Characterization of Semi-Synthetic Dataset for Big-Data Semantic Analysis

Robert Techentin[1], Daniel Foti[2], Sinan Al-Saffar[3], Peter Li[1], Erik Daniel[1], Barry Gilbert[1], David Holmes[1]
[1]Mayo Clinic College of Medicine, Rochester, MN, [2]University of Minnesota, Minneapolis, MN, [3]Semantic Scale, Tampa FL

***Abstract*— Over the past decade, the use of semantic databases has served as the basis for storing and analyzing complex, heterogeneous, and irregular data. While there are similarities with traditional relational database systems, semantic data stores provide a rich platform for conducting non-traditional analyses of data. In support of new graph analytic algorithms and specialized graph analytic hardware, we have developed a large semi-synthetic, semantically rich dataset. The construction of this dataset mimics the real-world scenario of using relational databases as the basis for semantic data construction. In order to achieve real-world variable distributions and variable dependencies, data.gov data was used as the basis for developing an approach to build arbitrarily large semi-synthetic datasets. The intent of the semi-synthetic dataset is to serve as a testbed for new semantic graph analyses and computational software/hardware platforms. The construction process and basic data characterization is described. All code related to the data collection, consolidation, and augmentation are available for distribution.**

Keywords—big data, semantic representation, data.gov, RDF, graph computing

I. INTRODUCTION

The emerging semantic web is delivering technologies that enable the creation and exploitation of large and complex, heterogeneous, and irregularly structured datasets. Until recently, the demonstration of semantic analysis of unstructured data has used small, "contrived" datasets [3]. With the development of distributed analyses [4] and large memory graph machines [5], it is important to develop semantically rich datasets for testing new semantic technologies. We extend our prior work [2] with an open, semi-synthetic, large, irregularly structured dataset for use in semantic analysis algorithm development and benchmarking of new triple-store technologies.

Unlike Relational Database Management Systems (RDBMS), which have evolved and improved over decades, semantic web data stores and query engines are recent developments and relatively immature [6]. Open source and commercial products are becoming available, but few have been demonstrated to yield good performance on terabyte-scale datasets. One of the factors limiting the demonstration of semantic database performance on large and complex datasets is the lack of practical use cases. There are several semantic benchmark suites that can be scaled to very large data sizes such as the Lehigh University Benchmark (LUBM) [7] and the Berlin SPARQL Benchmark (BSBM) [8]. While these benchmark datasets can be scaled to arbitrary sizes, data complexity is limited by the synthesizer program, and query complexity is limited to a set of notional queries developed for the benchmark's domain. There are also a number of semantic datasets created by "web crawl" or other automated collection mechanisms, but these have been shown to have less semantic richness than originally expected [9].

Considerable progress has been made in the construction of meaningful Life Science semantic datasets, integrating data from chemical, genomic, biological, clinical and other sources into linked data and ontologies that enable analysis based on the semantic relationships between datasets [10]. These datasets, unfortunately, are not semantically interesting due to the low diversity of the edge types and poor connectivity between nodes. For example, in [9], the authors examined UniProt along with two other real-world semantic databases. UniProt is a combined protein and genomic semantic database containing 2.04 billion triples [11]; however, Al-Saffar et al. found that there were only about 100 distinct nodes types and predicates.

There is significant value in semantic analysis of large medical record databases, which are already recognized to be semantically rich [12]. Unlike Life Sciences data, medical records are generally considered Protected Health Information, and stringently protected by HIPAA and other regulations. Even anonymized datasets, commonly used in research, are carefully controlled because reverse-engineering the identity of individuals in the dataset may be possible. This paper presents the design and construction of a large and complex dataset with characteristics similar to a large medical center data warehouse. The database was constructed using a relational DBMS and was translated into an equivalent Resource Description Format (RDF) semantic graph.

II. MAYO CLINIC ENTERPRISE DATA TRUST

The Mayo Clinic Enterprise Data Trust (EDT) is an example of a large and complex clinical data warehouse, integrating heterogeneous data from many patient care, education, research, and administrative database systems [1]. The EDT is constructed using industry standard methods of the data warehousing community, supporting information retrieval, business intelligence, and high-level decision making. Data is collected, curated, normalized, and stored in a non-volatile "living" data warehouse. The architecture of the EDT, illustrated in Fig. 1, supports capture of data from clinical information systems (on the left) into a staging area for normalization, and then storage in a non-volatile (i.e., new data is added, but never deleted) core. Several systems (on the right) support analysis and ad-hoc queries. Although data marts can be an effective tool for mining the data warehouse, creating a new data subset requires significant time and resources.

