Coronis: Towards Integrated and Open COVID-19 Data

Georgios M. Santipantakis
Dept. of Digital Systems
University of Piraeus
Piraeus, Greece
gsant@unipi.gr

George A. Vouros
Dept. of Digital Systems
University of Piraeus
Piraeus, Greece
georgev@unipi.gr

Christos Doulkeridis
Dept. of Digital Systems
University of Piraeus
Piraeus, Greece
cdoulk@unipi.gr

ABSTRACT

Motivated by the global unrest related to the COVID-19 pandemic, this demo paper presents a system for acquisition of COVID-related data from different, public sources, and interlinking under a common semantic data model at a fine level of granularity. The integrated data set contains data from several European countries, which come in different schemata, formats, granularity, and data integration acts as a facilitator towards querying data from different sources, joint data analysis, and identifying correlations at varying geographical level. Moreover, our work shows how such an integrated data set can be exploited to answer complex questions for the pandemic, also in combination with other data sets via federated queries.

1 INTRODUCTION

The COVID-19 virus outbreak presents major problems worldwide, as it affects every country in the world, both economically and socially. Governments publish COVID-related data daily, and it is commonly accepted that analyzing this data may unveil hidden patterns and aid in developing a better understanding of the pandemic. However, published data comes in different schemata, formats, granularity, which prevents data analysts from applying advanced spatio-temporal analysis methods for monitoring the evolution in space-time, due to the well-known problem of big data integration from disparate sources [7]. This motivates our work for building an extensible system based on linked data principles that allows integration of data that combines reports regarding COVID-19 cases from various countries with other public data sources.

To the best of our knowledge, the public data sets related to COVID-19 that are currently available, either report the total number of cases per country or per administrative region within a specific country. Typical examples of the first category are the World Health Organization (WHO) Coronavirus Disease Dashboard [6], the WorldOmeter [1] and the Johns Hopkins University dashboard [2], built mainly for monitoring the situation at a coarse level of detail, rather than for supporting any kind of analysis, as epidemiologists do with elaborated models. On the other hand, several countries report details about the confirmed cases in their administration regions [3]. Nevertheless, it is hard to extract the time series of reported cases for each country from the unstructured text. Also, summaries of reports provided per region through online repositories by individual countries, such as [4] and [5], do not share a common schema and use different data formats (JSON, CSV or ESRI shapefiles), which hinders joint data exploration and analysis. However, data integration needs to solve several problems at a technical level, such as different languages, text encodings and region identification systems used. For example, Germany uses "landkreis" codes, whereas Austria uses "Gemeindekennziffer" (GKZ) codes. As a result, it is necessary to convert this spatial data to a level of granularity that allows correlation analysis.

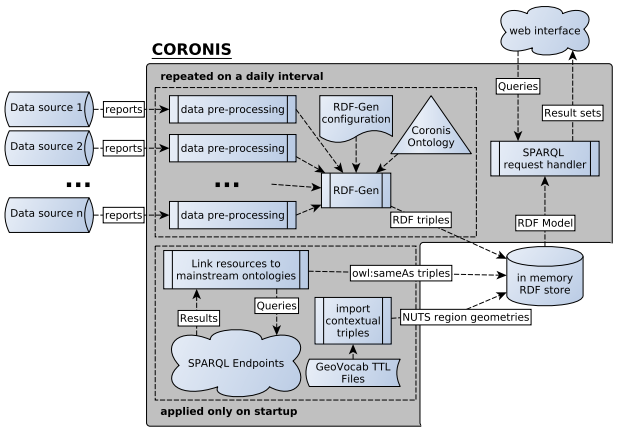

Figure 1: The overall workflow for retrieving, transforming updating and publishing data as RDF triples.

