Data Integration for Books Data using Graph Database

Darshana Shimpi
Dept. of IT
International Institute of
Information Technology
Pune

ABSTRACT
The modern developments through the internet have the opportunities to access heterogeneous information system any where in the world. We can integrate information which is structured or unstructured and deliver it to any system on big data platform. The information may be from heterogeneous sources and with different representations. In this paper, we propose an integration scheme for books data from different sources over the web.

Graph database allow simple and rapid retrieval of complex hierarchical structures that are difficult to model in relational systems. Nowadyas graphs have become very popular in various domains like social media analytics, healthcare, chemistry, business intelligence, networking and many more.

We have proposed a system that integrate data of books from various sources across the web. Here graph database is used named TITAN. We can easily retrieve complex information using simple queries using Gremlin, a graph query language.

General Terms
Databases, Heterogeneous sources, Graph query language engines.

Keywords
Heterogeneous sources, graph database, information integration,graph query

1. INTRODUCTION
Graph database is a database that uses graph structure for semantic queries with nodes, edges and properties to represent and store data. Graph structures could be represented in network model databases. Both represent general graphs, but network model databases operate on lower level of abstraction and lack easy traversal. Compared with relational databases, graph databases are very faster for associative data sets. They can scale more naturally to large data sets because they do not require expensive join operations. Graph databases are with less rigid schema so they are more suitable to manage ad hoc and changing dta with evolving schema. Graph databases are a powerful tool for graph like queries, for example computing the shortest path between the two nodes in the graph. Graph databases directly store the relationships between records. Graph databases have become more popular for a variety of uses from modeling online code repositories to tracking software engineering dependencies. Graph databases provides a level of flexibility and resilience that is a great match for todays fast-moving business and agile development methods. On the other hand, relational databases, need to map schema in detail to determine how things are connected, and then to retrieve related data records. When the database become larger then response time slows down and hence affects the business growth. However, with graph databse, traversal speed remains constant because it does not depend on the total amount of data stored. Graph database uses a technique called a index-free adjacency. I simple terms, this means that each data element points directly to its inbound and outbound relationships, which point directly to related nodes. This technique allows million of related records to be traversed per second.

Nowadays there are various sources of information, but they are not similar it is from heterogeneous sources. The data in these sources are in different formats so there is a need to integrate that data to makre it in uniform format; sothat user can easily retrieve the data of his need . One of the advantage of information integration is that the user of system obtains complete and concise overview of all existing data without needing to access all data sources separately. Information integration possesses many applications from portals integrating data from multiple sources, comparison shopping, electronics market places to medical genetics in integrating genomics data, Astrophysics to monitor the event in the sky and several other areas of enterprise data integration. The late 1990s have witnessed huge proliferation of electronically accessible information that has led to a great deal of research and development in information retrieval to help users so that they can search and quickly retrieve relevant and meaningful information [1]. The immensity of web data valuable for various human needs has led to research on information extraction from the

web [2]. With more and more information sources available via inexpensive network connections over the Internet, the desire to organize this information in a user friendly way has been the motivation behind the work presented here [3]. Much of the websites available over the web do not serve all the information on one platter so there is a need to extract information from different sources and apply information integration techniques to populate the data onto a common store.

In this paper, we propose a system that integrates distributed and heterogeneous information of books from various data sources across the web. We have developed a prototype that integrates information from different sources like **iblist.com, reddit, Goodreads, amazon.com, shelfari.com and pubmed.com** The different components of the above data sources are deployed using a graph model. This graph model is queried and information is retrieved from it in efficient way. The contents of the paper are organized as follows: section 2 contains related work done in the field of information integration na sgarph databases. Section 3 presents the proposed structure of our system. Section 4 gives results drawn from analysis of observations. Section 5 gives the conclusion and future aspects related with the work.

