Lessons Learned from the Semantic Translation of Healthcare Data

Robert Techentin*, Jennifer St. Sauver†, Jeanne Huddleston‡, Barry Gilbert*, and David Holmes, III*
*Physiology and Biomedical Engineering, †Health Science Research, ‡Hospital Internal Medicine
Mayo Clinic College of Medicine, Rochester, MN

***Abstract*—Healthcare data provides a wealth of information that can be used to study and improve patient outcomes. Electronic Medical Records and other sources of healthcare data are often managed in relational database system and archived using modern data warehousing techniques. Contemporary semantic database technology has many advantages over traditional database systems; however, the utility of the semantic data can be limited if the data is not converted properly from a tabular representation. There are a variety of tools which will naively convert tabular data into a Resource Description Format semantic graph. Without proper guidance from the operator, the tools will generate a semantically weak database which doesn't have the necessary richness for semantic analysis. This paper describes the conversion process for two healthcare databases, with the goal of creating a robust dataset for semantic analysis. The "lessons learned" from this process are detailed in order to serve as a resource for other biomedical researchers and clinicians interested in generating a useful semantic dataset from their own relational databases.**

I. Introduction

As of January 2014, all United States healthcare providers are required to use electronic medical record (EMR) technology [1]. Similar initiatives are being adopted in Europe [2]. Going forward, these initiatives will generate unique data which can be used to both care for patients and improve the practice of medicine [3]. Moreover, researchers are aggregating other non-EMR datasets rich with biological and clinical data [4]. The combination of EMR and non-EMR data will yield massive amounts of data to be analyzed and interpreted. Towards this end, scientists are developing new methods to mine healthcare data. While there are many approaches to mining data, the use of semantic data mining [5] is of interest because of the power of the semantic representation and associated analytic techniques. In this paper, we report our experience with translating healthcare data into a semantic model addressing some of the challenges of data translation and potential benefits of this model.

Following the initial proposals of an electronic medical record in the 1960s and 1970s [6], [7], there has been steady progress in the development of patient centric electronic records. Much of the early work in electronic records was based on the MGH Utility Multi-Programming System (MUMPS) platform [8] which is still used in some capacity today. Nonetheless, as electronic database platforms developed, there was a general convergence on the relational database models [9]. Moreover, the diversity of data collected has grown dramatically. In addition to electronic medical records, billing data, scheduling data, laboratory reports, and surveillance data have been added to healthcare databases. These healthcare databases are now routinely mined for a variety of purposes ranging from quality assurance [10] to clinical decision support [11] to epidemiologic research [12].

As an alternative to the relational model of data, researchers began to propose a semantic model of the data in the mid-1970s [13]. In contrast to relational databases which focused on efficient data storage, semantic databases focused on the underlying knowledge within the data [14]. Early notions of a semantic data model were characterized by the concepts of generalization and aggregation, as presented in [15]. Subsequent study of the semantic data model have demonstrated additional benefits including flexibility of the database schema [16] and improved expressivity of the queries [17]. Modern implementations of semantic databases, such as YarcData's Urika appliance, offer very high performance analytics that can exploit the semantic data model. Due to these advantages, there has been a dramatic shift towards the semantic representation of medical data in the past decade [18].

While the semantic data model is promising, there are significant challenges to migrating to a semantic representation. Legacy data migration is of primary concern. As noted above, legacy data is largely represented in relational form. Moreover, semantic heterogeneity (i.e. the inconsistent representation of data) makes data consolidation a challenging task [19]. In this work, we describe the semantification of two real-world healthcare datasets along with the lessons learned in the semantification process. We will identify key factors in generating useful semantic data such that scientists and clinicians will be able to ensure proper translation of relational data into a semantic model.