This work contributes a system, called *Coronis*[1], for gathering and linking COVID-related data from different sources under a common ontology, at a fine level of granularity and as five-star Open Data[8], enabling the correlated analysis of such data via queries that can extract useful knowledge and insights. Currently, our prototype automatically retrieves data from daily reports regarding COVID cases from 7 European countries, populating a single ontology, at a specific level of spatial and temporal granularity. Moreover, the administrative regions reported in the data set, are related to data regarding population density per region and per various age groups[2], as well as to external Open Data portals. Data exploration is enabled by a SPARQL[11] endpoint that supports federated queries. The result set is visualized using a variety of options including tables, charts and a map-based interface.

2 SYSTEM ARCHITECTURE

Figure 1 illustrates the overall system architecture as well as the workflow for data integration. Our system for COVID-19 data acquisition, integration and querying comprises the following main components:

- *Data connectors*: enabling data acquisition from different data sources.
- *Data transformation*: converts incoming data into a common semantic representation (RDF triples) according to a given ontology.

[1] In greek mythology, Coronis is a Thessalian princess and a lover of Apollo, also the mother of Asclepius, the Greek god of medicine.
[2] Data retrieved from: https://ec.europa.eu/eurostat/

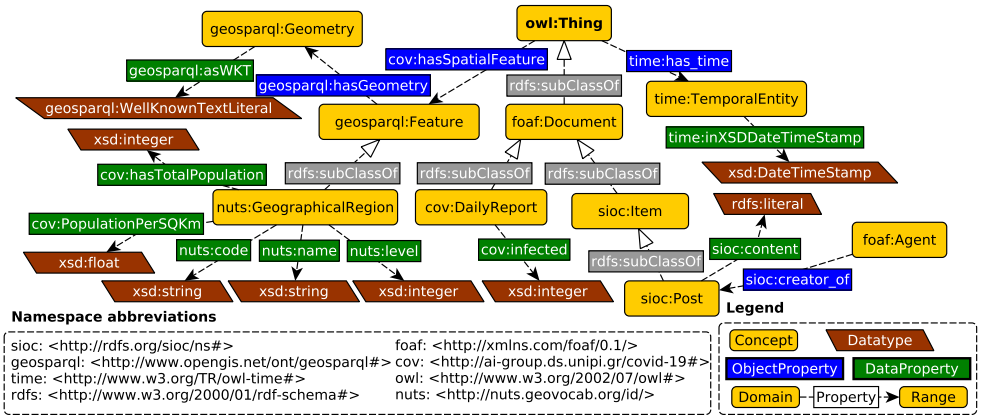

Figure 2: Core concepts (rectangles) and properties (labeled edges) of the ontology.

- *Data storage*: using an in-memory RDF store for efficient SPARQL query processing.
- *Query handler*: a SPARQL request handler that validates SPARQL queries and directs them to the RDF store.
- *User interface*: the web-based, graphical user interface that renders query results in different formats.

2.1 Data Connectors

We separate data acquisition from the remaining components, by implementing a set of data connectors, one for each data source. A data connector is responsible for establishing the connection to a remote data source, parse data, and perform data preparation tasks, such as conversion of spatial and textual encoding, build population groups, etc. The separation of data acquisition and parsing from data transformation to RDF, makes the system extensible, flexible and robust to data source modifications, or even failures of individual connectors.

The data sources accessed daily for confirmed cases per region include: Austria[3], Belgium[4], France[5], Germany[6], Greece[7], Italy[8], and Sweden[9]. Regions referenced in the data of these sources are converted to the corresponding level of *Nomenclature of Territorial Units for Statistics* (NUTS) regions. This conversion allows the correlation of regions of different countries, and also enables data integration with population data per region and age group, provided by Eurostat[10].

2.2 Data Transformation to RDF

The transformation of data into RDF triples is performed using RDF-Gen [10], our tool for efficient and flexible data transformation to RDF. RDF-Gen transforms input data using a *template* of triples that allows the use of variables or predefined functions on any of the constituent parts (subject, predicate, object) of a triple. The connector to a COVID-19 data source is initialized with contextual data related to the source, such as Administrative Regions and population (total and groups per age). The data are then automatically converted to RDF triples by RDF-Gen, and the whole process is repeated on a daily interval.

