

Integrating Scholarly Data with Knowledge Graphs

Henrique Noronha Facioli, Thiago Silva de Farias, Julio Cesar dos Reis

Relatório Técnico - IC-PFG-19-34

Projeto Final de Graduação

2019 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Integrating Scholarly Knowledge Graphs

Henrique Noronha Facioli, Thiago Silva de Farias, Julio Cesar dos Reis*

December 2019

Abstract

Scholarly data platforms have been extensively established internationally. However, although in the Brazilian academic community there is large academic research platforms, *e.g.*, Lattes, containing incredible amounts of information, it does not offer a structured way to query and analyze this data. Such information remains in an isolated way without connections with other sources, which prevents interesting data analyses. In this work, we design and develop the Knowlattes Graph framework for generating Knowledge graphs from scholarly data sources. We explore the current Lattes platform by extracting information from it to create a RDF triple store. In addition, we provide techniques to create links to external triple datasets, building a knowledge graph for scholarly data containing Brazilians researchers and publications.

1 Introduction

Searching for articles, journals and researchers has never been easier than today. On the web, there are several services that contain scholarly data from multiple universities and companies. This facilitates information dissemination, making publications more accessible to their audiences and fomenting collaboration between researchers from similar areas. Ultimately, this might improve knowledge acquisition and stimulate a more globalized and united scientific community.

The main cause for this ever more connected community is the so called *Linked Data*, which aims is promoting best practices for publishing and consuming semantically structured data on the web. This can turn the web easier to navigate by creating semantic relations among data sources, making data on the web ever more robust and also optimizing its access via queries, *i.e.*, searches or questions. In this sense, the ultimate goal of the Semantic Web [10] is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network.

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP

In this context, a Knowledge Graph (KG) refers to a group (two or more) *Linked Data* databases where it's possible to link subjects from each one of them to another using a proper predicate to connect them. Usually, it's implemented using triple store representation where the relations among any other entity on the base is created throughout the use of triples containing a subject-predicate-object, i.e., [João -*i* *typeOf* -*i* Person] where João is the subject, *typeOf* is the predicate and Person is the predicate of this triple). Using triples, all relations among the graph can be described and a semantic build through them.

When Google launched its KG [3], the search engine could show results related to the subject of query based on all other information that were stored in this graph via predicates, *i.e.*, the part of a sentence that gives information about the subject. This revolutionized how search engines worked, bringing even more information that the user asked for and establishing much more context surrounding their field of interest, opening a much wider world for people to explore.

Having in mind the potential behind this initiatives, some companies started to invest in this field. One of them is Springer Nature, a global scientific publisher dedicated to providing the best possible service to the whole research community. Their goal is to help authors to share their discoveries, enable researchers to find, access and understand the work of others and support librarians and institutions with innovations in technology and data. They are behind *SciGraph* [8], which is a Linked Open Data platform that aggregates data sources from Springer Nature and key partners from the scholarly domain. This consolidates information across the research landscape, for example, research projects, publications, authors and much more. This initiative demonstrates the potential of building a KG with scholarly data, improving the discovery connections on the vast sea of information pertaining the field of scholarly data.

While such scholarly data platforms are already established internationally, even offering some linked data API's for querying, there is still a long way to go for data belonging to the Brazilian academic community. Its most famous platform, Lattes [4], contains an amazing amount of information, offering countless analytical possibilities, even more when we consider the prospect of connecting its vast data with foreign datasets. This can facilitate the connection of Brazilians researchers with their international counterparts. However, information from the Lattes platform is in a unstructured form, which prevents its adequate linkage with other linked data repositories. The key challenges is how to extract the adequate information to generate triples and identify the correct resources to connect triples of distinct datasets.

In this work, we propose the Knowlattes Graph framework, which aims to create a KG from the lattes platform based on linked data principles. Our approach consists of building a triple store graph from data extracted of Lattes' HTML web pages. Our approach connects the generated triples to external datasets and offers an API for querying its results. Due to the large amount of data, the manual filling-in, and the

use of semi-structured data, there are several challenges in the use of Lattes as a source of data.