II. Data Sources

As noted in [4], there are many different types of healthcare datasets in the Mayo Clinic enterprise. As representative examples, two were chosen to serve as canonical examples of semantication of legacy data. The first semantic database translation was the Rochester Epidemiology Project (REP) [20] database, which is a unique research infrastructure system that links together all of the medical records of the residents of Olmsted County, MN for approved medical research. This infrastructure makes it possible to conduct population-based descriptive, case-control, historical and prospective cohort, and cross-sectional research studies of most diseases and medical conditions. The REP is an amalgamation of several relational databases linked through a master database. The databases, carefully curated over many years, are a collection of medical records from clinics, hospitals, nursing homes, and other health care providers, containing patient demographics,

978-1-4799-6644-8/14/$31.00 ©2014 IEEE

diagnostic codes, procedure codes, and prescription codes. One key characteristic of the REP database is the linkage of multiple medical records for one person from different medical providers — linking 2.2 million patient records to 1.3 million individual persons.

The second semantic database was constructed from Mayo Clinic in-hospital data for 114,943 patient encounters (hospital stays) over a two year period. The Bedside Patient Rescue (BPR) project contains patient demographics, admissions, nursing evaluations, vital signs, and laboratory values with the goal of improving the accuracy of early warning systems [21]. As this is a less mature database, all data was manually reviewed prior to semantification to remove invalid data elements. In contrast to the multi-database design of the REP, the BPR data is aggregated into a single sparsely populated table of 149 columns and 38.2 million rows. The combined dataset was indexed by patient encounter and timestamp. These datasets represent the spectrum of data sources likely to be found in legacy database in medical centers.

III. Semantification

The process of semantification involves the translation of tabular data (generally constructed from one or more relational databases) into a semantic representation. The Resource Description Framework (RDF) is a data specification endorsed by the W3C [22]. The core of the RDF semantic representation is a triplet of values containing a Subject, Predicate, and Object. Entities are represented by unique international resource identifiers (IRIs), and the predicates express relationships among them. Because of the uniqueness constraint, RDF can link disparate datasets through common entity definitions. When RDF triples are combined, they can be interpreted as a semantic graph, where the Subjects and Objects are vertices, and the Predicates are edges. RDF triple-stores serve as graph databases for semantic data. Much like relational databases are queried using SQL, semantic databases are queried using SPARQL [23].

A. RDF Translators

RDF translators map data tables into RDF triples. While the data table has an implicit relationship between the elements of the same data record (i.e., in the same row), RDF triples explicitly specify every implied relationship between objects. The translator creates a unique identifier for each row of the table as the Subject, and using syntax consistent with the WC3 standard for IRIs. The Predicate, or relationship name, is derived from the column name, and is also an IRI. And finally, the object is the RDF representation of the data from the specific row and column.

There are many software packages which can be used for RDF translation [24]. Reviews of tools [25] and underlying technologies [26] can assist in selecting the best for a specific application. For the purpose of demonstration, different translators were used to convert the REP and BPR data into RDF. D2RQ [27] was used for the REP data because it translates directly from a relational database — interpreting data types and primary and foreign key relationships from the schema. Each database row identified by a primary key value is translated into an IRI, and data columns from that row are translated into triples as attributes of the IRI. Foreign key relationships are translated into triples linking to primary key IRIs from other tables. In contrast, CSV2RDF4LOD [28] was used for the BPR dataset, as it could translate character-separated-value (CSV) format data into RDF triples. CSV2RDF4LOD will, by default, generate a unique IRI for every row in the table, and then one attribute triple for each column. In both cases, translation is controlled by a configuration file which can be modified to perform some computations and even produce IRI objects instead of literals (a distinction discussed below).

B. Example Translation

Hospital in-patient data and clinical out-patient data is generally organized in tables in relational database systems, and generally indexed by patient identifier and timestamp. While there is great semantic richness in non-tabular data such as free-form text clinical notes or imaging data, this work focused on tables of scalar or categorical values that comprise much of these types of healthcare databases. There is ongoing research in the interpretation of free-form text into a semantic representation which is beyond the scope of this work.

As an example, consider the hypothetical admission / discharge records presented in Table 1. The column headings describe the contents and data type of each cell in the rows below and each line represents a unique record of data. As with many relational database tables, the first column is utilized as a primary key into the table, uniquely identifying each data record so that it can easily be referenced by other tables in the database. Note that this table does not have an explicit primary key that uniquely identifies each record. While relational database systems can provide a machine-generated primary key, individual data tables such as this one do not include that key.