2. RELATED WORK
Many established practices exist for the traditional discipline of Data Fusion. However there is a need to expand the fusion

view from sensor networks [5], intrusion detection [8], [7], collaborative alerts correlation [12], anomaly detection [13] to integration of information available on the web on to a common store. Jiangfan et al. [9] investigated the cognitive model of spatial information, the conceptual model of space phenomena, and the logic of spatial data management mode, leading to the integration of various spatial data sets from the view of model integration. Furthermore, they provided a method for the effective storage and management of massive and multisource heterogeneous spatial data. Zhao et al. [07] proposed a new computer information security protection system based on data fusion theory. Multiple detection measures were fused in their system, so that it has lower false negatives rate and false positive rate as well as better scalabilities and robust. Their experiments proved that by fusing, the final detected rate of the entire system to every intrusions was higher than or close to the best detected rate of all the basic detectors to them, which makes the entire system has a relative high detected rate to all intrusions, and makes up for the flaw that single detecting method cannot has good detecting result to all intrusions. Zhuang et al. [8] presented an efficient and effective model for collaborative alerts analyzing. Their system enhanced the alert verification using assets contextual information. By applying alert fusion and using a precisely denned knowledge base in the correlation phase, they also provides a method to get general and synthetic alerts from the large volume of elementary alerts. They discussed the similarity function of alerts and proposed an algorithm which can do correlation among alerts from different security tools. Kirk et al. [10] proposed Information Mainfold (TM) as a novel framework for browsing and querying of multiple networked information sources. A system that demonstrated the viability of knowledge representation technology for retrieval and organization of information from structured and unstructured sources was presented in their work. Arens et al. [14] described a system that addresses the important problem of how to efficiently integrate information from multiple databases. They emphasized on the query planning problem i.e. the selection of appropriate data sources and ordering the access to them. They also stressed on reformation of queries i.e. the use of knowledge both about the domain and the databases to modify queries to make retrieval plans for them more efficiently. Vicknair et al. [16] performed a comparison of the relative usefulness of the relational database MySQL and the graph database Neo4j to store graph data. The goal of this study was to determine whether a traditional relational database system like MySQL, or a graph database, such as Neo4j, would be more effective as the underlying technology for the development of a data provenance system. Angles et al. [17] performed a survey study of different graph database models and analysed its performance and utility when compared with other database models. Applications that are successfully modelled using graph were also discussed in their work. Bordoloi et al. [18] described a method for transforming a relational database to a graph database model. In this approach, the dependency graphs for the entities in the system are transformed into star graphs. This star graph model is transformed into a hyper graph model for the relational database, which, in turn, can be used to develop the domain relationship model that can be converted into a graph database model.

Thus using the erstwhile concept of Information Mainfold and taking into account the advantages of graph databases over the relational one, this paper gives a method to integrate the data of books from various sources over the web and intends to use a graph data model to store and access the obtained integrated data. Here we have deployed the knowledge graph model unlike the existing information integration systems. The knowledge graph model is more robust and generic way to graphically store and analyse the data.

3. PROPOSED TECHNIQUE

With the advent of Internet over the years, most of the information is available on web in a well defined way, but on different websites. For retrieval and analysis, this information needs to be integrated on a common platform. We propose a technique to populate a database of books in form of a graph by gathering information from various sources on the web. We expose this graph database via some APIs and graph processing languages to extract useful information which is further queried and analysed. We follow a standard 3 step data integration process [12] as shown in Fig. 1. Firstly, we need to identify corresponding attributes that are used to describe the information items in the source. The result of this step is schema mapping, which is used to transform the data present in the sources into a common representation. Secondly, the different objects that are described in the data sources need to be identified and aligned. In this way, multiple, possibly inconsistent representations of the same real world objects are found using duplicate detection techniques.

