DATA MODELING AND DATA INTERLINKING QUERYING AN ENTERPRISE KNOWLEDGE GRAPH

M. Dharma Vardhani, M.C.A Lecturer Department of Computer Science Sri Durga Malleswara Siddhartha Mahila Kalasala, Vijayawada. (An Autonomous College in the jurisdiction of Krishna University) Reaccredited at 'A++' Grade by NAAC An ISO Certified Institution
P. Sri Bharathi, M.C.A Lecturer Department of Computer Science Sri Durga Malleswara Siddhartha Mahila Kalasala, Vijayawada. (An Autonomous College in the jurisdiction of Krishna University) Reaccredited at 'A++' Grade by NAAC An ISO Certified Institution

ABSTRACT: In This Paper One of the most significant results of the big data era is the broadening diversity of data types required to solidify data as an enterprise asset. In this paper, present Thomson Reuters' effort in developing a family of services for building and querying an enterprise knowledge graph in order to address this challenge. We first acquire data from various sources via different approaches. Furthermore, we mine useful information from the data by adopting a variety of techniques, including Named Entity Recognition and Relation Extraction; such mined information is further integrated with existing structured data (e.g., via Entity Linking techniques) in order to obtain relatively comprehensive descriptions of the entities. By modeling the data as an RDF graph model, we enable easy data management and the embedding of rich semantics in our data. Finally, in order to facilitate the querying of this mined and integrated data, i.e., the knowledge graph, we propose TR Discover, a natural language interface that allows users to ask questions of our knowledge graph in their own words.
KEYWORDS: Data Mining, Graph Theory, Knowledge Management, Natural Language Interfaces, Query Processing

INTRODUCTION

Modern businesses use tons of different apps to streamline their operations and capture enterprise data. These probably include Google Drive, Slack, Salesforce and Zendesk, to name just a few.We all know how helpful these are in boosting our day-to-day productivity, but you might not have considered the data as a treasure trove representing your enterprise.
For example, document metadata pulled from Google Drive and ingested into Neo4j can tell you what document topics are currently popular, which projects are getting lots of attention and who the Google Drive super users (i.e. people who review/edit lots of documents) of your organization are. The possibilities here are endless, and the power of your analysis is limited only by the amount/quality of your data.
Additional business insights include similarity of documents, identification of reusable content, skills inventory of employees, customer projects, technologies used, skills required and much more.

PROBLEM STATEMENT

One of the major reasons is the lack of any standardization in the format and vocabulary used in the reports. An automated system for resolution of intelligent financial queries is therefore difficult to design. Several works have been proposed to overcome these problems using Information Extraction; however, they do not address the semantic interoperability of the reports across different institutions.

PROPOSED APPROACH

This paper intends to fill this research gap by presenting a comprehensive survey on the quality control of KGs. First, this paper defines six main evaluation dimensions of KG quality and investigates their correlations and differences. Second, quality control treatments during KG construction are introduced from the perspective of these dimensions of KG quality. Third, the quality enhancement of a constructed

 DOI : 10.36893.DRSR.2021.V08I14.142-146

KG is described from various dimensions. This paper ultimately aims to promote the research and applications of KGs.

LITERATURE SURVEY

J. Pujara, H. Miao, L. Getoor and W. W. Cohen, "Knowledge graph identification", *Proc. Int. Semantic Web Conf.*, pp. 542-557, 2013.

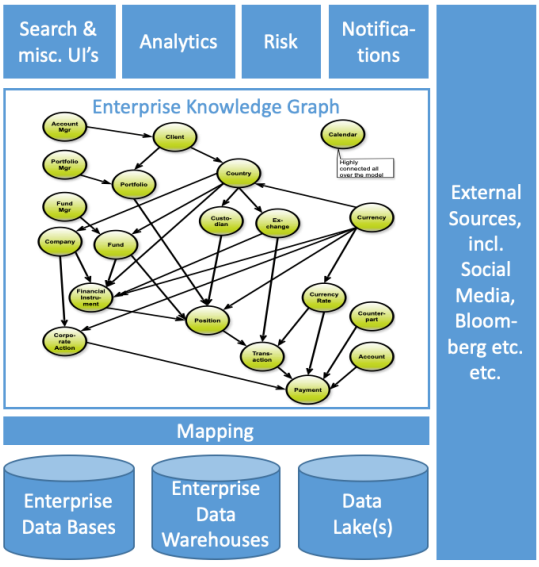
This work proposed an automated querying engine to answer the financial queries using Ontology based Information Extraction. For Semantic modeling of financial reports, a Financial Knowledge Graph, assisted by Financial Ontology, has been proposed. The nodes are populated with entities, while links are populated with relationships using Information Extraction applied on annual reports. Two benefits have been provided by this system to stakeholders through automation: decision making through queries and generation of custom financial stories. The work can further be extended to other domains including healthcare and academia where physical reports are used for communication.