TABLE I. Example of Admission / Discharge Data Table

Patient ID	Date	Admit_Discharge	Location	Room
1234	2010-05-11	A	Francis 3C	102
5678	2010-05-12	D	Eisenberg 21	102

The data in Table I can be used to generate the ten RDF triples shown in Table II and depicted as a graph in Figure 1. Each record in the table produces a "star" pattern relating the Subject to a number of Objects. In this case, the two patterns are connected because they both refer to "Room 102."

TABLE II. Default Translation of Admission / Discharge data to RDF Triples

Subject	Predicate	Object
<http://ADT_1>	<http://vocab/PatientID>	"1234"
<http://ADT_1>	<http://vocab/Date>	"2010-05-11"
<http://ADT_1>	<http://vocab/Admit_Discharge>	"A"
<http://ADT_1>	<http://vocab/Location>	"Francis 3C"
<http://ADT_1>	<http://vocab/Room>	"102"
<http://ADT_2>	<http://vocab/PatientID>	"5678"
<http://ADT_2>	<http://vocab/Date>	"2010-05-12"
<http://ADT_2>	<http://vocab/Admit_Discharge>	"D"
<http://ADT_2>	<http://vocab/Location>	"Eisenberg 21"
<http://ADT_2>	<http://vocab/Room>	"102"

However, there are several deficiencies in the default RDF translation. First, none of the literals have an associated data type, so it is impossible to distinguish the different semantics we might want to associate with "5678" (a Patient ID) and "102" (a room number). Even though both records in the

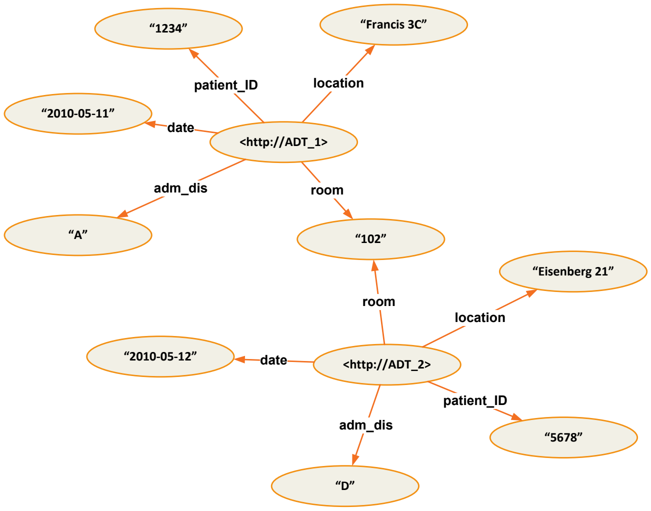

Fig. 1. Semantic Graph of Data Derived from Example Admit/Discharge Records (44487)

example have a room number of "102" they clearly refer to different rooms, as the locations are on different floors in different buildings (Francis 3C vs. Eisenberg 21). And further, the literal "102" could also refer to a blood pressure or temperature value in different contexts. This example demonstrates that simple RDF translation is insufficient to generate well-formed semantic data. To be successful, there are several practical issues which must be considered in the semantification process.

C. RDF Literals and Promotions

At the onset of semantification, the most important consideration in generating semantically meaningful RDF graphs is the quality and structure of the relational data. Returning to the example translation, if hospital floors and room numbers are normalized into relational database tables with appropriate constraints and foreign key linkages, translation tools can correctly synthesize unique IRIs for the room numbers and link them to patient data. If, on the other hand, room numbers are simply text values in tables, promoting those literals to identifiers will require extra effort. Unfortunately, even with adequate normalization and curation, simple translation of relational data into RDF may still yield inadequate results. Consideration of four broad categories of data representation will improve the semantic value of the data.

