Building Ontology for Big Data in Column-oriented NoSQL Database

Nang Kham Soe, Tin Tin Yee, Ei Chaw Htoon
University of Information Technology
Yangon, Myanmar
nangkhamsoe@uit.edu.mm, tintinyee@uit.edu.mm, eichawhtoon@uit.edu.mm

Abstract

Big data are usually integrated data from different sources in a structured, semi-structured and unstructured data collection. NoSQL databases are in the base of big data storage. These databases suffer from lack of semantics since they are able to handle unstructured data. In order to solve that issue, an ontology-based representation of the information stored in NoSQL databases is highly needed. This paper proposes an approach to construct an ontological form of big data stored in column-oriented NoSQL database namely Cassandra. The approach defines mapping rules and builds ontology by applying these rules.

Keywords- Ontology, Semantic Web, Big data, NoSQL, Cassandra

1. Introduction

Big data was defined as "datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze." Big data often ranges from a few dozen terabytes (TB: approximately 1012 bytes) to multiple petabytes (PB: approximately 1015 bytes) [5]. Big data is often represented by large amounts of high-dimensional and poorly structured or organized forms when the data is typically generated from heterogeneous sources. It can be either structured (e.g. spreadsheets, relational databases), unstructured (e.g. text, image), and/or semi-structured such as radio frequency identification (RFID) data and extensible markup language (XML) data [9].

NoSQL databases are in the base of big data storage. They permit to store large volumes of structured, semi-structured, and unstructured data. These databases suffer from lack of semantics since they are able to handle unstructured data. Cassandra is a type of column-oriented NoSQL database. It has no fixed schema. In this paper, an ontology based approach is proposed to extract hidden semantics from schema-less data store. The proposed approach defines mapping rules to represent a data source as an ontological form. The proposed system uses OWL (Web Ontology Language) to describe ontology as the target schema and data source corresponds to column-oriented database namely Cassandra. The system consists of three steps: understanding data source model, defining mapping rules, and building ontology.

The rest of the paper is organized as follows. Section 2 introduces about Cassandra database. Section 3 describes about ontology. Related works are presented in Section 4, and we explain our proposed system in Section 5. Finally, we draw conclusions in Section 6.

2. Column-oriented NoSQL DB: Cassandra

The concept of non-relational databases (NoSQL DBs) refers to a database alternative to the relational model that arranges data discretely into tables of columns and rows. NoSQL DBs are commonly associated with more flexible deployment, high read/write performance as well as scaling to very large data sets. There can be distinguished four different kinds of NoSQL DBs [10]: document databases (examples: Couchbase, MongoDB), key-value stores (examples: Riak, Amazon's Dynamo), graph databases (examples: InfoGrid, Infinite Graph, Neo4J), and column-oriented stores (examples: HBase, Cassandra).

Apache Cassandra is the type of column oriented NoSQL database. It is a free and open-source , a highly scalable, high-performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.

The data model of Cassandra is wide column store, and, as such, essentially a hybrid between a key-value and a tabular database management system. Rows are organized into tables; the first component of a table's primary key is the partition key; within a partition, rows are clustered by the remaining columns of the key.

A column family or a super column family (called "table" since CQL 3) resembles a table in an RDBMS. Column families contain rows and columns. A super column family consists of a row key and a number of super columns. Each row is uniquely identified by a row key. Each row has multiple columns, each of which has a name, value, and a timestamp. Unlike a table in an RDBMS, different rows in the same column family do not have to share the same set of columns, and a column may be added to one or multiple rows at any time. In

ISBN 978-99971-0-585-1

Cassandra, it also has special column called super column that stored a map of sub-columns. The data model of Cassandra is as shown in Figure 1.

