CoreKG: a Knowledge Lake Service

Amin Beheshti
Macquarie University
Sydney, Australia
amin.beheshti@mq.edu.au

Boualem Benatallah
University of New South
Wales, Sydney, Australia
boualem@cse.unsw.edu.au

Reza Nouri
University of New South
Wales, Sydney, Australia
snouri@cse.unsw.edu.au

Alireza Tabebordbar
University of New South
Wales, Sydney, Australia
alirezat@cse.unsw.edu.au

ABSTRACT

With Data Science continuing to emerge as a powerful differentiator across industries, organisations are now focused on transforming their data into actionable insights. This task is challenging as in today's knowledge-, service-, and cloud-based economy, businesses accumulate massive amounts of raw data from a variety of sources. Data Lakes introduced as a storage repository to organize this raw data in its native format (supporting from relational to NoSQL DBs) until it is needed. The rationale behind a Data Lake is to store raw data and let the data analyst decide how to cook/curate them later. In this paper, we present the notion of Knowledge Lake, i.e. a contextualized Data Lake. The Knowledge Lake will provide the foundation for big data analytics by automatically curating the raw data in the Data Lake and to prepare them for deriving insights. We present CoreKG -an open source Data and Knowledge Lake service- which offers researchers and developers a single REST API to organize, curate, index and query their data and metadata in the Lake and over time. CoreKG manages multiple database technologies (from Relational to NoSQL) and offers a built-in design for data curation, security and provenance.

PVLDB Reference Format:
Amin Beheshti, Boualem Benatallah, Reza Nouri, and Alireza Tabebordbar. CoreKG: a Knowledge Lake Service. *PVLDB*, 11(12):1942-1945, 2018.
DOI: https://doi.org/10.14778/3229863.3236230

1. INTRODUCTION

With data science continuing to emerge as a powerful differentiator across industries, almost every organization is now focused on understanding their business and transform data into actionable insights. For example, governments derive insights from vastly growing private and open data for improving government services, such as to personalize the advertisements in elections, improve government services, predict intelligence activities and to improve national security and public health [8]. In this context, organizing vast amount of data gathered from various private/open data islands, i.e. *Data Lake*, will facilitate dealing with a collection of independently-managed datasets (from relational to NoSQL), diversity of formats and non-standard data models. The notion of a Data Lake [9] has been coined to address this challenge and to convey the concept of a centralized repository containing limitless amounts of raw (or minimally curated) data stored in various data islands.

The rationale behind a Data Lake is to store raw data and let the data analyst decide how to cook/curate them later. While Data Lakes, do a great job in organizing big data and providing answers on known questions, the main challenges are to understand the potentially interconnected data stored in various data islands and to prepare them for analytics. In our previous work [2] we have presented an open source Data Lake service which enables researchers/developers to organize, index and query their data in various data islands (ranging from Relational/NoSQL databases) in an easy way. This service provides interfaces to create a Data Lake, uses various technologies to persist information items (from structured entities to unstructured documents) in data islands, traces and persists information about data and enables the power of standard SQL and full-text search to query the data in the Lake.

In this paper, we extend our previous work and present the notion of *Knowledge Lake*, i.e. a contextualized Data Lake. The term *Knowledge* here refers to a set of facts, information, and insights extracted from the raw data using data curation techniques such as extraction, linking, summarization, annotation, enrichment, classification and more. In particular, a Knowledge Lake is a centralized repository containing virtually inexhaustible amounts of both data and *contextualized data* that is readily made available anytime to anyone authorized to perform analytical activities. The Knowledge Lake will provide the foundation for big data analytics by automatically curating the raw data in the Data Lake and to prepare them for deriving insights. We present CoreKG -an open source Data and Knowledge Lake service- which offers researchers and developers a single REST API to organize, curate, index and query their data and meta-

Proceedings of the VLDB Endowment, Vol. 11, No. 12
Copyright 2018 VLDB Endowment 2150-8097/18/8.
DOI: https://doi.org/10.14778/3229863.3236230