1) Literal Data Types: By default, much of the data generated during semantification is converted to literal values. As literals, these data points do not link to other data points because they do not represent unique objects. RDF supports typed literal values of different data types defined in the XML Schema Definition [29]. Built-in primitive types include string, numeric, logical and date/time types, with specializations and derivations supporting semantic concepts such as integer, float, double, byte, non-negative-integer, day, month, year, or duration. The standard representations for these types are based on the *xsd:* prefix, and the type names can be compactly written as *xsd:integer*, *xsd:date*, etc. It is presumed that proper type assignments will enable efficient queries.

Selection of specific literal data types is dependent upon the data and data source. Database-connected RDF translators, such as D2RQ, will choose a literal data type based on the database schema column type, but CSV2RDF4LOD has no schema information and defaults to strings. For the BPR dataset, we chose *xsd:integer* for values that are typically captured as integers such as systolic blood pressure and respiration rate, and *xsd:double* for floating point values such as drug dosages.

For both datasets, date and time values were typed as *xsd:date* in the case of patient birth dates and *xsd:dateTime* for all event timestamps. Computing durations between timestamped events is an important part of healthcare research, but is not directly supported by the SPARQL query language. To enable duration computations, we augmented the *xsd:dateTime* triples with integer timestamps computed from the date and time fields. We used the UNIX Epoch integer timestamp (number of seconds since January 1, 1970) and generated additional triples with *xsd:integer* type.

2) Nominal or Categorical Values: Many literal data values are essentially labels with semantic meaning and are candidates for promotion to IRIs (i.e. uniquely identified objects). Candidate literals may be string, numeric, or logical; usually have a limited range; and the values are semantically distinct. Examples in the REP and BPR databases include patient identifiers, diagnostic codes, and room numbers.

In some cases IRIs can be automatically generated by the translation tool. For nursing evaluations, for example, translation tools will generally create a unique IRI for each event (each row in the table), and generate triples linking the event to parameter values such as patient identifier or wound assessment. In other cases, IRIs must be manually composed from the tabular data values. Patient identifiers, for example, are stored as numbers which are encoded into the IRI. Care must be taken when linking disparate datasets to ensure that the table values are always composed into the correct IRIs.

It may be appropriate to synthesize IRIs to match established semantic models for common concepts such as diagnostic codes or drugs. Standardized vocabularies exist and have often been translated into RDF, proving a template for promoting literals in the dataset. The BioPortal (http://bioportal.bioontology.org/) provides definitions of several of these ontologies. Unfortunately, as the BioPortal demonstrates, the same term may be mapped in different ways to several different ontologies, including the Unified Medical Language System (UMLS), the Systematized Nomenclature of Medicine - Clinical Terms (SNOMED) and the National Drug File - Reference Terminology (NDFRT). Moreover, the RDF translators may not support the direct mapping of data elements to the various ontologies. For example, the BioPortal identifies 41 different class mappings for Diabetes Mellitus. One ontologic representation is the International Coding for Diseases Version 9 (ICD9) [30]. Diabetes Mellitus has the basecode identifer of "ICD9CM:250.00," however, the full identifier is 168 characters long with the key "MM_CLASS_6101." Since translators such as D2RQ have only string literals of "250" or "250.00," it is difficult to compose the IRI in a straight forward manner.

3) Ordinal Values: Some clinical observations are encoded into numeric values — where each value is distinct, and the numbers indicate relationships. For example, nursing evaluations include a parameter for Assisted Daily Living (ADL) Care, which can take on a value of "1," "2," or "3," corresponding to "Minimal Care," "Partial Care," and "Complete Care." The numeric form has little semantic meaning, and literal values are indistinguishable from other single digit integers. Promoting each of the three literal values to a IRI distinguishes them from other numeric values, allowing them to participate in graph analysis algorithms. One challenge, however, in the definition of ordinal IRIs is to convey the ordinal nature of the data. When a scale is represented as unique IRIs, generating queries such as *"Return all ADL evaluations stating greater than Minimal Care"* can be difficult to express.