978-1-4799-6233-4/14/$31.00

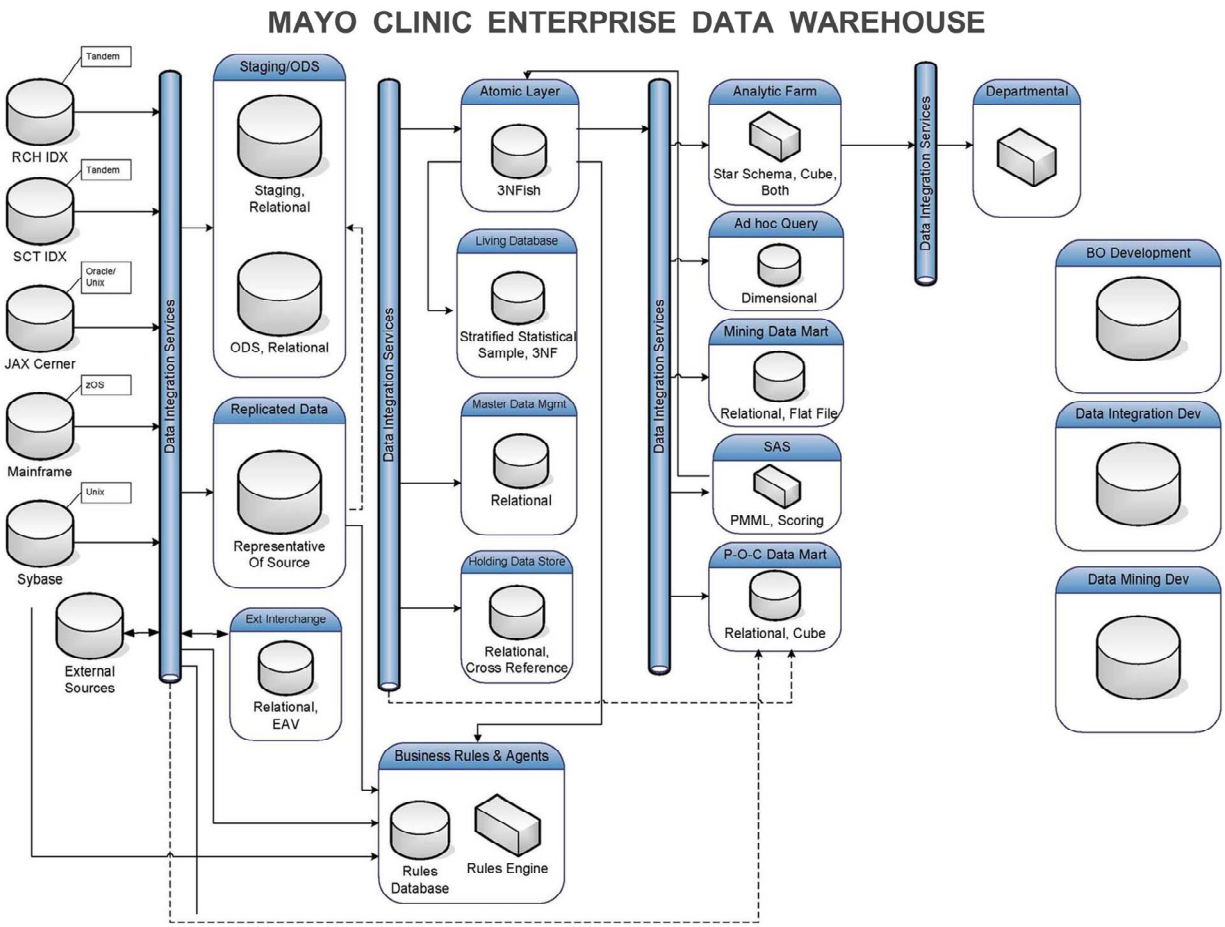

Figure 1. The Mayo Clinic Enterprise Data Warehouse is an amalgamation of many different departmental systems. Derived from [1] (42287)

The schema of the EDT is implemented in relational database systems with over 250 tables, with 2 to 58 columns, and an average of 15 columns per table. There are from zero to 15 foreign key constraints, with an average of 2 per table. To develop an understanding of the features of the data, the atomic layer of EDT was characterized using IBM Information Analyzer (v. 8.1.2). The subset, consisting of 61 tables and 1272 columns, was analyzed for data type, quantity and quality.