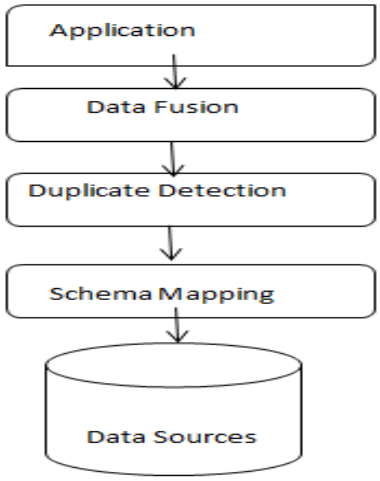

Fig. 1 Data Integration Process

In the last step, the data obtained is fused into a single representation, while resolving the inconsistencies in the data. This last step is referred to as Data Integration.

A. Technical Process

Based on the Data Integration process, we propose the following procedure that builds on top of it, and is described in Fig. 2. First, data stores, are selected and raw dumps of the data **iblist.com, reddit, Goodreads, amazon.com, shelfari.com and pubmed.com** are obtained using urllib [20] package for python and other commands viz. Wget [21] and cURL [22]. The obtained data is parsed and refined further. The refined data is further processed and a key value based data structure is obtained which is uploaded over a graph database.

B. Sources of Data

A large amount of data is available on the web for various books. We however restricted our work to four major sources:

1) **amazon.com**: The Internet shopping site which gives Database for information related to books.

2) **Goodreads:** It is social cataloging site. The website allows individuals to freely search Goodreads' extensive user-populated database of books, annotations, and reviews.

3) **Pubmed.com :** is a free search engine accessing primarily the mediline database of references and abstracts on life sciences and biomedical topics.

4) **shelfari.com :** Shelfari users build virtual bookshelves of the titles they own or have read, and can rate, review, tag, and discuss their books.

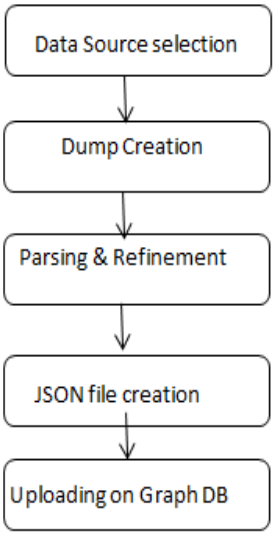

Fig. 2 Technical Process

C. Data Collection

Different commands and APIs are used to generate dumps of data from these various sources.

1) urllib [20]: It is a package that collects several modules for working with URLs.

2) Wget [21]: GNU Wget is an open source software package for retrieving files using HTTP, HTTPS

and FTP, the most widely-used Internet protocols. It is a non-interactive command line tool, so it may easily be called from scripts, cron jobs, terminals without X-Windows support, etc.

3) cURL [22]: cURL is a tool to transfer data from or to a server, using one of the supported protocols (HTTP, HTTPS, FTP, GOPHER, DICT, TELNET, LDAP or FILE). The command is designed to work without user interaction.

D. Data Parsing and Refinement

There are various heterogeneous sources , so we need to accumulate raw dumps of data from these sources using specific APIs and commands, the data is parsed and refined into a noise free format. Using the concept of regular expressions, suitable data structures, and other programming concepts in Python 2.7.2, data is obtained in tag free format. Pythons built-in re [24] module is used for parsing which provides excellent support for regular expressions. The re module raises the exception re.error if an error occurs while compiling or using a regular expression. Functions like matching functions re.match(), re.findall(), search function re.search(), search and replace function re.sub(), split function re.split(), compile function re.compileO from this module are used for parsing the raw dumps.

E. JSON File Creation

The refined data is converted into a key value data structure which is stored in Python ordered dictionary OrderdDict. There are three keys in this ordered dictionary namely mode, vertices and edges. The value part of the vertices and edges key is also a dictionary. Thus it forms a nested dictionary type of data structure. Using JSON module from Python [28], this dictionary based data store is stored in a properly formatted JSON file [25]. The JSON file thus obtained is uploaded on to TITAN [26], a graph database via Gremlin, a graph query language [27].