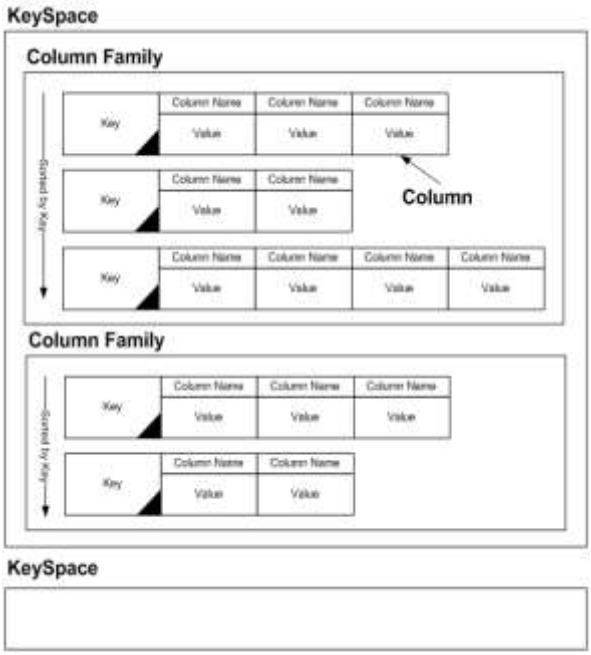

Figure 1. Cassandra data model

3. Ontology

Ontology [6] is the core of the Semantic Web technology, which is also an important method to conceptualize domain knowledge and modeling that can be used to describe the semantics of the data. The potential role of semantics in big data [7] is to describe data source 'schemas using ontologies. Ontologies, as formal models of representation with explicitly defined concepts and named relationships linking them. To make data schematically comparable, for each one of data sources would be represented by its local ontology. In NoSQL databases, parts of structure definition are mixed with the data (e.g..Keys in key-value format) and some are unstructuredness. The lack of formal schema in NoSQL databases can be compensated by ontologies. There are different languages to describe the ontology. Different ontology languages provide different facilities. OWL is the World Wide Web Consortium (W3C) recommended standard ontology description language.

There are three main components in OWL [4]: Classes, Properties and Individuals. Classes are a concrete representation of concepts. Classes may be organized into a superclass-subclass hierarchy, which is also known as taxonomy. OWL Properties represent relationships. There are two main types of properties, Object properties and Datatype properties. Object properties are relationships between two individuals. Datatype properties describe relationships between individuals and data values. Individuals represent objects in the domain and are also known as instances. Individuals can be referred to as being 'instances of classes'.

4. Related Works

Abbes et. al intended to develop a novel approach for ontology learning from a NOSQL database to manage big data about a specific domain [1]. The paper introduced links between ontologies and Big Data. They presented transformation rules for building OWL ontologies from NOSQL database. They transform all possible cases in the MongoDB database into ontological constructs. It is divided into four main steps to learn ontology from MongoDB. The first step is the creation of the ontology skeleton. It consists of class definition and subsumption relation detection between classes. The second step is to learn objectProperties and dataTypeProperties. In the third step, individuals are identified. Finally, in the fourth step, it deduces axioms and constraints.

Abbes et. al proposed ontology based big data integration approach based on a NoSQL database, namely MongoDB and modular ontologies. It follows three steps: wrapping data sources to MongoDB databases, generating local ontology corresponding to each data source, and composing the local ontologies to get a global one. The target schema is represented as OWL ontology [2].

In [8], the authors proposed a semantic Extract-Transform-Load (ETL) framework for big data integration. The proposed framework uses semantic technologies to integrate and publish data from multiple sources as open linked data. The use of semantic technologies is introduced in the Transform phase of an ETL process to create a semantic data model and generate semantically linked data (RDF triples) to be stored in a data mart or a data warehouse.

Kiran et. al presented ontology based semantic integration system for a column-oriented NoSQL data store like HBase [3]. It follows three steps: schema extraction for each data source based on two methods: On-Line schema generation and Offline schema generation, schema to ontology conversion and ontology alignment-merging. The proposed system represents RDF triples as the target schema.