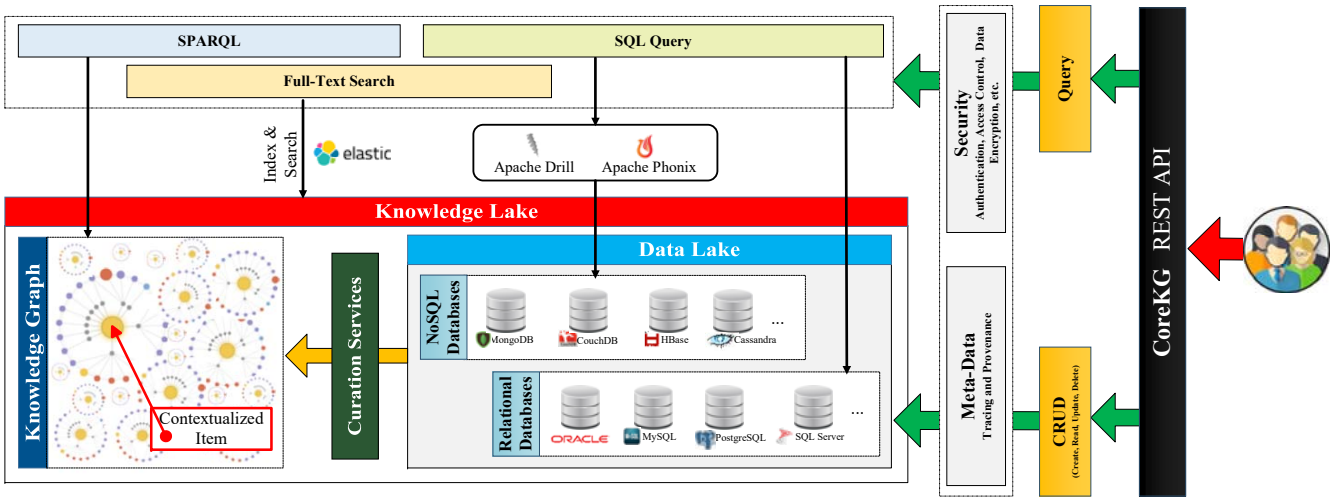

Figure 1: CoreKG Architecture.

data over time. CoreKG manages multiple database technologies and offers a built-in design to support:

- *Automatic Data Curation*: For the raw information items stored in the Lake, we provide services to automatically: (a) Extract features such as keyword, part of speech, and named entities such as Persons, Locations, Organizations, Companies, Products, and more; (b) Enrich the extracted features by providing synonyms and stems leveraging lexical knowledge bases for the English language such as WordNet; (c) Link the extracted enriched features to external knowledge bases (such as Google Knowledge Graph[1] and Wikidata[2]) as well as the contextualized data islands; and (d) Annotate the items in a data island by information about the similarity among the extracted information items, classifying and categorizing items into various types, forms or any other distinct class.

- *Security and Access Control*: to provide a database security protection mechanism including authentication, access control and data encryption for both data and the contextualized data.

- *Tracing and Provenance* [6]: to collect and aggregate tracing metadata (including descriptive, administrative and temporal metadata and build a provenance graph) for both data and the contextualized data.

The CoreKG API is available as an open source project on GitHub[3]. The rest of the paper is organized as follows. Section 2 presents an overview of the CoreKG. In Section 3 we describe our demonstration scenario.

2. SYSTEM OVERVIEW

CoreKG is an open source complete Data and Knowledge Lake Service. At the Data Lake layer (our previous work [2]), CoreKG powers multiple relational and NoSQL database-as-a-service for developing data-driven Web applications. This will enable analysts to create islands of data (relational and/or NoSQL datasets) within the Data Lake, CRUD (Create, Read, Update and Delete) entities in those datasets, and apply federated search on top of various islands of data. It also provides a built-in design to enable top database security threats (Authentication, Access Control and Data Encryption) along with Tracing and Provenance support.

On top of the Data Lake layer, in this paper, we present services to automatically curate the raw data in the Data Lake and prepare them for analytics. These services include extraction, summarization, enrichment, linking and classification. We present the notion of Knowledge Lake, a centralized repository containing virtually inexhaustible amounts of both raw data and *contextualized data*: the goal is to provide the foundation for big data analytics by automatically curating the raw data in data islands and to cook/curated data for deriving insights from the vastly growing amounts of local, external and open data.