F. Uploading on Graph Database

A set of gremlin queries for uploading a JSON file have been shown in Fig. 3 where storage backend is HBase, a non-relational, distributed database [28] and single machine environment is used for testing purpose.

gremlin> config = new BaseConfigurationf)

==>org.apache.corrons.configuration.BaseConfiguration@lb9 f7e8

gremlin> config.setProperty("storage.backend",”hbase”)

==>null

gremlin> config.setProperty(“storage.hostname",”127.0.0.1”)

—>null

gremlin>g= new TinkerGraph()

==>tinkergraph[vertices:0 edges:0]

gremliln>g. loadGraphSON('sanple. j son')

==>null

Fig. 3: Uploading JSON

Graph database refers to any storage system that can contain, represent and query a graph consisting of a set of vertices and a set of edges relating to a pair of vertices. Titan [26] is a scalable graph database optimized for storing and querying graphs containing hundreds of billions of vertices and edges distributed across a multi-machine cluster. It is open source with the liberal Apache 2 license. It supports storage Backends like Apache HBase [28], Apache Cassandra [29], Oracle BerkeleyDB [30]. Titan is also a transactional database and can support thousands of concurrent users executing complex graph traversals.

4. RESULTS

The system thus developed is experimented with some popular books taken from amazon. Every book is represented as vertex in this database and is associated with attributes like author, price, publication, genre, rating, reader reviews, critic reviews etc. These attributes would need a number of relational tables to store the data to prevent anamolies and to maintain the database integrity. In a relational database model, we apply multiple JOIN queries to obtain the results which makes the system slow and less efficient. To overcome this, the presented approach stores complete data in a single graph with the relationships expressed as edges and supports

all kinds of complex querying and traversal in linear time in a much efficient and hastle free way. In the following presented gremlin queries g represents the graph, V represents Vertices, E represents Edges, outE represents out edges from a vertex, inV represents vertex to which an edge points , inE represents the incoming edges of a vertex. Heavy join operations even with full indexing. On the other hand, a graph database suited for this type of query processing avoids explicit intermediate and dummy table creation. The benefit of the graph database comes from not only its direct data representation and storage, but also its intuitive query structure. The compact representation of the graph query facilitated the management, user validation and exploration of the analytic intent of the query.

5. CONCLUSION AND FUTURE WORK

We have developed a system that populates a graph database by integrating data of books from different sources over the web. We thus obtained a graph database for books by integrating information like author, price, publication, genre, rating, reader reviews, critic reviews etc. from four sources namely amazon.com, pubmed.com, Goodreads, shelfari.com. Putting together the basic concept of information integration along with our proposed technique, we thus analysed and presented a graphical way of efficiently storing the data. The work opens many areas where it can further be extended. An application can be build on top of this data store that could be used as a Meta Search Engine for movies and with use of Machine Learning Algorithms, a recommender system could also be proposed.

6. REFERENCES

[1] Romano, N.C., Roussinov, D., Nunamaker, J.F., Chen, H.: Collaborative Information Retrieval Environment: Integration of Information Retrieval with Group Support Systems. In: Proc. of the 32nd Hawaii InternationalConference on System Sciences. . Maui, pp. 5 8. Society Press, Hawaii (1999)

[2] Chidlovskii, B.: Information Extraction from Tree Documents by Learning Subtree Delimiters. In: Proc. UCAI-03 Workshop on Information Integration on the Web (IIWeb-03), pp. 38. ACM Digital Library, Acapulco, Mexico (2003)

[3] Bleiholder, J., Naumann, F.: Data fusion. ACM Computing Surveys 41(1), 141 (2008)

[4] Akiba Persistit, https://github.com/pbeaman/persistit

[5] Jiangfan, F., Wei, W.: Multiple Spatial Model Fusion in Heterogeneous Sensor Networks. International Journal of Multimedia and Ubiquitous Engineering 9(2), 114 (2014)

[6] Bass, T: Intrusion detection systems and multisensor data fusion. Communications of the ACM 43(4), 99105 (2000)