In our study, we deploy several analyses to demonstrate the potential of the framework for data analysis. To this end, 1000 pages from Lattes were parsed and added to our KG. We show the creation of links with the SciGraph dataset from the Springer Nature KG [8]. Our results indicate the potential of the framework to enable the generation of scholarly data in KGs contributing to the linked data initiative.

The remaining of this document is organized as follows: Section 2 presents fundamental concepts related to Semantic Web techniques. In addition, this section discusses related work. Section 3 describes our proposal for building the triple store graph and connecting it with external sources. We present the implemented algorithms and key decisions when implementing it. Section 4 reports on some analyses providing some use cases of our framework. Section 5 discusses how our Knowlattes graph system could be expanded and improved. Finally, Section 6 refers to the final thoughts and conclusions.

2 Theoretical Background and Related Work

In this section, we start by defining relevant terms for this study in Subsection 2.1. Subsection 2.2 discusses related work.

2.1 Definitions and Terminology

2.1.1 Ontology and OWL

In the context of information science, there are many definitions of ontology, some of which may even contradict one another. For this study, an ontology is a formal explicit description of concepts in a domain of discourse (classes), as well as properties describing various features, attributes or relationships that these classes may have [19]. In other words, an ontology provides a shared vocabulary, which can be used to model a domain by defining which types of concepts (or classes) exist, and their properties and relations [16].

Classes are the focus of most ontologies. Classes describe concepts in the domain. For example, a class of shapes represents all shapes, and specific shapes are instances of this class. A class can have subclasses that represent concepts that are more specific than the superclass. For example, we can divide the class of all shapes into circles and squares, as shown in Figure 1.

W3C Web Ontology Language (OWL) [5] refers to one of the most common ways in computationally representing ontologies. It is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge

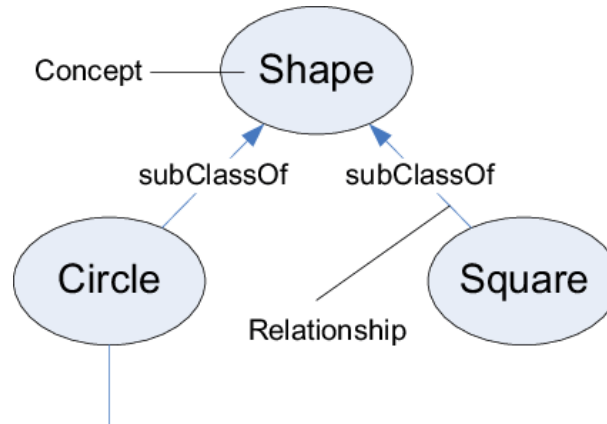


Figure 1: Example of an Ontology

expressed in OWL can be exploited by computer programs. It is also part of the W3C's Semantic Web technology stack.

An ontology together with a set of individual instances of classes constitutes a knowledge base. In the next section, we further cover about knowledge bases and the fine line in which the ontology ends and the knowledge base begins.

2.1.2 RDF and Structured Knowledge Bases

In this work, we extensively use another part of the W3C's Semantic Web technology stack: the Resource Description Framework (RDF) [6]. It is an information modeling standard that facilitates data merging across the information of different underlying schemas. This has features that facilitate data merging even if the underlying schemas differ. RDF is one of the key aspects in the Semantic Web, which is an effort towards organizing information with machine-readable semantics, and thus allowing connecting and processing distributed knowledge [9].

It revolves around the concept of Linked Open Data (LOD) [13]: free, open-source, reusable data composed by URIs (Uniform Resource Identifiers) and connected to other datasets that follows the LOD Principles[14]:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs, so that they can discover more things

Currently, several projects provide data in this standard by increasing its reach. One meaningful example is *DBPedia* [1], a community effort to extract structured data from Wikipedia projects. An RDF graph is composed of a set of triples formed by subject, predicate, and object (cf. Figure 2 A). The subject denotes the resource, and the predicate denotes traits or aspects of the resource, and expresses a relationship between the subject and the object. These resources are defined by a URI as an address that unambiguously identifies an element of the triple. The set of triples generates an oriented graph, which can be queried.