We present CoreKG - an open source Knowledge Lake service - which offers researchers and developers a single REST API to organize, curate, index and query their data and metadata over time. Figure 1 illustrates the architecture and the main components of the CoreKG.

2.1 Data Lake Services

CoreKG offers a single REST API to construct a Data Lake which may contain a collection of data islands, i.e. datasets of type relational and/or NoSQL database. To create a relational database, a database connection configuration operation has been provided to enable access to many of relational databases such as MySQL, PostgreSQL and Oracle. CoreKG leverages appropriate NoSQL database such as MongoDB and HBase to organize key-value, document and graph stores requirements. CoreKG persists the entities (structured and unstructured) in a JSON format, an easy-to-parse structure, for its growing adoption in the Web data applications. Considering the self-describing nature of JSON documents, in CoreKG we extend JSON with the option of defining a schema for all or part of the data.

CoreKG has a built-in design to enable top database security protection mechanism (to supports Identification and Authentication requirements, System Privilege and Object Access Control and Data Encryption) along with a built-

[1] https://developers.google.com/knowledge-graph/
[2] https://www.wikidata.org/
[3] https://github.com/unsw-cse-soc/CoreKG

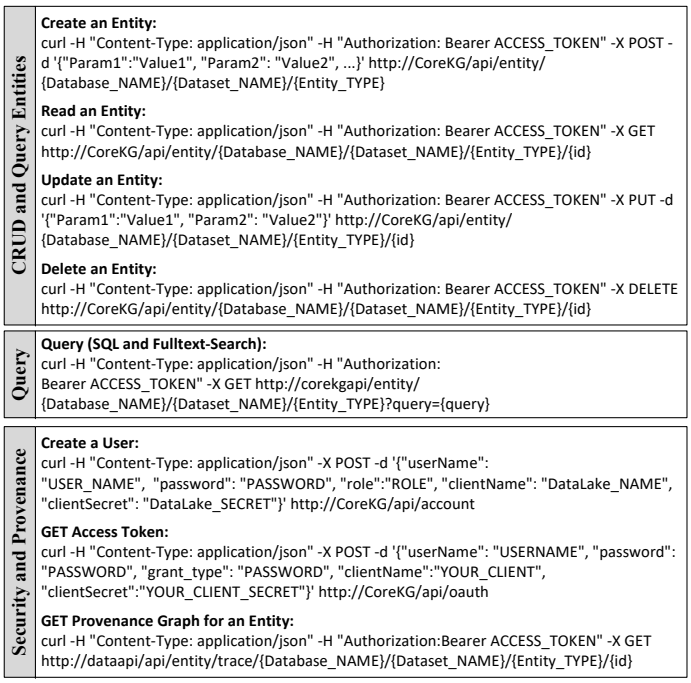

Figure 2: Data Lake Services.

in design to collect and aggregate tracing metadata including descriptive, administrative and temporal metadata. We use the tracing metadata to build a provenance graph [6]: a directed acyclic attributed graph where the nodes are users/roles and entities and the relationships among them represents the activities such as created, read, updated, deleted or queried. The relationships will be tagged with metadata such as timestamp and location. Figure 2 illustrates statements how to use Data Lake service to CRUD (Create, Read, Update and Delete) and Query (SQL Queries and Fulltext-Search Queries) entities in the Data Lake as well as creating users (Access Level) and retrieving the provenance graph (automatically generated using the built-in components in the system) of an entity. Details about Data Lake services can be found in our previous work [2].

2.2 Curation Services

To transform the Data Lake into a Knowledge Lake (i.e. a contextualized Data Lake), we propose a framework for data curation feature engineering: this refers to characterizing variables that grasp and encode information from raw or curated data, thereby enabling to derive meaningful inferences from data. An example of a feature is 'mentions of a person in data items like Tweets, news articles and social media posts'. We propose that features will be implemented and available as uniformly accessible data curation Micro-Services: functions or pipelines implementing features. In particular, we will support a concrete set of features [7] organized in the following categories:

Extraction. The majority of the digital information produced globally presented in the form of web pages, text documents, news articles, emails, and presentations expressed in natural language text [5]. Accordingly, it is important to extract features such as keyword, part of speech, and named entities such as Persons, Locations, Organizations, Companies, Products and more; from the raw data stored in the Lake. To achieve this goal, we provide services to automatically extract such features from raw data and turn them into curate data.