The transformation process also computes `owl:sameAs` triples between resources referring to regions in our data set and resources referring to the same regions on Open Data, such as the EU Open Data Portal, wikiData[11] and FactForge[12].

2.3 Triple Store and Query Handler

The generated RDF triples are preserved in an in-memory RDF store (Jena 3.14), which enables efficient evaluation of federated SPARQL queries over the integrated data set. In federated queries, a portion of the query is directed to a particular remote SPARQL endpoint and results returned to the federated query processor are combined with results from the rest of the query. The RDF store is initialized with static data (GeoVocab TTL files) that describe the geometries of NUTS regions and their topological relations.

The Query Handler implements SPARQL 1.1 protocol and enables our endpoint to participate in federated queries. Queries to the RDF store are supported by means of YASGUI [9], which features a user-friendly SPARQL query editor and allows rendering the result set in a wide range of formats, varying from plain CSV tables to 2D-3D maps, enriched with HTML formatted pop-ups.

2.4 The Coronis Ontology

To support the process of data integration, we build an ontology that describes the domain. Figure 2 illustrates the core concepts and properties of the ontology, where the rounded rectangles represent concepts, while edges and skewed parallelograms illustrate properties and datatypes respectively. Our ontology imports at the conceptual level:

SIOC (Semantically-Interlinked Online Communities) Core Ontology[13], OGC GeoSPARQL standard[14], OWL-Time ontology[15], and RAMON geographic ontology[16]. The integration of COVID-19 reports from different countries with EU NUTS/RAMON introduces the spatial dimension to the data, and allows the detection of topological relations between regions.

We use `cov:`, `geosparql:`, `nuts:`, as prefix abbreviations for the namespaces of our ontology, GeoSPARQL and EU NUTS RDF ontologies, respectively. The concept `cov:DailyReport`

[3]https://www.drawingdata.net/covmap/
[4]https://epistat.wiv-isp.be/
[5]https://www.data.gouv.fr/fr/datasets/donnees-hospitalieres-covid-19/
[6]https://corona.rki.de
[7]https://github.com/iMEdD-Lab/open-data/tree/master/COVID-19
[8]https://github.com/pcm-dpc/COVID-19
[9]https://visalist.io/emergency/coronavirus/sweden-country
[10]http://ec/europa.eu

[11]https://www.wikidata.org/
[12]http://factforge.net/
[13]https://www.w3.org/Submission/sioc-spec/
[14]http://www.opengis.net/ont/geosparql
[15]https://www.w3.org/TR/owl-time/
[16]https://ec.europa.eu/eurostat/ramon/ontologies/geographic.rdf

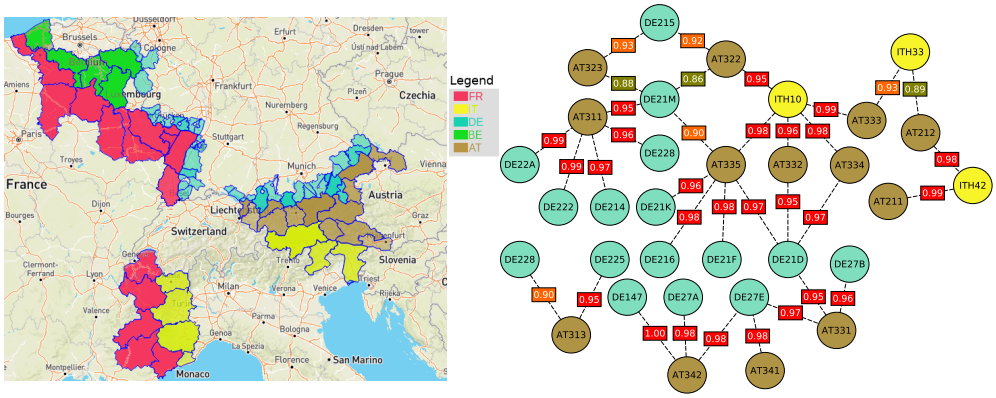

Figure 3: Adjacent regions (left) and correlations between German, Austrian and Italian regions (right).

