Keyword Search over COVID-19 Data

Yenier T. Izquierdo[1,2], Grettel M. García[2], Melissa Lemos[1,2], Alexandre Novello[1], Bruno Novelli[2], Cleber Damasceno[2], Luiz André P.P. Leme[3], Marco A. Casanova[1,2]

[1]Department of Informatics, PUC-Rio, Rio de Janeiro, RJ – Brazil

[2]Instituto TecGraf, PUC-Rio, Rio de Janeiro, RJ – Brazil
[3]Institute of Computing, UFF, Niterói, RJ, Brazil

{ytorres,ggarcia,melissa}@tecgraf.puc-rio.br, anovello@inf.puc-rio.br, {bnovelli,cleberoli}@tecgraf.puc-rio.br, lapaesleme@ic.uff.br, casanova@inf.puc-rio.br

Abstract. *Keyword search is typically associated with information retrieval systems. However, recently, keyword search has been expanded to relational databases and RDF datasets, as an attractive alternative to traditional database access. With this motivation, this paper first introduces a platform for data and knowledge retrieval, called DANKE, concentrating on the keyword search component. It then describes an application that uses DANKE to implement keyword search over two COVID-19 data scenarios.*

1. Introduction

A large variety of COVID-19 data became available during the first semester of 2020. The COVID-19 Data Repository maintained by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University is perhaps the worldwide reference repository of COVID-19 data. In Brazil, the Ministry of Health and other private institutions maintain detailed data, including demographic, clinical and lab exams, and hospitalization information.

However, some of the datasets are sufficiently large to be cumbersome to process with the usual desktop spreadsheet tools. A perhaps more robust approach is to: (1) store the data in a standard DBMS; (2) define a query that retrieves the data the user is interested in; (3) export the query results to a data analysis tool. In particular, the second step requires the user to know how the database is organized and to master SQL. This paper addresses the query definition problem by proposing a different alternative based on keyword search over databases. Keyword search is typically associated with information retrieval systems, where the user specifies a few terms, called *keywords*, and the system must retrieve the documents, such as Web pages, that best match the list of keywords. A *keyword query* is simply a list of keywords. Recently, keyword search has been expanded to relational databases and RDF datasets, as an attractive alternative to traditional database access.

The paper first introduces a platform for data and knowledge retrieval, called DANKE, concentrating on the keyword search component. The paper then describes an application that uses DANKE to implement keyword queries over two COVID-19 data scenarios.

2. Brief Review of Related Work

This section very briefly reviews work related to database keyword search and lists some COVID-19 data collection efforts.

A survey of keyword query processing tools over relational databases and RDF datasets can be found in (Bast et al., 2016). Tools, such as (Bergamaschi et al., 2016; Oliveira et al., 2015) explore the foreign/primary keys declared in the relational schema to compile a keyword query into an SQL query with a minimal set of join clauses. RDF keyword query processing tools can be schema-based, when they exploit the RDF schema to compile a keyword query into a SPARQL query, or graph-based, when they directly explore the RDF dataset or summaries thereof. QUIOW (García et al., 2017)(Izquierdo et al., 2018) is an example of a schema-based tool. Examples of graph-based tools can be found in (Han et al., 2017; Tran et al., 2009).

To name a few repositories, the COVID-19 Data Repository maintained by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University[1] is perhaps the worldwide reference repository of COVID-19 data. The Academic Data Science Alliance also collects data and data science resources related to the COVID-19 pandemic[2] and the Copyright Clearance Center maintains a list of links to COVID-19 data resources[3]. The Google Cloud services created the COVID-19 Public Datasets program[4] to make COVID-19 data more accessible to researchers. This effort includes the BigQuery sandbox, which allows free queries over certain COVID-related datasets. In Brazil, the Ministry of Health maintains a specific dataset, which we will refer to as NSG (*Notificações de Síndrome Gripal*), with notifications of suspected COVID-19 cases[5], and a more comprehensive database, called SRAG 2020[6], with Severe Acute Respiratory Syndrome (SARS) and COVID-19 data. The COVID-19 Data Sharing / BR initiative published open data with demographic, clinical and laboratory data about patients tested for COVID-19 in the State of São Paulo (Mello et al., 2020).

3. The DANKE Keyword Search Component

The DANKE keyword search component uses the technology described in (García et al., 2017; Izquierdo et al., 2018; García, 2020), operates over both relational databases and RDF datasets, and explores the database schema to compile a keyword query into an SQL or SPARQL query that returns data that best match the keywords. It is capable of synthesizing database queries with projections and selections, as well as joins involving several tables, without user intervention. DANKE has other components that automatically summarize (long) answers, allow the user to add more attributes to an answer, navigate through the database starting from the answer, transform tabular data to certain graphical forms, and export an answer as a CSV file. This section outlines how the keyword search component operates, assuming that the underlying database is relational; the processing of RDF datasets is entirely similar.