Enrichment. We provide services to enrich the extracted features by providing synonyms (a word or phrase that means exactly or nearly the same as another word or phrase in the same language) and stems (a form to which affixes can be attached) leveraging lexical knowledge bases for the English language such as WordNet. For example, considering the keyword 'health', using the Stem service, it is possible to identify derived forms such as healthy, healthier, healthiest, healthful, healthfully, healthfulness, etc. Also an example of synonyms is the words begin, start, and commence.

Linking. We provide services to link the extracted enriched features (e.g. named entities, keywords, synonyms, and stems) to external knowledge bases (such as Google Knowledge Graph and Wikidata) and the contextualized data islands. For example, if we extract a named entity of type person 'Greg Hunt' from the text of a Tweet and automatically link it to an existing entity in Wikidata 'Australian Minister for Health'[4], then we would be able to understand that the Tweet may be related to 'Australia' and 'Health'. This will provide more insight about the information items stored in the Lake.

Annotation. We provide services to Annotate the items in a data island using information about the similarity among the extracted information items, classifying and categorizing items into various types, forms or any other distinct class.

The following statements illustrate how to call the CoreKG service to get the contextualized entity in JSON format.

```
curl -H "Content-Type: application/json" -H "Authorization:
Bearer ACCESS_TOKEN" -X GET http://CoreKG/api/CuratedEntity/
{Database_NAME}/{Dataset_NAME}/{Entity_TYPE}/{id}
```

It should be highlighted that information items (stored in the Data Lake) could be structured or unstructured. Structured items are instances of typed entities and associated with a schema (consists of a set of attributes). Unstructured items, are also described by a set of attributes but may not conform to a schema. Details about curation services can be found in our previous work [7].

2.3 Index and Query

We provide a single service which enables the analyst querying the raw data (in the Data Lake) as well as the contextalized items in the Knowledge Lake using Full-text Search, SQL and SPARQL.

Full-text Search. CoreKG exposes the power of Elasticsearch without the operational burden of managing it by developers and supports querying both raw data and the meta-data generated during the curation process. When the user enables indexing while creating a dataset in the Lake, the entities as well as the features will be automatically indexed for powerful Lucene queries.

SQL. CoreKG enables the power of standard SQL with full ACID transaction capabilities for querying data held in relational/NoSQL databases. In particular, using a simple REST API, it is possible to send a SQL query to be applied to the datasets created in the Data Lake. To achieve this, in CoreKG, we leverage Apache Phoenix (phoenix.apache.org/)

[4] `https://en.wikipedia.org/wiki/Greg_Hunt`

to take the SQL query and compiles it into native NoSQL store APIs. Moreover, to support queries that need to join data from multiple datastores in the Data Lake, we leverage Apache Drill(drill.apache.org/).

SPARQL. As illustrated in Figure 2, we model the contextualized data and knowledge as a graph of typed nodes (e.g. raw data and extracted features) and edges (relationships among items such as 'keyword–extractedFrom–eMail', 'Person–extractedFrom–Tweet' or 'item–similarTo–item'). In order to query this graph a graph query language is needed. Among languages for querying graphs, SPARQL[5] is an official W3C standard and based on a powerful graph matching mechanism. For this type of query, we leverage our previous work [3, 4], a SPARQL query engine for analyzing large graphs, to organize the data and extracted-enriched-linked features.