4) Numeric Values: Many of the literal values in a hospital dataset are numeric with a continuous domain. Examples include systolic blood pressure, temperature, and age. It makes sense to maintain these as numeric values to support computations such as differences and averages. The individual numeric values, however, have limited semantic meaning. Unlike distinct labels, numeric values are related within their range. A systolic blood pressure of "101" is very similar to a value of "102." Some semantic analysis algorithms can take advantage of that similarity only if similar values are related by edges within the graph. Managing numeric data in semantic databases requires particular attention to the valid range of the data. While this may be assumed to be correct when coming from the source relational data, it is still necessary to place constraints on the data values during translation.

D. Literal Promotion Processing

There are several approaches to promoting literal values to IRIs, each of which has strengths and weaknesses. In some cases the choices are merely for convenience, while in other cases there are significant computational performance differences.

Translation tools such as CSV2RDF4LOD and D2RQ support mapping languages, allowing the user to customize the translation process by editing a configuration file. The tools generate a default configuration from an initial scan of the data source, which can then be edited to provide customization for future runs. This is particularly effective for assigning literal data types and promoting literals to IRIs containing the literal string. There are some limitations, however, in the computational capabilities of the mapping language, and different approaches may be required. These tools also run relatively slowly, and additional logic (e.g., SQL functions to trim white space) can negatively impact performance.

Alternatively, literal promotion can occur post-translation. Many semantic databases provide features for modifying data via an inference engine based on the Jena rule set. The inference engine evaluates the rules while building the database from RDF triples, and can create new triples based on complex logic and pattern evaluation. The inference engine can, for example, synthesize triples directly linking Persons to their respective Patient records in the REP data, bypassing the complexity of the linking table in the source relational databases.

```
# Inference Rule Linking Patient to Person, and vice versa
(?personLink vocab:person_id ?person)
(?personLink vocab:matched_pt_internal_id ?patient)
    -> (?person vocab:hasPatientRecord ?patient)
       (?patient vocab:isPerson ?person) .
```

Semantic databases using the SPARQL query language also support SPARQL UPDATE, allowing insertion and deletion of triples based on SPARQL evaluation semantics. The system can create a database checkpoint, saving the state of the modified database. In the case of the REP and BPR sematification process, a hybrid approach was used, leveraging CONSTRUCT queries and UPDATEs to promote labeled literals (e.g., "Demographics Ethnicity" and "Nursing ADL Assist" values) to IRIs.

```
# SPARQL Update Promote Nursing ADL Assist Literals to IRI
DELETE {?event NURSING:rn_adl_assist ?value .}
INSERT {?event NURSING:rn_adl_assist ?iri .}
WHERE {
  ?event NURSING:rn_adl_assist ?value .
  BIND
     (IRI(CONCAT(STR(NURSING:rn_adl_assist),"/",?value)
         AS ?iri)}
}
```

Post processing RDF triples can be effective when the translation tools or database system cannot perform the required tasks. The post processing program can be written in any language, as long as it can parse the format of the RDF triples. N-Triples format is quite simple, with exactly one triple on each line of the text file, enabling simple text pattern matching and transformations, and libraries exist for parsing more complex RDF representations. We employed the grep program with extended regular expressions to filter out provenance triples generated by CSV2RDF4LOD and a Tcl script to parse *xsd:dateTime* values and generate equivalent integer timestamp triples.

IV. Discussion

Naive translation of relational data to a semantic representation essentially offset the benefits of the semantic representation. The majority of the data is stored as literals and connectivity of the data is reduced. By doing so, the flexibility of the data model is compromised, and the expressivity of queries is limited.

First, consider the ability to create an expressive query. One of the primary goals of the REP program is identify complex clinical scenarios in the regional population. These types of queries may be used for hypothesis generation or possibly patient selection for clinical trials. One such complex query might resemble: *Find all procedures and diagnoses for patients who where diagnosed with Diabetes in 2009.* In a relational database, this requires first finding patients with Diabetes and then using those patient identifiers to find additional diagnoses and procedures. Alternatively, a single SQL query could be constructed but would require a self-join of the diagnosis table. In a properly constructed semantic database, the query can be accomplished in a single SPARQL query. The pattern shown in Figure 2 reflects the query applied to a naively translated database, where ICD-9 codes are represented as literals rather than IRIs. While the SPARQL query efficiently finds procedures and diagnoses without a self-join, it is limited to matching the single literal diagnostic code "249.10 ", which is only one of scores of codes related to diabetes. A more sophisticated semantic translation would link patient diagnostic events directly to IRIs of the ICD-9 vocabulary, eliminating the need for pattern UNIONs or filters.