G. Zhou, J. Su, J. Zhang and M. Zhang, "Exploring various knowledge in relation extraction", *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, pp. 427-434, 2005. A knowledge graph (KG), a special form of semantic network, integrates fragmentary data into a graph to support knowledge processing and reasoning. KG quality control is important to the utility of KGs. It is essential to investigate KG quality and the parameters influencing KG quality to better understand its quality control. Although many works have been conducted to evaluate the dimensions of KG quality, quality control of the construction process, and enhancement methods for quality, a comprehensive literature review has not been presented on this topic.

IMPLEMENTATION OF A KNOWLEDGE GRAPH

Let's look at the implementation of a knowledge graph from the ground up using the Google Drive API and Neo4j Python driver.This is only a piece of a larger end-to-end knowledge management solution, which would include*:

- Enterprise Knowledge Graph: the repository of enterprise knowledge (this is what we will cover in this post)
- GraphQL API: to sit on top of the knowledge graph
- React App: for non technical end users to explore the graph
- Recommendation Engines: to leverage graph patterns/algorithms and serve up relevant content to be displayed within the React app



 DOI : 10.36893.DRSR.2021.V08I14.142-146

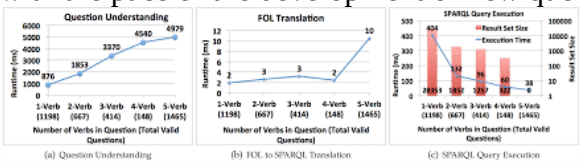
DATA MODELING AND PHYSICAL STORAGE

Modeling Data as RDF : One emerging data representation technique is the Resource Description Framework (RDF). RDF is a graph based data model for describing entities and their relationships on the Web. Although RDF is commonly described as a directed and labeled graph, many researchers prefer to think of it as a set of triples, each consisting of a subject, predicate and object in the form of . Triples are stored in a triple store and are queried with the SPARQL query language. Compared to both inverted indices and plain text files, triple stores and the SPARQL query language enable users to search for information with expressive queries in order to satisfy complex user needs. Although a model is required for representing data in triples (similar to relational databases), RDF enables the expression of rich semantics and supports knowledge inference. Another big advantage of adopting an RDF model is that it enables easier data deletion and update. Traditional data storage systems are "schema on write", i.e., the structure of the data (the data model) is decided at design time and any data that does not fit this structure is lost when ingesting the data. In contrast, "schema on read" systems attempt to capture everything and then apply computation horsepower to enforce a schema when the data is retrieved.

Data Storage: In our current implementation, we store the triples in two ways. We index the triples on their subject, predicate and object respectively with the Elastic search engine. We also build a full-text search index on objects that are literal values, where such literal values are tokenized and treated as terms in the index. This enables fast retrieval of the data with simple keyword queries. Additionally, we store all the triples in a triple store in order to support search with complex SPARQL queries. Currently, our TR knowledge graph manages about 5 billion triples; however, this only represents a small percentage of our data and the number of triples is expected to grow rapidly over time.

KNOWLEDGE GRAPH WITH NATURAL LANGUAGE

In previous sections, we have presented a Big Data framework and infrastructure for building an enterprise knowledge graph. However, given the built graph, one important question is how to enable end users to retrieve the data from this graph in an intuitive and convenient manner. Technical professionals, such as database experts and data scientists, may simply employ SPARQL queries to access this information. But non-technical information professionals, such as journalists, financial analysts and patent lawyers, who cannot be expected to learn such specialized query languages, still need a fast and effective means for accessing the data that is relevant to the task at hand. Keyword-based queries have been frequently adopted to allow non-technical users to access large-scale RDF data [10], and can be applied in a uniform fashion to information sources that may have wildly divergent logical and physical structure. But they do not always allow precise specification of the user's intent, so the returned result sets may be unmanageably large and of limited relevance. However, it would be really difficult for non-technical users to learn specialized query languages (e.g., SPARQL) and to keep up with the pace of the development of new query languages.



(a) Question Understanding (b) FOL to SPARQL Translation (c) SPARQL Query Execution

EVALUATION OF NATURAL LANGUAGE QUERYING

Dataset
Infrastructure.
Random Question Generation.
Time Complexity Analysis

 DOI : 10.36893.DRSR.2021.V08I14.142-146

Data Modeling. Our content covers diverse domains that range from finance to intellectual property & science and to legal and tax. It would be difficult for our engineers to precisely model such a complex space of domains and convert the ingested and integrated data into RDF triples. As we have initially attempted to adopt this data modeling approach, it has become clear that this is severely constrained by our engineering staffs' relative lack of expertise in the content. This is pushing us towards a need to invest in editorial focused self-service tooling to separate the software and content expertise. Rather than having engineers understand and perform the modeling, we collaborate closely with our editorial colleagues in order to model the data, apply the model to new contents, and embed the semantics into our data alongside its generation.