The following statement illustrate how to call the CoreKG service to apply a query (Full-text search, SQL or SPARQL) to the Knowledge Lake:

```
curl -H "Content-Type: application/json" -H "Authorization:
Bearer ACCESS_TOKEN" -X GET http://corekgapi/entity/
{Database_NAME}/{Dataset_NAME}/{Entity_TYPE}?query={query}
```

For example, to find the Tweets (stored in 'dsTweets' dataset persisted in the MongoDB database) that contains the keyword 'VLDB' the following query can be used.

```
curl -H "Content-Type: application/json" -H "Authorization:
Bearer ACCESS_TOKEN" -X GET http://corekgapi/entity/
dsTweets/MongoDB/Tweet? query="{"match": {"text":"VLDB"}}"
```

3. DEMONSTRATION SCENARIO

The objective of this scenario is to enable police investigators to understand and organize the large amount of open and private data used and generated during the investigation process, as well as to automate the data creation and value generation from this raw data. A criminal investigation refers to the process of collecting information (or evidence) about a crime in order to: determine if a crime has been committed, identify/apprehend the perpetrator and provide evidence to support a conviction in court. In this context, an investigator may deal with large amount of raw data, from structured to unstructured, including the data in social networks, emails, telephone intercepts and more. A simple scenario would be to assist an investigator in automatically transforming an audio file generated in a telephone intercept task into text, extract name of a person or a phone number, find similar people in the police historical data or archived news, identify and link these items to entity profiles and more. Accordingly, a Knowledge Lake can dramatically improve and automate many time consuming and manual tasks. The demonstration scenario consists of three parts. We illustrate how an investigation team can use CoreKG services to: (i) construct a Data Lake and store a set of Tweets, emails and telephone intercepts (audio files transformed into text files) in different islands of data; (ii) view the automatically contextalized Tweets, emails and phone conversations; and (iii) apply full-text search, SQL and SPARQL queries to the data and meta-data stored in the Knowledge Lake.

Experiment. The performance of Data Lake and the Curation Services has been illustrated in [2, 7]. For the demonstration, we will run the experiment on Amazon EC2 platform; We will demonstrate the scalability experiment on a single, four and eight machines; and on 15 million Tweets from Twitter and the emails in the Enron Email Dataset (`cs.cmu.edu/~enron/`).

[5] `https://www.w3.org/TR/rdf-sparql-query/`

4. RELATED WORK AND CONCLUSION

The two closest systems to our work include AsterixDB [1] (asterixdb.apache.org/) and Orchestrate (orchestrate.io/). The added value of CoreKG compare to these systems include: managing multiple database technologies (From Relational to NoSQL) and providing a built-in design for data curation, security and provenance. As an ongoing work, we are working on novel techniques to summarize the contextualized data in the Knowledge Lake and automatically generating narratives and telling stories with data.

Acknowledgments. We Acknowledge the Data to Decisions CRC and the Cooperative Research Centres Programme for funding part of this research.

5. REFERENCES

[1] S. Alsubaiee, A. Behm, V. R. Borkar, Z. Heilbron, Y. Kim, M. J. Carey, M. Dreseler, and C. Li. Storage management in AsterixDB. *PVLDB*, 7(10):841–852, 2014.

[2] A. Beheshti, B. Benatallah, R. Nouri, V. M. Chhieng, H. Xiong, and X. Zhao. Coredb: a data lake service. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore*, pages 2451–2454, 2017.

[3] S. Beheshti, B. Benatallah, and H. R. Motahari-Nezhad. Galaxy: A platform for explorative analysis of open data sources. In *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France*, pages 640–643, 2016.

[4] S. Beheshti, B. Benatallah, and H. R. Motahari-Nezhad. Scalable graph-based OLAP analytics over process execution data. *Distributed and Parallel Databases*, 34(3):379–423, 2016.

[5] S. Beheshti, B. Benatallah, S. Venugopal, S. H. Ryu, H. R. Motahari-Nezhad, and W. Wang. A systematic review and comparative analysis of cross-document coreference resolution methods and tools. *Computing*, 99(4):313–349, 2017.

[6] S. Beheshti, H. R. M. Nezhad, and B. Benatallah. Temporal provenance model (TPM): model and query language. *CoRR*, abs/1211.5009, 2012.

[7] S. Beheshti, A. Tabebordbar, B. Benatallah, and R. Nouri. On automating basic data curation tasks. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 165–169, 2017.

[8] O. Tene and J. Polonetsky. Big data for all: Privacy and user control in the age of analytics. *Nw. J. Tech. & Intell. Prop.*, 11:xxvii, 2012.

[9] I. G. Terrizzano, P. M. Schwarz, M. Roth, and J. E. Colino. Data wrangling: The challenging yourney from the wild to the lake. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA*, 2015.