Approximately 25% of the data columns are numeric, 25% are dates and times; and the remaining 50% are represented by strings. The wide range of data quantity distributions shown in Fig. 2 are an indication that the system includes metadata (e.g., procedures, locations) in addition to data records. However, it should also be noted that several complex tables, with dozens of columns, hold many millions of records.

Adjusting for future growth in the field of healthcare information, we suggest that the target complexity for a synthetic healthcare database should exceed 500 tables (double the number in the existing EDT dataset) with an average of 15 (up to 60) columns per table, an average of 2 (up to 15) foreign keys per table, and record counts ranging from thousands to hundreds of millions. An equivalent semantic graph would have an equivalent classes (nominally, based on the number of tables) and approximately 500 distinct predicates – linking instances of those classes (based approximately on the assumption of 2 foreign keys per table). The total size of the semantic dataset could range into the hundreds of billions of triples.

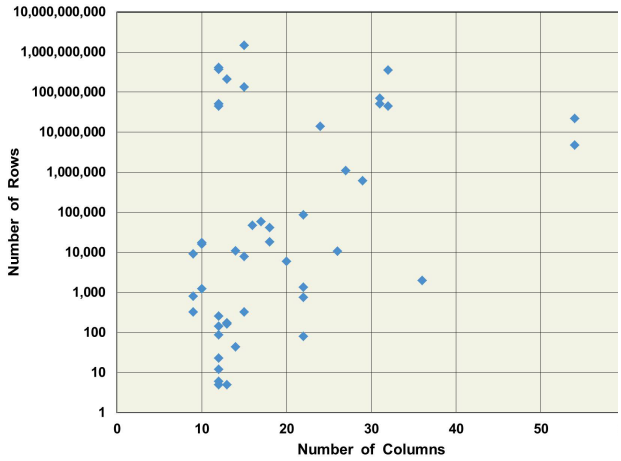

Figure 2. Data Quantity Distribution of the Mayo Clinic Enterprise Data Trust. The number of columns and rows are shown for a subset of 67 tables in order to characterize the data quantities likely to be found in real-world data. (44009)

III. Development of Synthetic Dataset

Although the purpose of this work was to develop a large semi-synthetic, semantic dataset, the real-world use case is the migration of existing data into a semantic model. Most medical records either use a traditional RDBMS [13] or a MUMPS database environment [14]; however, they all have a standard SQL interface for accessing the data. To mimic this scenario, we first created a semi-synthetic RDBMS dataset containing features similar to the EDT. After synthetically augmenting the data, the complete dataset was mapped into RDF.

A. Source Data

Data.gov (http://data.gov/) is a compilation of US government data collected and maintained by several different government organizations. Per the data.gov website, "a primary goal of Data.gov is to improve access to Federal data and expand creative use of those data beyond the walls of government by encouraging innovative ideas." A cursory review of the website showed that there were 6,230 data sources listed, representing data from 57 different agencies. The data were stored in a variety of different formats including comma separated values (CSV), Microsoft Excel (XLS), extensible markup language (XML), keyhole markup language (KML), and other spreadsheet or word processing document formats. Some data was stored and maintained by data.gov while other datasets are managed by their respective agencies.

Initial review of the data.gov repository found that the datasets were poorly linked between sources. Specifically, common variables found in several datasets (country, state, city, etc.) were not represented in a linkable form. However, the datasets did provide strong variability in the distribution of values. Some variables (i.e., columns of data) were uniformly random while others fit a normal distribution. The datatypes of each variable included categorical (e.g., states, zipcodes), continuous (floating point or integer), string, geocode, and date/time. Accordingly, the data.gov datasets provide a rich source of highly varied data containing real-world distributions of datatypes and data values; however, the connectivity of the data is limited and requires additional synthesizing.