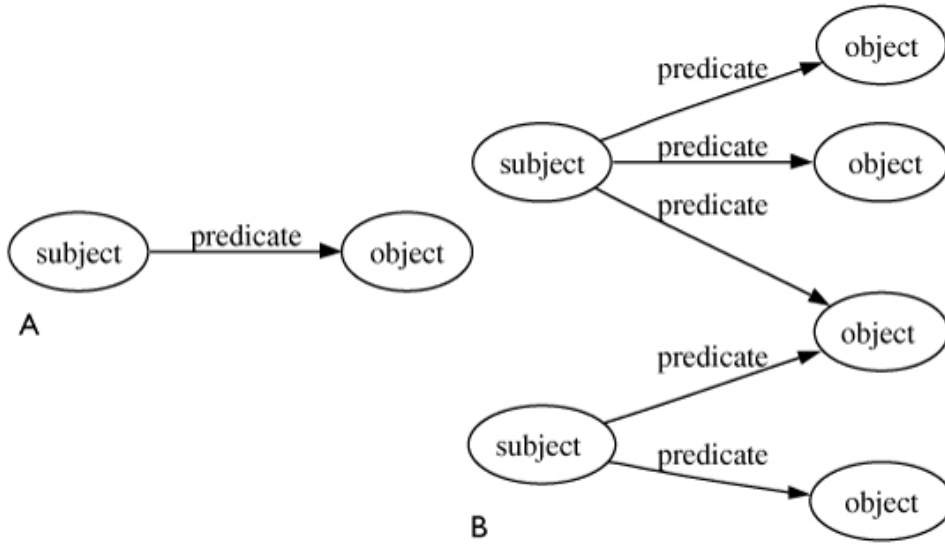


Figure 2: Example of an RDF triple (A) and a set of combined triples forming a linked data structure

For example, one way to represent the notion “The article X was published by Sigmund Freud” in RDF is as the triple: a subject denoting “the article X”, a predicate denoting “was published by”, and finally an object denoting “Sigmund Freud”.

An RDF graph is also called a Knowledge Graph (KG). Most of the knowledge graphs follow a simple principle: organizing information in a structured way by explicitly describing the relations among resources, as shown in Figure 3.

2.1.3 Querying with SPARQL

Data in a Knowledge Graph can be accessed using a query language such as SPARQL (SPARQL Protocol and RDF Query Language). This allows the user to retrieve and manipulate data by specifying a set of constraints (graph patterns). The information queried can be related to the subject, object, predicate or any combination of those three, and a range of combinations is available [11].

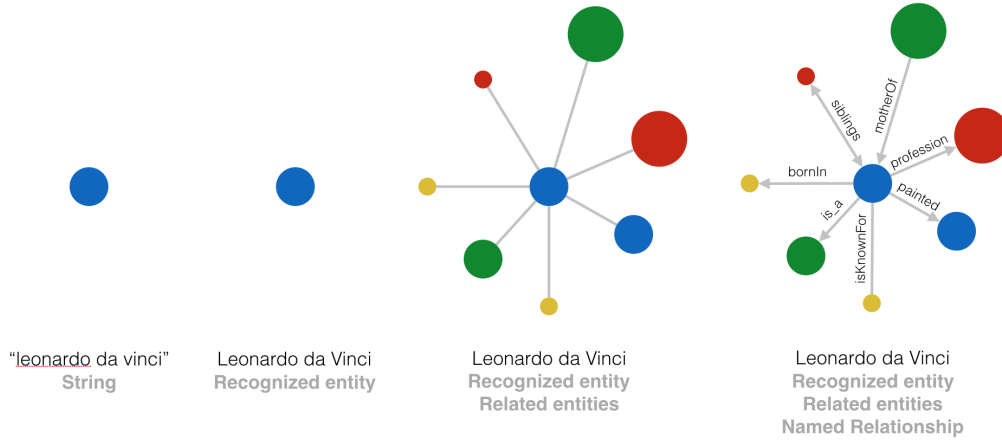


Figure 3: Example of Knowledge Graph implementation [2]. From strings to things, knowledge graphs aim to structure what is known about the world, making information much easier to discover