represents the set of reports of confirmed cases. A daily report has both spatial and temporal constituents: it is associated to a resource representing a region in the EU NUTS RDF vocabulary by the property `cov:hasSpatialFeature`, and it is also related to exactly one temporal resource via the property `time:has_time`. We also define `nuts:GeographicalRegion` as a subclass of `geosparql:Feature`.

In addition, we specify a set of data properties related to the population profile of any region. Specifically, the total population, the population density, and the population per age group, with domain `nuts:GeographicalRegion`. Finally, the properties `cov:infected` and `cov:deceased` specify the number of infected and deceased cases for a region, respectively.

2.5 Queries

The Coronis ontology enables the retrieval of interlinked data with simple SPARQL queries. For example, the reported cases for a specific region (e.g., Ravensburg), sorted by date, can be obtained with the query:

```
PREFIX : <http://ai-group.ds.unipi.gr/covid-19#>
PREFIX nuts: <http://nuts.geovocab.org/id/>
PREFIX time: <http://www.w3.org/2006/time#>
SELECT ?report ?date ?population ?infected ?deceased
WHERE {
   ?report :hasSpatialFeature ?region ;
      :infected ?infected ; :deceased ?deceased ;
      time:has_time/time:inXSDDateTimeStamp ?date .
   ?region :totalPopulation ?population ;
      nuts:name "Ravensburg" .
}order by ?date
```

Figure 3 depicts an example of exploiting interlinked data from European regions, in order to investigate whether the number of infections reported in adjacent regions (of different countries) show a linear correlation. Figure 3 (left) illustrates the 67 pairs of regions returned for this query. Interestingly, the results show high correlation (0.86–1.00) between Austrian and German regions, depicted in Figure 3 (right). Lower correlation (0.25–0.46) is observed between French and Belgian regions, while negative values (-0.04 – -0.48) between French and Italian regions (probably a result of measures in Italy, when the reported number of infections dramatically increased).

To identify the adjacent regions $g1, g2$ in queries, we use the spatial predicate $touches(g1, g2)$. The Pearson R coefficient in Figure 3 (right) is computed from the results of the query:

```
PREFIX : <http://ai-group.ds.unipi.gr/covid-19#>
PREFIX nuts: <http://nuts.geovocab.org/id/>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX f: <java:SPARQL_functions.>
SELECT ?r1 ?r2 ?inf1 ?inf2 ?date WHERE {
   ?r1 nuts:name ?name ;
      geosparql:hasGeometry/geosparql:asWKT ?wkt ;
      nuts:level ?l1 .
   ?r2 nuts:name ?name2 ;
      geosparql:hasGeometry/geosparql:asWKT ?wkt2 ;
      nuts:level ?l2 .
   ?c1 :hasPart ?r1 ; nuts:level "0" .
   ?c2 :hasPart ?r2 ; nuts:level "0" .
   ?rp1 :hasSpatialFeature ?r1 ; :infected ?inf1 ;
      time:has_time/time:inXSDDateTimeStamp ?date.
   ?rp2 :hasSpatialFeature ?r2 ; :infected ?inf2 ;
      time:has_time/time:inXSDDateTimeStamp ?date.
   FILTER(f:touches(?wkt,?wkt2)&&(?c1!=?c2) &&
      ((?l1="2")||(?l1="3"))&&((?l2="2")||(?l2="3")))
} ORDER BY ?date
```

3 SYSTEM DEMONSTRATION

Coronis provides a wide range of options for query building and rendering the result set, by supporting federated queries and a user-friendly web interface based on YASGUI. In this section, we provide the demonstration scenario briefly[17]. The system prototype uses a SPARQL editor to query the integrated data set.