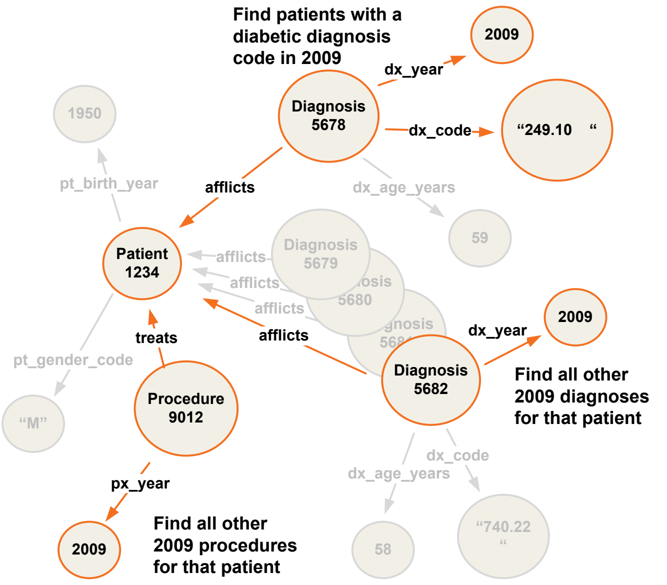

Fig. 2. Example Graph Query Pattern for Identifying Diabetic Patient Cohort in Rochester Epidemiology Project Semantic Data Model (43545)

The ability to create a flexible schema is also dependent on the quality of semantification. As noted above, one use case for collecting in-hospital data is to create an early warning system for patients who are deteriorating in the hospital. Several investigators have applied traditional statistical approaches to create various early warning systems [31]–[33]. These warning systems are, in essence, a composite score of several clinical variables. Depending on the model, the scoring system varies, however, the final result is to use the continuous variables to group them into "normal" and "abnormal" ranges. The scoring systems define ranges for parameters such as systolic blood pressure or temperature, and assign points for abnormal values. If a patient composite score exceeds a threshold, they will receive special attention. In a poorly constructed semantic model, the clinical variables are appropriately represented as numeric values, but the ordinal values (representative of scores for the clinical variables) would be represented as a literal as well. It would be difficult to tally the composite score for a given patient as their clinical variables are not linked to scores through IRIs. In contrast, a well-constructed semantic database allows for a flexible schema wherein individual scores could be linked to the data as IRIs. This is represented in the graph shown in Figure 3.

V. Conclusion

While semantic databases for healthcare have been discussed for over four decades, the field has not seen wide-spread adoption of these database models until recently. Several initiatives focus on the "ground up" development of semantic healthcare data which are used for specialized research applications. It is challenging to successfully migrate existing relational databases into a semantic representation without consideration for literal representation and promotion. In this text, we identify several key "lessons learned" from translating two real-world healthcare datasets. First, the quality of the source data is critical to the success of semantification process. Data should be properly normalized to minimize the need for manual promotion of literals. Second, when translating tabular data, care must be taken to correctly represent literal data. Of particular importance in biomedical research is the timestamp of events which can be converted into a UNIX timestamp to allow for time arithmetic. Third, when possible, data from multiple data sources should be properly promoted to ensure consistent IRI representation of entities. This can be facilitated through the use of an existing bio-ontology. Fourth, ordinal and nominal values should be promoted to IRIs to improve the expressibility of the semantic queries; however, care must be taken to manage the ordinality of such data. Lastly, when converting numeric data into a semantic representation, one should develop strategies to ensure the data is properly bounded during translation. Several approaches exist to facilitate the semantification and literal promotion process, however, users should be careful to ensure the quality of the semantic database.

