (2016)
18. Pan, L., Fu, X., Cai, F., Meng, Y., Zhang, C.: Design a novel electronic medical record system for regional clinics and health centers in china. In: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), pp. 38–41 (2016). IEEE
19. Inokuchi, A., Takeda, K., Inaoka, N., Wakao, F.: Medtakmi-cdi: interactive knowledge discovery for clinical decision intelligence. IBM Systems Journal **46**(1), 115–133 (2007)

Figures

Figure 1 Schemas used for knowledge interoperablity.

Figure 2 Sample figure title

Figure 3 Sample figure title

Tables

Table 1 Sample table title. This is where the description of the table should go

	B1	B2	B3
A1	0.1	0.2	0.3
A2	...	..	.
A3	..	.	.

Additional Files

Additional file 1 — Sample additional file title
Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title
Additional file descriptions text.