[1] https://github.com/CSSEGISandData/COVID-19
[2] https://academicdatascience.org/covid
[3] http://www.copyright.com/coronavirus-covid-19-data/
[4] https://console.cloud.google.com/marketplace/product/bigquery-public-datasets/covid19-public-data-program?pli=1
[5] https://opendatasus.saude.gov.br/dataset/casos-nacionais
[6] https://opendatasus.saude.gov.br/dataset/bd-srag-2020

A *keyword query* is just a list of terms, or *keywords*, that the user wants to search the database for, and may include reserved terms, such as "<". Section 4 provides examples of keyword queries. An answer to a keyword query is formatted as a table whose columns (or column names) contain the keyword matches. The answer may be the result of joining several database tables, that is, an answer to a keyword query does not need to be constructed out of a single table.

To use a relational database D with schema S, the first step is to register D with DANKE. This step is executed only once. The registration process includes, among other tasks, indicating which columns will have their values indexed, and adding descriptions to the relation schemes and attributes. Such column values and descriptions provide the terms against which DANKE will match the keywords. The database schema S is compiled into an abstract schema, which is independent of the model (relational or RDF) of the underlying database.

The DANKE keyword search component features an algorithm (García et al., 2017) that accepts a keyword query K over a relational database D, together with its relational schema S, and: (1) finds matches with the keywords in K; (2) creates an abstract query by exploring the keyword matches found and the schema S; (3) compiles the abstract query into an SQL query, which is then executed.

In more detail, with the help of special indices, the algorithm computes a set of relation schemes and attributes in S whose metadata (names and descriptions) match keywords in K and a set of attribute/value pairs whose values match keywords in K.

The algorithm synthesizes an abstract query that captures the keyword query, using information from the abstract schema and the matching process output. To synthesize an abstract query, the algorithm implements two heuristics, called the scoring and the minimization heuristics. Briefly, the scoring heuristic: (1) considers how good a match is; (2) assigns a higher score to metadata matches, on the grounds that, if the user specifies a keyword that matches both a relation scheme name (or description) and an attribute value of a tuple, then the user is probably more interested in the scheme than the specific instance; (3) assigns a higher score to relation schemes, called *nucleuses*, whose attributes cover a larger number of keywords. The minimization heuristic connects the nucleuses using a small number of equijoins. This is equivalent to generating a Steiner tree ST of the abstract schema graph that covers the nucleuses that were prioritized.

The algorithm creates the WHERE clause of the abstract query using: the filters included in the keyword query to generate selection clauses; the edges of ST to generate join clauses; the nucleuses to generate additional selection clauses.

The final stage of the algorithm is to compile the abstract query into a concrete SQL query for the underlying relational DBMS. The SQL query is then executed to generate an answer to the keyword query, which is passed to the user.

4. *CovidKeyS* – A Keyword Search Application over COVID-19 Data

CovidKeyS is a Web Application that uses the DANKE services to query COVID-19 data and is accessible at the endpoints indicated in what follows. To illustrate *CovidKeyS*, this section describes two COVID-19 data scenarios, with a few sample keyword queries.

Scenario 1: The Brazilian NSG (Notificações de Síndrome Gripal) dataset. As mentioned in Section 2, NSG stores notifications of suspected COVID-19 cases. Such data is sufficiently rich to allow relating demographic, clinical, and laboratory data. NSG data was organized in DANKE as a relational database with three tables, `Paciente`, `Teste`, and `Desfecho`, respectively storing patient data, COVID-19 test data, and how the case evolved. Two examples of keyword queries are:

- NL Query: "Quais foram os casos em Saquarema de gestantes, com sintoma de febre, e idade menor do que 40 anos?" ("Which were the cases in Saquarema involving pregnant women, with fever symptoms, and age less than 40 years?")
  Keyword query: *Saquarema gestante febre idade < 40*

- NL Query: "Quais enfermeiros, e em que municípios, fizeram teste rápido e tiveram resultado positivo?" ("Which nurses in which counties had a rapid test with positive result?")
  Keyword query: *enfermeiro município "teste rápido" resultado positivo*