5. Proposed System

In big data environment, data come from different data sources. NoSQL databases are used as big data stores. These databases suffer from lack of semantics. In order to solve the issue, ontology based approach to extract hidden semantics from big data is proposed. The objective of the proposed system is to later construct an ontological form

(global ontology) for big data from different data stores. As the first step, this paper mainly focuses on building local ontologies for each of data stores. The proposed system deals with Cassandra data stores as input. The system consists of three steps: understanding the data source model, defining mapping rules, and building ontology as shown in Figure 2.

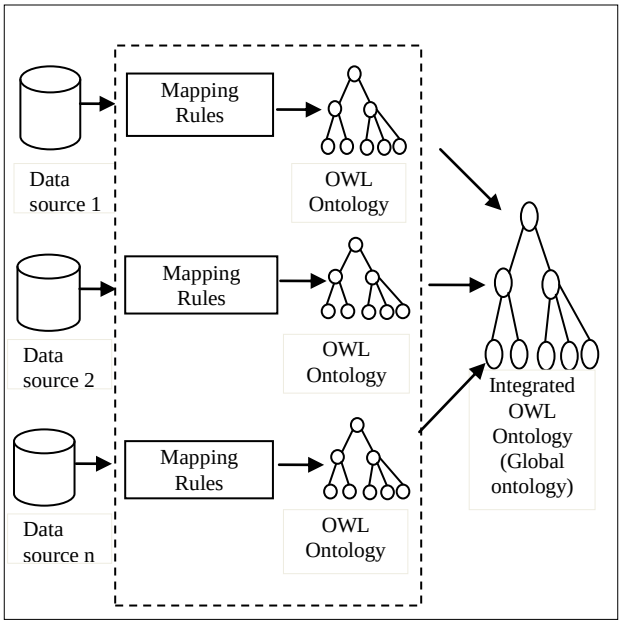

Figure 2. Proposed system architecture

5.1. Understanding Data Source Model

The proposed system uses NoSQL database, namely Cassandra. As mentioned in Section 2.1, outermost container for data in Cassandra is keyspace. Each keyspace has at least one and often many column families. A column family is a container of a collection of rows. Each row contains ordered columns. In Cassandra, it has special columns called super columns that stored a map of sub-columns.

5.2. Defining Mapping Rules

To generate semantics from Cassandra database, it is mapped to the OWL ontology by means of the following rules.

Rule 1: Defining Class

Each column family in the keyspace is transformed into OWL class.

Rule 2: Defining Datatype Property

The value of column may have different data types, namely basic data types (int, string, etc.), collection types (map, set, list) and user-defined data types. If the column has basic data types, that column is transformed into a dataTypeProperty for which "domain" the class corresponding to the column family that contains that column.

Rule 3: Defining Object Property

If the column has collection types or user-defined data types, it is transformed into an object property for which "domain" the class corresponding to the column family that contains the column and "range" is the class corresponding to the collection or user-defined type.

Rule 4: Defining Individuals

The value of each column in the row is transformed into individuals in the ontology.

Rule 5: Defining Axioms

If two different rows in the column family contain the same set of data values for all columns, it deduces two different classes corresponding to these two different rows are equivalent. Otherwise, it deduces these two different rows are disjoint.

5.3. Building OWL Ontology

As a test dataset, students' attendance data of a university is used to show ontology conversion from data stored in Cassandra. The dataset consists of four column families (tables) as depicted in Figure 3.

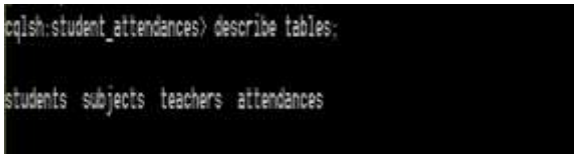

Figure 3. Test cassandra dataset

According to Rule 1, all column families are transformed to ontology class.

<owl:Class rdf:ID="attendances"/>
<owl:Class rdf:ID="students"/>
<owl:Class rdf:ID="teachers"/>
<owl:Class rdf:ID="subjects"/>

According to Rule 2, the column that has basic data types is transformed into a dataTypeProperty. The following only gives dataType property transformation for the column family "students" due to space limitation. The internal structure of "students" is as shown in Figure 4. Three columns have basic data type and the column "email" has collection datatype.