Because the first-stage goal was to construct a relational database, only tabular datasets from data.gov were appropriate. Software was written to identify and download all text, CSV, and XLS sources from the data.gov repository. In all, 3,178 data sources were identified, downloaded, and reviewed. Some datasets were excluded to avoid the use of human or corporate identifiers or information that could be used to identify a specific individual or private entity. All datasets were converted into CSV format and then loaded into an SQL database.

B. Building a consolidated SQL Dataset

The source datasets were loaded into a MySQL (v. 5.2) database, which is an open source SQL compliant database system. Each data source was mapped into one or more independent tables. Closely related data sources (e.g., the same data, reported over multiple years) were loaded into the same tables.

Since the source data did not contain keys between different source tables, relationships among the database tables were added as arbitrarily-defined foreign key constraints, linking each table to at least one other table, forming a completely connected schema. Each table was augmented with a primary key column and a random number of foreign key columns. The number of foreign keys was based on a Gamma distribution with a range of 1 to 15, with an average of 3 keys per table. Foreign key reference tables were randomly chosen, including self-referencing tables. There was no semantic meaning to the links. Primary key columns were populated with unique integers. Foreign key columns were populated by references to the primary key values of the referenced table.

C. Synthesizing Additional Data Records

In order to create databases of sizes larger than the initial data, the number of records in the database had to be augmented. Unlike existing synthetic databases which make assumptions about data distributions, our approach made no assumptions; instead it sampled the real-world distributions of the source data. By artificially augmenting the number of rows in each table through statistical sampling, the database can be made arbitrarily large while maintaining the distributions of the initial data columns. The number of additional rows to be added to each table was selected at random using an exponential of the lognormal distribution. This approach ensured that most tables would have similar numbers of records while a few tables have one or two orders of magnitude more records. The data augmentation script had default values built in for row ranges (from 10,000 - 500,000), but script coefficients could be adjusted to give larger or smaller results. Multiple different augmentations were applied using different coefficients to create databases with differing sizes, ranging from 2 GB to 1 TB.

The inverse sampling method was used to generate synthetic values. Inverse sampling uses a uniform random sample (U) to generate new values (X) matching the variable distribution (D), according to $X = F^{-1}(U) \sim D$. $F^{-1}(\cdot)$ is the inverse cumulative distribution function (CDF). In the case of non-categorical (floating point or integer data type) variables, the CDF is both continuous and monotonically increasing. For categorical variables (determined by the cardinality of the data), a similar discrete approximation was used to create the inverse CDF. Most variables were sampled via the univariate inverse sampling technique; however, each table included one pair of correlated variables. To mimic jointly distributed variables, two columns in the table were assumed to have a

joint probability density function (PDF) with respect to each other. The inverse sampling method was applied by first sampling the marginal PDF of one variable and the conditional PDF of the second variable.

Once all the columns and tables had been augmented with new records, the foreign key constraint columns were updated to reference the augmented tables.

D.Mapping to RDF

The Resource Description Framework (RDF) is used to describe data and relationships for the Semantic Web [15]. The Data-to-RDF mapping application (D2R) is a tool for publishing the contents of a relational database in RDF syntax [16]. The default D2R mapping created an RDF schema class (rdfs:Class) for each database table, and a Uniform Resource Identifier (URI) for each row. Data columns were generally translated into literals with types based on the relational database schema. Foreign keys were translated into links to URIs from other tables. The resulting RDF was written in N-Triples format, which was compatible with the semantic databases.

One of the side effects of converting relational data into RDF is that significant data expansion occurs. First, the RDF language is verbose by design, as identifiers and relationships are expressed as URIs. Second, while a relational database implicitly relates every column in a row, every RDF relationship must be explicit. Every column value of every row of a relational table will produce an RDF triple, very similar to an entity-attribute-value database model. And finally, formats such as N-Triples and XML are generally verbose.

IV. EXPLORING THE DATASET

The aggregated data spans 629 tables, and occupies approximately 2 GB of disk space prior to augmentation. The total number of primary-foreign key relationships is 2,124. Fig. 3 shows the data distribution of the semi-synthetic dataset.