In the following, we present an example of a SPARQL query. It is designed to output the movies, and their respective directors, belonging to the comedy genre. This data is collected from a fictional knowledge graph called “Films_Knowledge_Graph”, from which the resources descriptions and their relationships are based on the “fictional” ontology “Media_Ontology”.

```

1 PREFIX mo: <http://example.com/Media_Ontology#>
2 SELECT ?movie ?director
3 WHERE {
4     ?x mo:directorName ?director .
5     ?x mo:directs ?y .
6     ?y mo:movieName ?movie .
7     ?y mo:genre mo:Comedy .
8 }
9 FROM Films_Knowledge_Graph

```

The first line in the SPARQL example indicates that we are using the fictional Media Ontology. In the second line, we state that we are interested in (and therefore *selecting*) the movies and directors from our knowledge graph. In lines 3 through 8, we define the set of constraints that we want our results to match, such as:

- “x” URI must be a director with a “directorName” property.
- “x” must direct the movie defined by the “y” URI.

- “y” URI must be a movie with a “movieName” property.
- “y” must be a movie of the comedy genre.

This query would then return all the movies from the knowledge graph that are considered a comedy, as well as the movies’ directors.

2.2 Related Work

We report a brief literature review concerned with existing investigations exploring KGs mostly focusing in scholarly data. Szekely *et al.* [20] presented an approach to build knowledge graphs by exploiting semantic technologies to reconcile the data continuously crawled from diverse sources. Their aim was to scale to billions of triples extracted from the crawled content, and to support interactive queries on the data. They proposed a system that acquires data, represents all data in a common ontology, defines URIs for all entities, links them to external Semantic Web resources (*e.g.*, Geonames), generates a knowledge graph and displays it with an interface. The results of their study were deployed to six law enforcement agencies and several non-governmental organizations to assist them with finding traffickers and helping victims. In addition to the results achieved, their work served as an important source of knowledge towards the implementation of this study.

There are studies dealing with scholarly data, in particular dealing with extracting and using the vast source of information for the creation and analysis of researchers’ social networks which from the Lattes Platform. Digiampietri *et al.* [15] presented an study as database produced from the mining of more than one million of Brazilian Lattes curricula. The authors highlight some descriptive characteristics and relationships among these curricula and among the knowledge areas, directions and challenges to the production and analyzes of social networks generated from these data.

On the other hand, Mena-Chalco and Cesar Junior [17] described in their work the design, implementation and experiences with scriptLattes, which is an open-source system to create academic reports of groups based on curricula of the Lattes Database. The scriptLattes system’s source code, usage instructions and examples are available at ¹. The reuse and adaptation of their project was a big part of accomplishing our study.

Castañó [18] explored the use of ontologies to model data from the Lattes Platform. His work created and populated an ontology with resumes from the Lattes platform to be used mainly as a database to be queried for reports generation. It describes the obstacles encountered and how they were solved. Such work was relevant to guide us towards surpassing such challenges. It goes into details regarding the process to

¹<http://scriptlattes.sourceforge.net/>

model the ontology, read, interpret and insert the information from the resumes in this model.

The study conducted by Vahdati *et al.* [21] tackled the problem of knowledge discovery in scholarly knowledge graphs. They used a knowledge-driven framework able to unveil scholarly communities for the prediction of scholarly networks. Results observed from their evaluations suggested that exploiting semantics in scholarly knowledge graphs enables the identification of previously unknown relations between researchers. By extending such concept, these observations can be generalized to other scholarly entities (*e.g.* articles or institutions) as well as be used for the prediction of other scholarly patterns, such as academic collaboration.

There are existing work approaching the proposal to create a Linked Open Data platform that collates information across the academic landscape. The academic publishing company Springer Nature built such a platform, called *SciGraph*². This aggregates data sources from Springer Nature and key partners from the scholarly domain. There are several benefits of having such system, including:

1. Authors and editors enjoy easy access to high quality data from trusted and reliable sources.
2. Funders, librarians, conference organizers find optimal data for analysis and recommendation tools.
3. Users of the scholarly domain broaden their perspective by semantic relations being revealed visually.
4. Researchers may benefit of overcoming internal and external data silos in research communities, *i.e.*, organizations that might place obstacles in accessing their data.