The first keyword query was evaluated entirely over the `Paciente` table. However, the second keyword query required joining two tables, and was compiled into an SQL query as follows. The keyword *"enfermeiro"* matches values of attribute `Paciente.profissao`, generating the restriction "`Paciente.profissao = '*enfermeiro*'` " – note that the keyword "*profissão*" is not required, since attribute `profissao` is inferred from the match; the keyword "*município*" matches the name of attribute `Paciente.municipio`; the keyword *"teste rápido"* (treated as a single term due to the use of double quotes in the keyword query) partially matches values of attribute `Teste.tipo`, generating the restriction "`Teste.tipo='*teste rápido*'` "; the keywords "*resultado*" and "*positivo*" were likewise processed. The SQL query is:

```
select Paciente.id_paciente, Paciente.profissao,
       Paciente.municipio, Teste.tipo, Teste.Resultado
  from Paciente, Teste
 where Paciente.profissao = '*enfermeiro*'
   and Teste.tipo = = '*teste rapido*'
   and Teste.resultado = '*positivo'
   and Paciente.id_paciente = Teste.id_paciente
```

Observe that the target clause also contains attributes `Paciente.profissao`, `Teste.tipo`, and `Teste.Resultado` to reinforce that each row in the result table indeed indicates a match with the keywords in the keyword query. The reader is invited to try the keyword query { *enfermeiro município "teste rápido" resultado positivo* } at https://danke.tecgraf.puc-rio.br/covid-sus/.

Scenario 2: Global Data. To further illustrate the use of *CovidKeyS*, we chose the John Hopkins University (JHU) datasets and datasets from the Google COVID-19 Public Datasets program that refer to the pandemic evolution at the global level. The JHU data include the location and number of confirmed COVID-19 cases, deaths, and recoveries for all affected countries. From the Google COVID-19 Public Datasets repository, we chose the following country-level aggregated datasets: Oxford Policy Tracker, which summarizes COVID-19 policies at the country level; Google Mobility Report, which reports movement trends over time by geography across different categories of places; a global synthesis of COVID-19 cases by country; and the World Bank global data about

population, education, and external debt by country. In particular, Google Mobility dataset has sample mobility queries, which can be emulated with *CovidKeyS*.

An example keyword query over these tables would be:

- NL Query: "List the total number of deaths and the recreation mobility index on the same date for Belarus."
- Keyword query: *Belarus deaths recreation date*

The keyword query was compiled into an SQL query as follows. The keyword *"Belarus"* matches a value of attribute "`Country Name`" of table `Country` (from the World Bank global data about population); the keyword *"deaths"* partially matches the name of attribute "`Total Deaths`" and the keyword *"date"* matches the name of attribute `Date` of table `Contamination` (from the global synthesis of COVID-19 cases by country); and the keyword *"recreation"* partially matches the name of attribute "`Retail and Recreation Mobility Percentage`" of table `Mobility` (from the Google Mobility dataset). From these matches, equijoins between the keys of these tables are inferred (the keys of all three tables are composed of attributes "country name" and date) to create a coherent answer. Figure 1 shows the query result, which the user can then export as a CSV file and submit it to a different tool to analyze correlations between the retail and recreation mobility percentage variation and the number of deaths in a given period. The reader is invited to try this keyword query at https://danke.tecgraf.puc-rio.br/covid-global/.