[7] Zhao, X., Jiang, H., Jiano, L.: A Data Fusion Based Intrusion Detection Model. In: First International Workshop on Education Technology and Computer Science, pp. 10171021. IEEE Press, Wuhan, China (2009)

[8] Zhuang, X., Xiao, D., Xuejiao, L., Zhang, Y.: Applying Data Fusion in Collaborative Alerts Correlation. In: International Symposium on Computer Science and Computational Technology, pp. 124127. IEEE Press, Shanghai, China (2008)

[9] Chatzigiannakis, V., Androulidakis, G., Pelechrinis, K., Papavassiliou, S., Maglaris, V.: Data fusion algorithms for network anomaly detection:classification and evaluation. In: Third International Conference on Networking and Services (ICNS07), pp. 5051. IEEE Press, Athens, (2007)

[10] Kirk, T, Levy, A.V., Sagiv, Y., Srivastava, D.: The Information Mainfold. In: Proc. of the AAAI 1995 Spring Symp. on Information Gathering from Heterogeneous, Distributed Enviroments, pp. 8591. AAAI Digital Library, Palo Alto, California, USA (1995)

[11] Levy, A.Y, Rajaraman, A., Ordille.J.: Querying Heterogeneous Information Sources Using Source Descriptions. In: Proc. of 22th International Conference on Very Large Databases, pp. 251262. Morgan Kaufmann, Bombay, India (1996)

[12] Halevy, A., Rajaraman, A., Ordille,J.: Data Integration: The Teenage Years. In: Proc. of the 32nd international conference on Very large Databases, pp. 916. ACM Digital Library, Seoul, Korea (2006)

[13] Ciravegma, F.: Integrating Information to Bootstrap Information Extraction from Web Sites. In: Proc. UCAI-03 Workshop on Information Integration on the Web (IIWeb-03), pp. 914. ACM Digital Library, Acapulco, Mexico (2003)

[14] Arens, Y, Knoblock, C.A.: Planning and Reforming Queries for Semantically-Modeled Multidatabase Systems. In: Proc. of the Second International Conference on Information and Knowledge Management, pp. 423432. ACM Digital Library, Washington, DC, USA (1993)

[15] Januja, N.K., Hussain, F.K., Hussain, O.K.: Semantic information and knowledge integration through argumentative reasoning to support intelligent decision making. Information Systems Frontiers 15(2), 126 (2013)

[16] Vicknair, C, Macias, M., Zhao, Z., Nan, X., Chen, Y, Wilkins, D: A comparison of a graph database and a relational database: a data provenance perspective. In: Proceedings of the 48th Annual Southeast Regional Conference (ACM SE10), pp. 16. ACM Digital Library.

[17] Angels, R., Gutierrez, C: Survey of graph database models. In: ACM Computing Surveys (CSUR), Volume 40 Issue 1, February 2008, Article No. 1.

[18] Bordoloi, S., Kalita, B.: Designing Graph Database Models from Existing Relational Databases. In: International Journal of Computer Applications (IJCA13), Volume 74Number 1.

[19] Park, Y, Shankar, M., Park, B., Ghosh, J.: Graph databases for largescale healthcare systems: A framework for efficient data management and data services. In: IEEE 30th International Conference on Data Engineering Workshops (ICDEW), 2014 ,pp. 1219

[20] Python urllib Library, https://docs.python.Org/2/library/urllib.html

[21] Wget, https://www.gnu.org/software/wget/

[22] URL project , http://cURL.haxx.se/

[23] Python re Module, https://docs.python.Org/2/library/re.html

[24] Python JSON Module, https://docs.python.Org/2/library/json.html

[25] JSON: JavaScript Object Notation, http://json.org/

[26] TITAN: Graph Database, http://thinkaurelius.github.io/titan/

[27] Gremlin Documentation, http://gremlindocs.com/

[28] Apache HBase, http://HBase.apache.org/

[29] Apache Cassandra, http://cassandra.apache.org/

[30] Oracle Berkely DB, http://www.oracle.com/technetwork/database