Illustrate results in tabular format: In this option, the result set is presented as a table where each column corresponds to the projected variables, and each row corresponds to the combination of values that match the query pattern. The web interface enables sorting the results by values of specific columns or filtering the results by value. A table reporting the total number of confirmed cases for a specific date is shown in Figure 4.

Illustrate results in a grid: The result set of queries are rendered in a grid, where the cells are HTML formatted blocks dynamically constructed from the result set. For example, Figure 5 illustrates the result set of a federated query, combining number of hospitals (from wikidata) and confirmed cases (from the locally stored RDF triples) per region.

Illustrate results in a chart: In this option, a result sets that contain numerical values is presented in one of various chart types. The user can select the type of chart (and customize) by clicking on "configure". The chart can be also downloaded as an SVG file for offline use. Figure 6 depicts the number confirmed cases in Bayern per day.

[17]The queries presented in this paper and additional examples are publicly available at the endpoint's URL address: http://83.212.169.101/datasets/yasgui.html

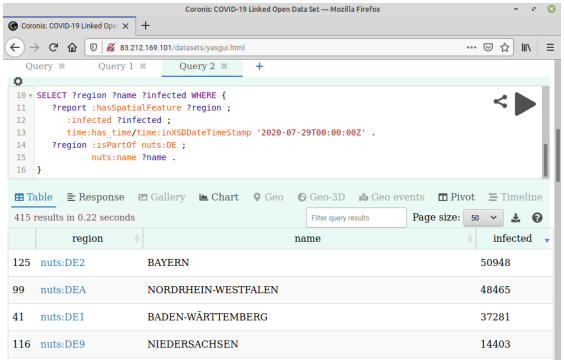

Figure 4: Result set rendered as a table.

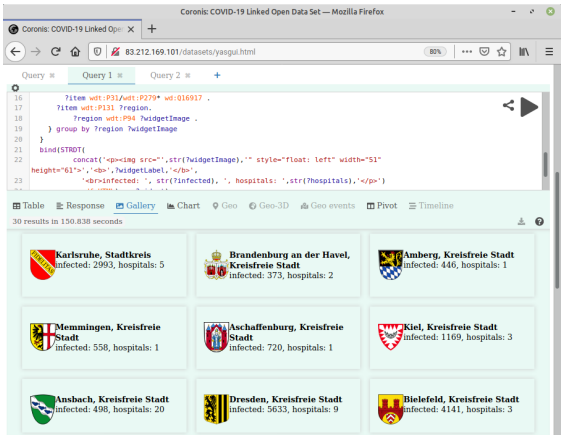

Figure 5: Result set rendered as a gallery.

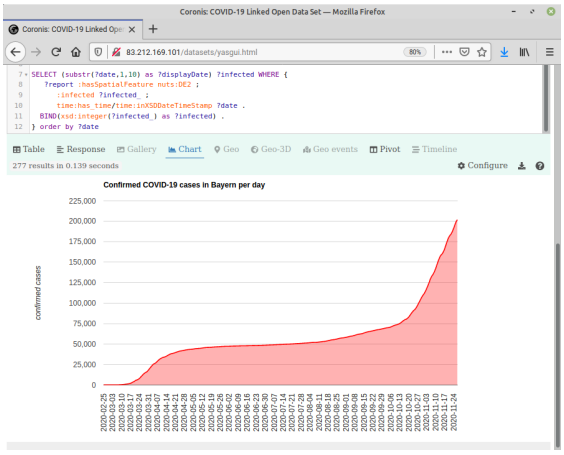

Figure 6: Example of a result set as a chart.

Illustrate results in a map: This option can be applied on results sets with spatial dimension. It renders each spatial object in the result set as a point or polygon on a 2D or a 3D map, on top of OpenStreetMap layer. The 2D map of Figure 3 and 3D map of Figure 7 are examples of this option.