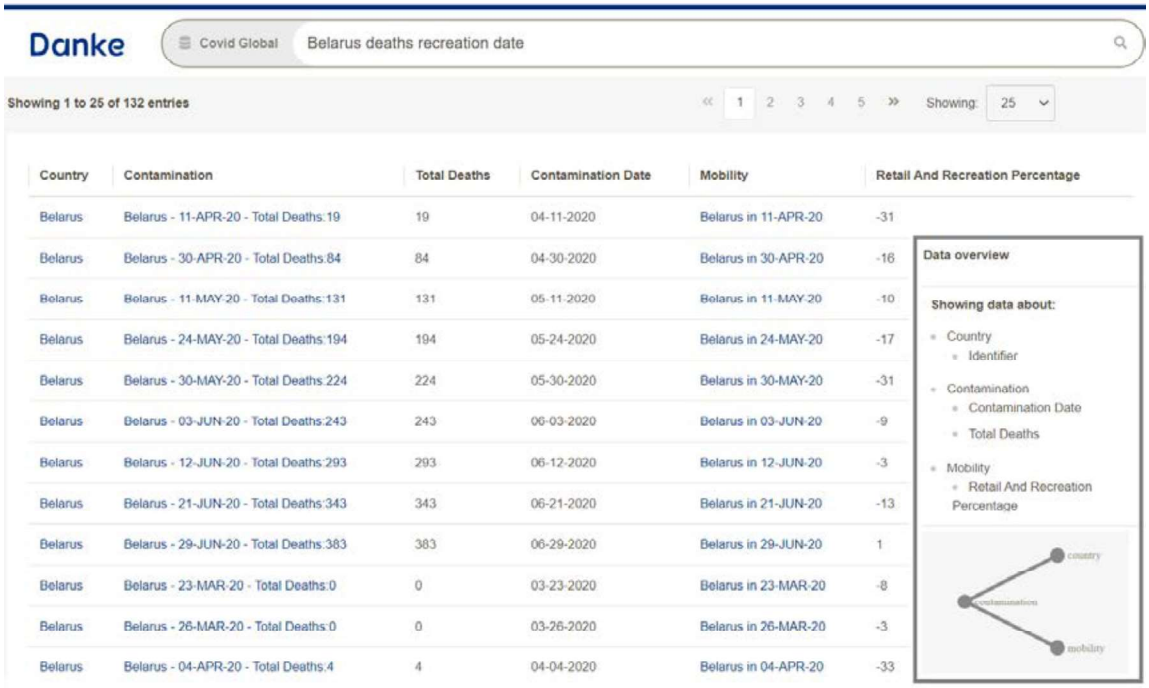

Figure 1. Result of the keyword query "Belarus deaths recreation date".

5. Conclusions

This paper first introduced a platform for data and knowledge retrieval, called DANKE, concentrating on the keyword search component. Then, the paper described an application, called *CovidKeyS,* that uses DANKE to implement keyword queries over two COVID-19 data scenarios. *CovidKeyS* is the central contribution of the paper.

We may point out at least two future developments. We plan to extend *CovidKeyS* to other COVID-19 data, perhaps including a crawler tool to capture such data. As for the platform, we are extending DANKE to operate over federated databases, as described in (Izquierdo et al. 2017), to support aggregation and group-by operations in natural language-like syntax, and to offer other graphical output features. We are also incorporating term expansion into DANKE, with the help of domain-specific thesauri.

Acknowledgments

This work was partly funded by grants CAPES/88881.134081/2016-01, CNPq/302303/2017-0, and FAPERJ/E-26-202.818/2017 and E-26/200.770/2019.

References

Bast, H., Buchhold, B., Haussmann, E. Semantic search on text and knowledge bases. *Found. and Trends® in Info. Retr.*, 10(1), (2016), 119-271. DOI: 10.1561/1500000032

Bergamaschi, S., Guerra, F., Interlandi, M., Trillo-Lado, R., Velegrakis, Y. Combining user and database perspective for solving keyword queries over relational databases. *Inf. Syst.* 55, C (Jan. 2016), 1-19. DOI: 10.1016/j.is.2015.07.005

García, G.M., Izquierdo, Y.T., Menendez, E., Dartayre, F., Casanova, M.A. RDF Keyword-based Query Technology Meets a Real-World Dataset. In: Proc. 20th Int'l. Conf. on Extending Database Technology (EDBT 2017), pp. 656-667.

García, G.M. A Keyword-based Query Processing Method for Datasets with Schemas. Thesis presented to the Graduate Program in Informatics, PUC-Rio (March 2020). DOI: https://doi.org/10.17771/PUCRio.acad.48728

Han, S., Zou, L., Yu, X., Zhao, D. Keyword Search on RDF Graphs - A Query Graph Assembly Approach. In: Proc. 2017 ACM Conf. on Information and Knowledge Management (CIKM 2017), pp. 227-236. DOI: 10.1145/3132847.3132957

Izquierdo, Y.T., García, G.M., Menendez, E.S., Casanova, M.A., Dartayre, F., Levy, C.H., QUIOW: A Keyword-Based Query Processing Tool for RDF Datasets and Relational Databases. In: DEXA 2018, LNCS 11030 (2018), pp. 259-269. DOI: 10.1007/978-3-319-98812-2_22

Izquierdo, Y.T., Casanova, M.A., García, G.M., Dartayre, F., Levy, C.H. Keyword Search over Federated RDF Datasets. In: Proc. ER Forum 2017 and ER Demo track co-located with the 36th Int'l. Conf. on Conceptual Modelling (ER 2017), CEUR Workshop Proc., Vol. 1979, CEUR-WS.org

Mello, L.E. et al. Opening Brazilian COVID-19 patient data to support world research on pandemics (July 30, 2020). DOI: 10.5281/zenodo.3966427

Oliveira, P., Silva, A., Moura, E. Ranking Candidate Networks of relations to improve keyword search over relational databases. In: Proc. IEEE 31st Int'l. Conf. on Data Engineering (ICDE 2015), pp. 399-410. DOI: 10.1109/ICDE.2015.7113301

Tran, T., Wang, H., Rudolph, S., Cimiano, P. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In: Proc. 2009 IEEE Int'l. Conf. on Data Engineering (ICDE 2009), pp. 405-416. DOI: 10.1109/ICDE.2009.119