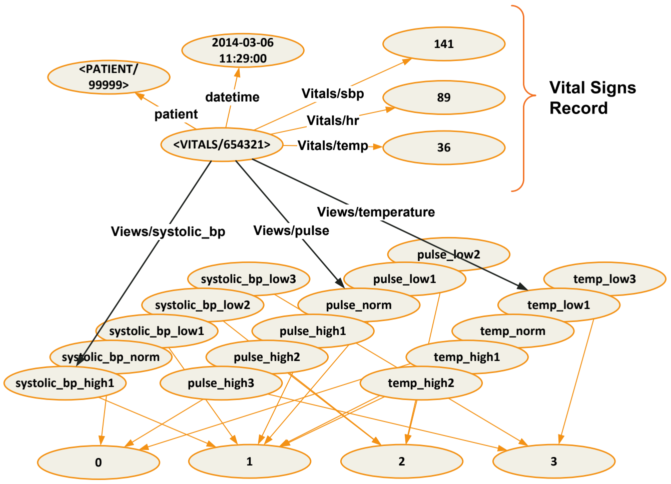

Fig. 3. Patient Vital Signs Record Mapped to ViEWS Scoring System (44488)

Acknowledgement

The authors would like to acknowledge support from the Rochester Epidemiology Project (NIA R01 AR030582) and Mayo Clinic Center for the Science of Healthcare Delivery. Moreover, the authors would like to thank the graph analytics team of the Special Purpose Processor Develop Group, Biomedical Imaging Resource, and Department of IT at the Mayo Clinic for developing the infrastructure to conduct novel semantic analysis on this data.

References

[1] "American Recovery and Reinvestment Act (ARRA)," February 2009.

[2] "EMR in Europe," Kalorama Information, Technical Report KLI4889963.

[3] A. K. Jha, C. M. DesRoches, E. G. Campbell, K. Donelan, S. R. Rao, T. G. Ferris, A. Shields, S. Rosenbaum, and D. Blumenthal, "Use of electronic health records in us hospitals," *New England Journal of Medicine*, vol. 360, no. 16, pp. 1628–1638, 2009.

[4] C. G. Chute, S. A. Beck, T. B. Fisk, and D. N. Mohr, "The enterprise data trust at mayo clinic: a semantically integrated warehouse of biomedical data," *Journal of the American Medical Informatics Association*, vol. 17, no. 2, pp. 131–135, 2010.

[5] K.-H. Cheung, K. Y. Yip, J. P. Townsend, and M. Scotch, "Hcls 2.0/3.0: Health care and life sciences data mashup using web 2.0/3.0," *Journal of biomedical informatics*, vol. 41, no. 5, pp. 694–705, 2008.

[6] L. L. Weed, "Medical records, patient care, and medical education," *Irish Journal of Medical Science (1926-1967)*, vol. 39, no. 6, pp. 271–282, 1964.

[7] ——, "Special article: Medical records that guide and teach," *New England Journal of Medicine*, vol. 278, no. 12, pp. 593–600, 1968.

[8] R. Greenes, A. N. Pappalardo, C. Marble, and G. Barnett, "Design and implementation of a clinical data management system," *Computers and Biomedical Research*, vol. 2, no. 5, pp. 469–485, 1969.

[9] C. Friedman, G. Hripcsak, S. B. Johnson, J. J. Cimino, and P. D. Clayton, "A generalized relational schema for an integrated clinical patient database," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, 1990, p. 335.

[10] Y. M. Chae, H. S. Kim, K. C. Tark, H. J. Park, and S. H. Ho, "Analysis of healthcare quality indicator using data mining and decision support system," *Expert Systems with Applications*, vol. 24, no. 2, pp. 167–172, 2003.

[11] R. Bose, "Knowledge management-enabled health care management systems: capabilities, infrastructure, and decision-support," *Expert Systems with Applications*, vol. 24, no. 1, pp. 59–71, 2003.

[12] M. K. Obenshain, "Application of data mining techniques to healthcare data," *Infection Control and Hospital Epidemiology*, vol. 25, no. 8, pp. 690–695, 2004.