4 CONCLUSIONS

This work presents Coronis, a prototype for data collection and integration of an Open Data set about COVID-19 confirmed cases

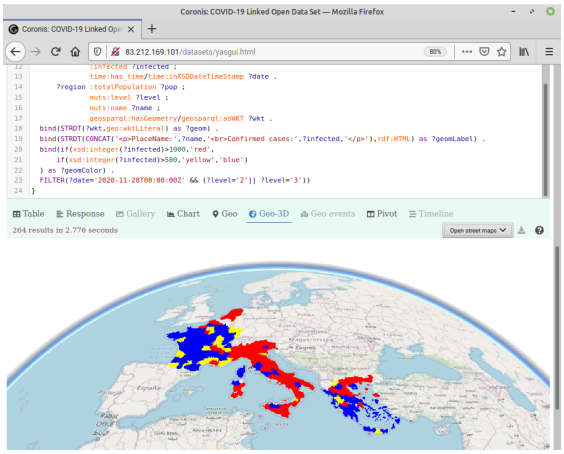

Figure 7: Result set rendered as a 3D map.

that enables cross-country, spatio-temporal analysis at different levels of granularity. Our work facilitates querying and analyzing data from different data sources, which would otherwise be a tedious and time-consuming task. The integrated data set is transformed into RDF triples to populate an ontology built on top of well-known ontologies, and resources are linked to external Open Data repositories. In turn, this enables the formulation of complex queries over interlinked COVID data with external sources, thus offering the opportunity for advanced data analysis. In our future work, we plan to expand our data set with more countries and link the data with more portals that provide information about social events, news feeds and human activities that possibly affect the spreading of the virus.

ACKNOWLEDGMENTS

The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: HFRI-FM17-81).

REFERENCES

[1] 2020 (accessed July 20, 2020). *COVID-19 Coronavirus Pandemic.* https://www.worldometers.info/coronavirus/

[2] 2020 (accessed July 20, 2020). *COVID-19 Dashboard by the Center for Systems Science and Engineering at Johns Hopkins University.* https://coronavirus.jhu.edu/map.html

[3] 2020 (accessed July 20, 2020). *COVID-19 pandemic.* https://en.wikipedia.org/wiki/COVID-19_pandemic

[4] 2020 (accessed July 20, 2020). *iMedd webpage.* https://www.imedd.org/new-covid-19-i-watch-the-spread-of-the-disease-in-greece-and-around-the-world/

[5] 2020 (accessed July 20, 2020). *Presidenza del Consiglio dei Ministri - Dipartimento della Protezione Civile.* https://github.com/pcm-dpc/COVID-19

[6] 2020 (accessed July 20, 2020). *World Health Organization Coronavirus Disease Dashboard.* https://covid19.who.int/

[7] Xin Luna Dong and Divesh Srivastava. 2013. Big Data Integration. *Proc. VLDB Endow.* 6, 11 (2013), 1188–1189.

[8] Krzysztof Janowicz, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman. 2014. Five Stars of Linked Data Vocabulary Use. *Semant. Web* 5, 3 (July 2014), 173–176.

[9] Laurens Rietveld and Rinke Hoekstra. 2013. YASGUI: Not Just Another SPARQL Client. In *The Semantic Web: ESWC 2013 Satellite Events.* Springer Berlin Heidelberg, Berlin, Heidelberg, 78–86.

[10] Georgios M. Santipantakis, Konstantinos I. Kotis, George A. Vouros, and Christos Doulkeridis. 2018. RDF-Gen: Generating RDF from Streaming and Archival Data. In *WIMS.* ACM, 28:1–28:10.

[11] Emanuele Della Valle and Stefano Ceri. 2011. *Querying the Semantic Web: SPARQL.* Springer Berlin Heidelberg, Berlin, Heidelberg, 299–363. https://doi.org/10.1007/978-3-540-92913-0_8