One of the biggest challenges of this study was linking the data to external sources. Several studies have been done regarding this issue. The state of the art is described by Volz *et al.*, in the work “Silk – A Link Discovery Framework for the Web of Data” [22]. Their work produced several features that assist in linking different sources of data, providing different ways to assess both how the data can be matched and to what extent it must be similar to the external source to create a link.

3 Knowlattes Graph

In this study, we create a knowledge graph based on the information from the Lattes Platform [4] database. Also, we aim to establish relations between articles and researches based on an international scholarly RDF dataset available. We sample

²<https://www.springernature.com/gp/researchers/scigraph>

some Lattes accounts and build a triple store graph using RDF triples. Relying on this sample, we provide some analyses and queries to provide highlights on the data. Next, our KG generated based on lattes page is connected to another KG, already built, as SpringerNature SciGraph [8].

Figure 4 presents the key modules of the Knowlattes graph framework developed in this work. This is organized as follows:

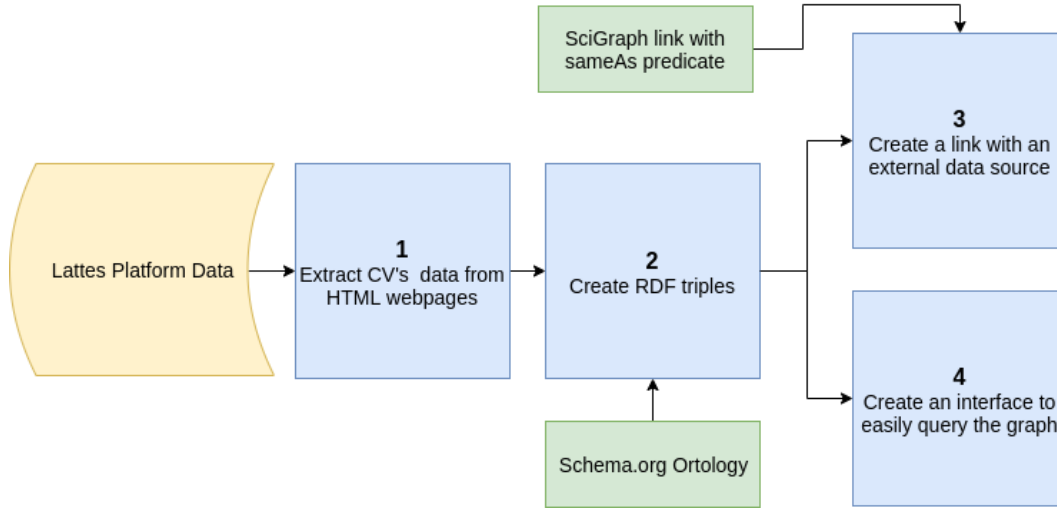


Figure 4: Knowlattes Graph framework modules

1. Parse Lattes publication platform (cf. Subsection 3.1)
2. Generate the triples with the Lattes pages (cf. Subsection 3.2)
3. Connect the lattes triples with another publication platform (cf. Subsection 3.3)
4. Create an interface to allow users to query our triple store (cf. Subsection 3.4)

All these steps were implemented and the source code can be found on project Github page³

3.1 Parsing Lattes publication platform

Lattes platform does not provide any API for querying their data, turning its data extraction hard. In this sense, we implemented an HTML parser capable of getting the needed information. Knowlattes graph made use of an open-source project,

³<https://knowlattesgraph.github.io>

scriptLattes [17], with the need for code modification for updating it to deal with recent changes on the CV platform. The code gets the HTML source from the lattes CV and parses it, searching for HTML elements to build a object that contains all the information from the page. The Algorithm 1 presents the implemented procedure for this purpose.