We created databases of six different sizes, in both relational and semantic forms. The relational databases shared the same schema of 629 tables, and ranged in size from 2 GB to 100 GB. The equivalent semantic databases were 5 to 850 GB in size, and contained 39 million to 5.6 billion triples. As commented earlier, it is important to note that data expansion occurs when mapping relational data into a semantic form. As an example, one SQL database of 18.3 GB expanded to a 213 GB N-Triples file: a ratio of 11.6 to 1. The benefit, however, is that the data is completely indexed, which can yield performance benefits during queries. Table 1 describes the size and characteristics of each dataset. The size refers to the raw data size in SQL form. The semantic representation (RDF format) is approximately 10 times the raw data size.

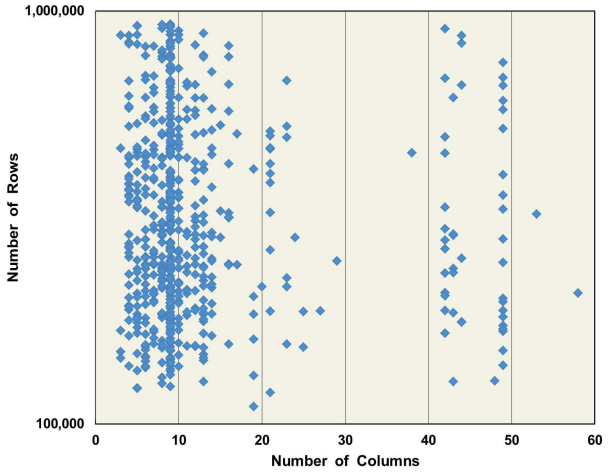

Figure 3. Data quantity distribution of 80 GB synthetic dataset. A total of 629 tables were created from data.gov in order to create a baseline dataset of 2 GB in size. The baseline dataset was used as the basis for generating the 80 GB dataset. (44134)

TABLE I. SIZES OF DATASETS GENERATED BY SYNTHETICALLY AUGMENTING DATA FROM DATA.GOV

Size	**Notes**
2 GB	Baseline data
20 GB	Between baseline - 100,000 rows / table Includes self-joins
80 GB	Between baseline – 900,000 rows/table
100 GB	Between baseline – 1,000,000 rows / table
200 GB	Between baseline - 2,000,000 rows / table
500 GB	Between baseline - 5,000,000 rows / table
1 TB	Between 1M-1B rows / table
1.6 TB	Between 1M-1B rows / table Includes self-joins

In order to explore the new semi-synthetic datasets, the semantic data was loaded into multiple semantic database engines. First, the data was loaded into Virtuoso Open Source Edition. Virtuoso runs on conventional x86 architectures. Although a SQL engine, OpenLink Virtuoso provides the necessary extensions to serve data as an RDF triplestore. Second, the data was loaded into the YarcData Urika appliance. The Urika appliance is an RDF triplestore semantic database built on top of Cray XMT2 hardware. The Cray XMT2 is a massively multithreaded memory-latency tolerant system with large shared global memory [17]. The Cray XMT2 at Mayo is a 2 TB RAM system with 64 nodes (8,192 threads) and a 20 TB Lustre file system. The Urika Software version used in testing was v 0.9. Third, the data was loaded into SPEED-MT, an open source platform developed by Sandia National Labs [18], which also runs on the Cray XMT2. After loading the database, a collection of 2,000 random queries were constructed. Queries were constructed as a sequential series of SQL joins by generating random walks between tables in the fully connected relational database

schema, as shown in Fig 4. The SQL queries were converted into SPARQL queries and executed through a Jena interface [19] which provides a web-enabled interface to the Urika semantic database.

In prior work, we demonstrated that query time varied based on complexity; however, we also observed substantial variability in Virtuoso execution time for queries of similar complexity [2]. In order to better understand the variability, we computed all of the instances of "in edges" and "out edges" of the semantic graph schema for two queries with seven joins but a 20-fold difference in execution time. The analysis of the first query found that the number of edges ranged from 4,600 to 100,670. For the second query of the same complexity, the smallest edge count was 39, with a correspondingly shorter time required to execute the query. This natural variability in both the schema and the data, which exposes variability in query performance, may not exist in purely synthetic benchmark datasets.