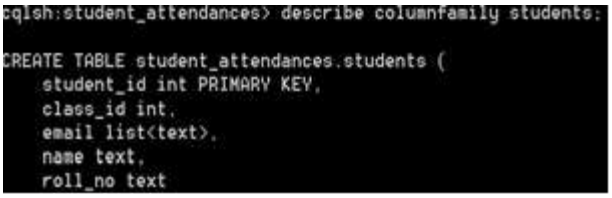

Figure 4. Column family "students"

```
<owl:DatatypeProperty rdf:ID="roll_no">
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema # string"/>
<rdfs:domain rdf:resource="#students"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="class_id">
 <rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#integer"/>
<rdfs:domain rdf:resource="#students"/>
</owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="name">
 <rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#string"/>
<rdfs:domain rdf:resource="#students"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="class_id">
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#integer"/>
<rdfs:domain rdf:resource="#students"/>
</owl:DatatypeProperty>
```

According to the Rule 3, the column "email" in the column family "students" is transformed into object property.

```
<owl:ObjectProperty rdf:ID="hasemail">
   <rdfs:domain rdf:resource="#students"/>
   <rdfs:range rdf:resource="#email"/>
 </owl:ObjectProperty>
```

The individual is an instance of the class in the ontology. Accordance with Rule 5, the value of each column is transformed to individual in the ontology. The following only gives individual transformation for one record of the column family "students" due to space limitation.

```
<students:student_id  rdf:about="#82">
    < students:roll_no>5SE-1 </ students:roll_no>
    < students:class_id>4</ students: class_id >
    < students:name>Juzi Hein </ students:name>
    < students:email>juzihein@uit.edu.mm
    </ students: email >
  </ students:students_id>
```

6. Conclusion

In big data environment, data come from different data sources. NOSQL databases are in the base of storage of big data. These databases suffer from lack of semantics since they are able to handle unstructured data. Therefore, an ontology-based representation of the information stored in NOSQL databases is highly needed. The paper intends to develop an approach for building ontology from big data. The proposed system mainly focuses on hidden semantics extraction from data stored in column-oriented database namely Cassandra.

The next step of our research tries to build ontologies for different data model and focuses on semantic heterogeneity problem in the big data integration process.

7. References

[1] Abbes, H., Boukettaya, S., Gargouri, F, "Learning ontology from Big Data through MongoDB database." , Proceedings of IEEE/ACS 12th International Conference of Computer Systems and Applications, 2015, pp. 1–7.

[2]Abbes, H., Gargouri, .F, "Big Data Integration: a MongoDB Database and Modular Ontologies based Approach", 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems (KES2016), 5-7 September 2016.

[3] Kiran V K, Vijayakumar R, "Ontology Based Data Integration of NoSQL Datastores", 9th International Conference on Industrial and Information Systems (ICIIS), 15-17 Dec. 2014

[4] Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, Chris Wroe, "A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.1"

[5] Minelli, M., Chambers, M. & Dhiraj, A., "Big data, big analytics: emerging business intelligence and analytic trends for today's businesses." , 2012.

[6] Li Kang, Li Yi, LIU Dong, " Research on Construction Methods of Big Data Semantic Model", Proceedings of the World Congress on Engineering 2014 Vol I,WCE 2014, July 2 - 4, 2014.

[7] Marina P., Boris V., "Semantic Web Technologies and Big Data Warehousing", MIPRO, May 21-25, 2018.

[8] Srividya K Bansal, "Towards a Semantic Extract-Transform-Load (ETL) Framework for Big Data Integration," IEEE International Congress on Big Data, 2014.

[9] Wang Lidong, Randy Jones, "Big Data Analytics for Disparate Data", American Journal of Intelligent Systems, 7(2): 39-46 (2017).

[10] Woodey A., "Forrester Ranks the NoSQL Database Vendors. Datanami", 2014. Available: http://www.datanami.com/2014/10/03/forrester-ranks-nosql-database-vendors/.