[13] H. A. Schmid and J. R. Swenson, "On the semantics of the relational data model," in *Proceedings of the 1975 ACM SIGMOD international conference on Management of data*. ACM, 1975, pp. 211–223.

[14] J. Peckham and F. Maryanski, "Semantic data models," *ACM Computing Surveys (CSUR)*, vol. 20, no. 3, pp. 153–189, 1988.

[15] J. M. Smith and D. C. Smith, "Database abstractions: aggregation and generalization," *ACM Transactions on Database Systems (TODS)*, vol. 2, no. 2, pp. 105–133, 1977.

[16] R. Hull and R. King, "Semantic database modeling: Survey, applications, and research issues," *ACM Computing Surveys (CSUR)*, vol. 19, no. 3, pp. 201–260, 1987.

[17] S. Abiteboul, *Querying semi-structured data*. Springer, 1997.

[18] C. M. Machado, D. Rebholz-Schuhmann, A. T. Freitas, and F. M. Couto, "The semantic web in translational medicine: current applications and future directions," *Briefings in bioinformatics*, p. bbt079, 2013.

[19] A. Halevy, "Why your data won't mix," *Queue*, vol. 3, no. 8, pp. 50–58, 2005.

[20] W. A. Rocca, B. P. Yawn, J. L. St Sauver, B. R. Grossardt, and L. J. Melton, "History of the rochester epidemiology project: half a century of medical records linkage in a us population," in *Mayo Clinic Proceedings*, vol. 87, no. 12. Elsevier, 2012, pp. 1202–1213.

[21] J. M. Naessens, C. R. Campbell, J. M. Huddleston, B. P. Berg, J. J. Lefante, A. R. Williams, and R. A. Culbertson, "A comparison of hospital adverse events identified by three widely used detection methods," *International Journal for Quality in Health Care*, vol. 21, no. 4, pp. 301–307, 2009.

[22] R. C. W. Group. Resource description framework (rdf). [Online]. Available: http://www.w3.org/RDF/

[23] S. Harris and A. Seaborne. Sparql 1.1 query language. [Online]. Available: http://www.w3.org/TR/sparql11-query/

[24] C. Wiki. Semantic web converter tools. [Online]. Available: http://www.w3.org/2001/sw/wiki/Category:Converter

[25] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, and A. Ezzat, "A survey of current approaches for mapping of relational databases to rdf," *W3C RDB2RDF Incubator Group Report*, 2009.

[26] J. F. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker, "Survey of directly mapping sql databases to the semantic web," *The Knowledge Engineering Review*, vol. 26, no. 04, pp. 445–486, 2011.

[27] C. Bizer and A. Seaborne, "D2rq-treating non-rdf databases as virtual rdf graphs," in *Proceedings of the 3rd international semantic web conference (ISWC2004)*, vol. 2004, 2004.

[28] T. Lebo, J. S. Erickson, L. Ding, A. Graves, G. T. Williams, D. DiFranzo, X. Li, J. Michaelis, J. G. Zheng, J. Flores *et al.*, "Producing and using linked open government data in the twc logd portal," in *Linking Government Data*. Springer, 2011, pp. 51–72.

[29] P. Biron and A. Malhotra. Xml schema part 2: Datatypes. [Online]. Available: http://www.w3.org/TR/xmlschema-2/

[30] P. Brooks. International classification of diseases, version 9 (icd9). [Online]. Available: http://bioportal.bioontology.org/ontologies/ICD9CM

[31] R. Morgan, F. Williams, and M. Wright, "An early warning scoring system for detecting developing critical illness," *Clin Intensive Care*, vol. 8, no. 2, p. 100, 1997.

[32] C. Subbe, M. Kruger, P. Rutherford, and L. Gemmel, "Validation of a modified early warning score in medical admissions," *Qjm*, vol. 94, no. 10, pp. 521–526, 2001.

[33] D. R. Prytherch, G. B. Smith, P. E. Schmidt, and P. I. Featherstone, "Viewstowards a national early warning score for detecting adult inpatient deterioration," *Resuscitation*, vol. 81, no. 8, pp. 932–937, 2010.