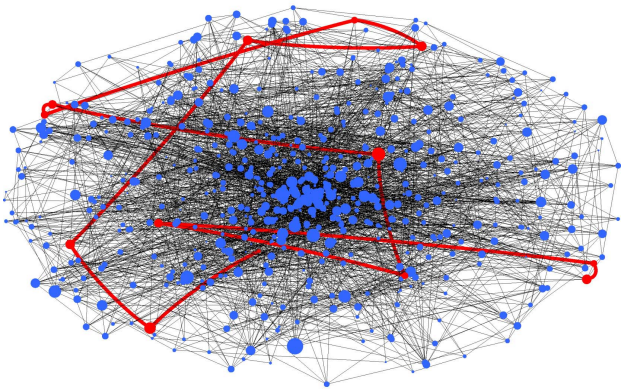

Figure 4. Linear Query mapped onto graph depiction of relational database, where each vertex represents a table, and the size of each vertex corresponds to the number of foreign key linkages. (43617)

In a separate analysis, a progression of queries was generated. A single join query between two related tables was constructed. Additional joins were added to progressively increase query complexity. Because the semi-synthetic database includes corresponding relational and semantic representations, queries can be executed on both relation and semantic databases. Using the 100 GB database, the query set was executed on three hardware/software platforms. MySQL Server 5.5 for Linux was installed on a Dell PowerEdge M610 which is a dual socket Xeon (12 core) system, running at 2.93 GHz with 96 GB DRAM and dual 15K RPM SAS drives in a RAID 0 configuration. Using the Cray XMT2, the semantic datasets were loaded into both the YarcData Urika semantic database and SPEED-MT. Fig. 5 shows the query execution time on all three platforms.

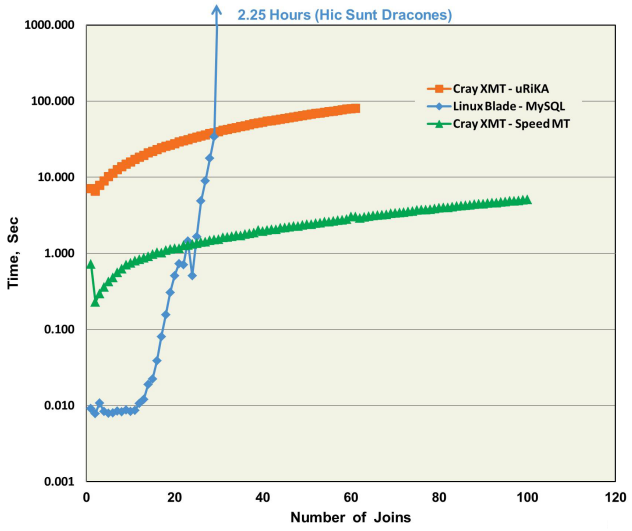

Figure 5. Comparison of Semantic platform compared to relational platform using 100GB dataset. Using the number of table joins as a measure of complexity, the MySQL platform outperforms the semantic hardware/software platform until 28 joins. For complex queries greater than 28 joins, the MySQL platform is unable to complete the query in a reasonable amount of time. (44135) Derived from [2]

Although preliminary, some trends can be observed in this data. While the semantic graph query time increases with complexity, the RDBMS has a very different performance characteristic. The relational system is quite fast for less complex queries; however there appears to be a point of complexity above which the RDBMS does not perform well. Neither the semantic nor the relational database systems were optimized for this experiment. Thus, the results should not be construed as a competitive benchmark. However, the result does indicate a difference in the two systems when scaling one dimension of complexity.

V. CONCLUSIONS

The proposed semi-synthetic database is unique compared to other open databases that have been published. First and foremost, the data characteristics of the database mimic that of a large medical center data warehouse. Thus, the complexity of the dataset can serve as a benchmark surrogate to real-world data. Using real data as the source tables, we were able to demonstrate realistic distributions of data values. Additionally, the creation of new records follows the single and joint distribution properties of the real-world data. By superficially linking the data, we were able to build complex (albeit arbitrary) relationships. Lastly, the data and the methods are constructed from open data which is easily available to other investigators. Code to collect and augment the data, as well as the databases themselves, are available at http://datahub.io/organization/semi-synthetic-benchmark.

Preliminary analyses of the data demonstrated the power of this semi-synthetic data source. Unlike other benchmark

datasets, there is natural variation in the relationship between nodes, which is well-suited for developing global optimization strategies for query engines. Another difference between this dataset and existing benchmarks is the concurrent representations in SQL and RDF, thereby allowing engineers and scientists to compare two fundamentally different approaches to data storage and access. When paired with optimized hardware, the use of this dataset can inform a strategy for using one approach versus another. Traditional database approaches are well suited for some problems; however, a semantic database on optimized hardware is well-suited for large-memory, complex, irregular datasets.