Distributed and Efficient RDF Data Processing. The relative scarcity of distributed tools for storing and querying RDF triples is another challenge. This reflects the inherent complexities of dealing with graph-based data at scale. Storing all triples in a single node would allow efficient graph operations while this approach may not scale well when we have an extremely large number of triples. Although we have been studying existing approaches for distributed RDF data processing and querying, these approaches often require a large and expensive infrastructure [50]. Our current solution is to use a highly scalable data warehouse (e.g., Apache Cassandra16 and Elasticsearch) for storing the RDF triples; in the meanwhile, slices of this graph can then be retrieved from the entire graph, put in specialized stores, and optimized to meet particular user needs.

Converging Triples from Multiple Sources. Another challenge is the lack of inherent capability within RDF for update and delete operations, particularly when multiple sources converge predicates under a single subject. In this scenario, one cannot simply delete all predicates and apply the new ones: triples from another source will be lost. While a simplistic solution might be to delete by predicate, this approach does not account for the same predicate coming from multiple sources. For example, if two sources state a "director-of" predicate for a given subject, an update from one source cannot delete the triple from the other source. Our solution is to use quads with the fourth element as a named graph allowing us to track the source of the triple and act upon subsets of the predicates under a subject.

Natural Language Interface. The first challenge is the tension between the desire to keep the grammar lean and the need for broad coverage. Our current grammar is highly lexicalized, i.e., all entities (lawyers, drugs, persons, etc.) are maintained as entries to the grammar. As the size of grammar expands, the complexity of troubleshooting issues that arise increases as well.

CONCLUSION

In this paper, The underlying big data based content pipelines (i.e., data acquisition, transformation and integration) continually update the graph with new facts and throttling is used to preserve the balance between ingest and online availability. The domain coverage of the contents within the graph continues to expand as we build out additional products dependent on the graph.

REFERENCES:

[1] J. Pujara, H. Miao, L. Getoor and W. W. Cohen, "Knowledge graph identification", *Proc. Int. Semantic Web Conf.*, pp. 542-557, 2013.

[2] C. Wang, M. Gao, X. He and R. Zhang, "Challenges in Chinese knowledge graph construction", *Proc. 31st IEEE Int. Conf. Data Eng. Workshops*, pp. 59-61, 2015.

[3] G. Zhou, J. Su, J. Zhang and M. Zhang, "Exploring various knowledge in relation extraction", *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, pp. 427-434, 2005.

[4] G. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring various knowledge in relation extraction," in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics ACL, 2005.

 DOI : 10.36893.DRSR.2021.V08I14.142-146

[5] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," IEEE Trans. Knowl. Data Eng., vol. 24, no. 9, pp. 1537–1555, 2012.
[6] S. Veeramachaneni and R. K. Kondadadi, "Surrogate learning: From feature independence to semi-supervised classification," in Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, 2009, pp. 10–18.
[7] C. Dozier, H. Molina-Salgado, M. Thomas, and S. Veeramachaneni, "Concord - a tool that automates the construction of record resolution systems," in Proceedings of Entity Workshop of LREC, 2010.
[8] A. Harth and S. Decker, "Optimized index structures for querying RDF from the web," in Third Latin American Web Congress, 2005, pp. 71–80.
[9] J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler, "Named graphs, provenance and trust," in Proceedings of the 14th Int'l conference on World Wide Web (WWW), 2005, pp. 613–622.
[10] L. Matteis, A. Hogan, and R. Navigli, "Keyword-based navigation and search over the linked data web," in Proceedings of the Workshop on Linked Data on the Web (LDOW), 2015.
[11] R. C. Cornea and N. B. Weininger, "Providing autocomplete suggestions," Feb. 4 2014, US Patent 8,645,825.
[12] C. Unger, L. Buhmann, J. Lehmann, A. N. Ngomo, D. Gerber, and ¨ P. Cimiano, "Template-based question answering over RDF data," in 21st World Wide Web Conference, 2012, pp. 639–648.
[13] J. Bovet and T. Parr, "Antlrworks: an ANTLR grammar development environment," Software: Practice and Experience, vol. 38, no. 12, pp. 1305–1332, 2008.
[14] B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, and R. Velkov, "OWLIM: A family of scalable semantic repositories," Semantic Web, vol. 2, no. 1, pp. 33–42, 2011. [15] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," CoRR, vol. abs/1601.00770, 2016.

 DOI : 10.36893.DRSR.2021.V08I14.142-146