Further characterization of this dataset could proceed in several directions. The synthetic queries discussed here were primarily focused on foreign key table joins, but a more advanced study would include data columns and operations such as filtering and aggregation. Additional characterization could be done to assess the global structure created by the synthesized foreign keys, evaluating similarities to real-world datasets in terms of clustering or partitioning.

There is considerable interest in the collection and analysis of "big data." Several approaches are available include traditional RDBMS and Semantic graphs. Both have appropriate use cases. In support of big data, HPC companies are developing new architectures to facilitate the storage and analysis of datasets. In order to optimize hardware as well as to develop new software analytics, it is important to have an open dataset which real-world complexity. We have developed an open, semi-synthetic, large, complex dataset for use in the optimization and benchmarking of "big data" technologies.

REFERENCES

[1] 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, pp. 131-135, March 1, 2010 2010.

[2] R. Techentin, D. Foti, S. Al-Saffar, P. Li, E. Daniel, B. Gilbert, and D. Holmes, "Development of a Semi-Synthetic Dataset as a Testbed for Big-Data Semantic Analytics," presented at the IEEE International Conference on Semantic Computing, Newport Beach, CA, 2014.

[3] R. McCool, "Rethinking the semantic web. Part 2," *Internet Computing, IEEE,* vol. 10, pp. 93-96, 2006.

[4] C. Yang, C. Yen, C. Tan, and S. R. Madden, "Osprey: Implementing MapReduce-style fault tolerance in a shared-nothing distributed database," in *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, 2010, pp. 657-668.

[5] D. Mizell and K. Maschhoff, "Early experiences with large-scale Cray XMT systems," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, 2009, pp. 1-9.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american,* vol. 284, pp. 28-37, 2001.

[7] Y. Guo, Z. Pan, and J. Heflin, "LUBM: A benchmark for OWL knowledge base systems," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 3, pp. 158-182, 2005.

[8] C. Bizer and A. Schultz, "The berlin sparql benchmark," *International Journal on Semantic Web and Information Systems (IJSWIS),* vol. 5, pp. 1-24, 2009.

[9] S. Al-Saffar, C. Joslyn, and A. Chappell, "Structure discovery in large semantic graphs using extant ontological scaling and descriptive semantics," in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, 2011, pp. 211-218.

[10] B. Chen, X. Dong, D. Jiao, H. Wang, Q. Zhu, Y. Ding, and D. Wild, "Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data," *BMC bioinformatics,* vol. 11, p. 255, 2010.

[11] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, and M. Magrane, "The universal protein resource (UniProt)," *Nucleic acids research,* vol. 33, pp. D154-D159, 2005.

[12] S. Cerutti, "Semantic models in medical record data-bases," *Informatics for Health and Social Care,* vol. 5, pp. 215-226, 1980.

[13] J. M. Fisk, P. Mutalik, F. W. Levin, J. Erdos, C. Taylor, and P. Nadkarni, "Integrating query of relational and textual data in clinical databases: a case study," *Journal of the American Medical Informatics Association,* vol. 10, pp. 21-38, 2003.

[14] S. Webster, M. Morgan, and G. O. Barnett, "Medical Query Language: improved access to MUMPS databases," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 1987, p. 306.

[15] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks, "The semantic web: The roles of XML and RDF," *Internet Computing, IEEE,* vol. 4, pp. 63-73, 2000.

[16] C. Bizer, "D2r map-a database to rdf mapping language," 2003.

[17] P. Konecny, "Introducing the Cray XMT," in *Proc. Cray User Group meeting (CUG 2007)*, 2007.

[18] E. L. Goodman, E. Jimenez, D. Mizell, S. Al-Saffar, B. Adolf, and D. Haglin, "High-performance computing applied to semantic databases," in *The Semanic Web: Research and Applications*, ed: Springer, 2011, pp. 31-45.

[19] C. Trim. (2013). *Jena: A Semantic Web Framework* Available: http://trimc-nlp.blogspot.com/2013/06/introduction-to-jena.html