Algorithm 1: Parsing Lattes CV as HTML and return the Structured Data

Data: Lattes HTML Page

```

1 LattesCV  $\leftarrow$  {} ;
2 while HTML tags exists do
3   | get current line tag;
4   | if tag is related to sectionX then
5   |   | LattesCV  $\leftarrow$  all information from the sectionX + LattesCV ;
```

Result: An object containing all the HTML information

The page in Figure 5 shows an example of the HTML page that is parsed. When parsing the page, the parser gets the researcher's name, the researcher's Lattes ID and a brief researcher description. The result contains all the needed information that could exist on the web page, making it easy to manipulate the data in the following steps of our methodology.

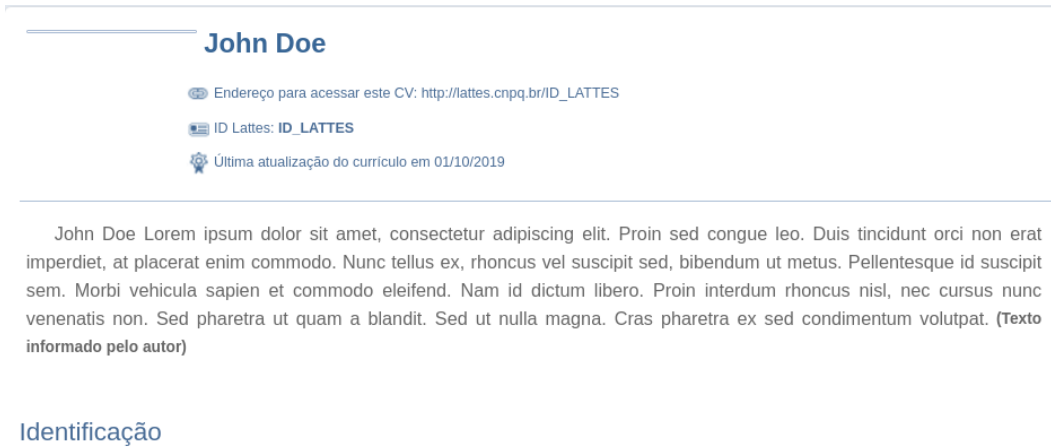


Figure 5: Example of a Lattes Curriculum page. Here there is the Lattes ID, name and a brief description of the researcher that will be parsed and placed on the python object

Figure 6 presents an internal representation of how the extracted data from the Lattes is encoded in our framework.

For evaluation purpose, in the Knowlattes Graph framework, there were 85.000

LattesCV_JohnDoe
+ name: String = John Doe
+ ID: String = ID_LATTES
+ description: String = Lorem Ipsum...
(...)

Figure 6: Representation of a LattesCV object after parsing the HTML

pages available for building the graph. This can be found on GitHub repository ⁴

3.2 Creating the Triple Store

After parsing the data from the Lattes pages, we move on to start building our triple store graph. The next step towards that goal is to create triples from the parsed data.

We made use of *Schema.org*⁵ to structure how the data is represented in our triple store, which helps to establish the ontology in which the RDF triples are being built. Schema.org is a collaborative set of ontologies built by an agreement of big companies such as Google, Microsoft, and Yahoo. It has been used across multiple platforms when needing vocabularies to come to a relation between the elements. Knowlattes decision to use it came because Schema.org is growing and has become widely used on multiple projects. Also, *Springer Nature SciGraph*⁶ implemented their scholarly data triple store by using it. The use of *Schema.org* facilitates further data integration and extension of the KG.

Triples are created by iterating over the parsed data in our internal representation, that contains a Lattes page, creating the appropriate triples based on the data extracted i.e: Person, Article, Research.

Figure 6 shows a representation of the triples that are created when parsing the object. From Figure 6 one of the triple created would be [ID_LATTES - *schema:type* - *schema:Person*] and another would be [ID_LATTES - *schema:name* - John Doe]. All the possible type of entries on the graph are described on the Appendix section, which contains the correlation of lattes pages data and the triples, in addition to how the data is transformed before being included in the graph.

Both the creation of the graph itself, as well as the addition of triples to the graph was performed by using RDFLib library [7], which is a Python package for working

⁴<https://github.com/knowlattesgraph/lattes-cached-pages>

⁵<https://schema.org>

⁶https://scigraph.springernature.com/explorer/datasets/data_at_a_glance/, accessed on December 2019

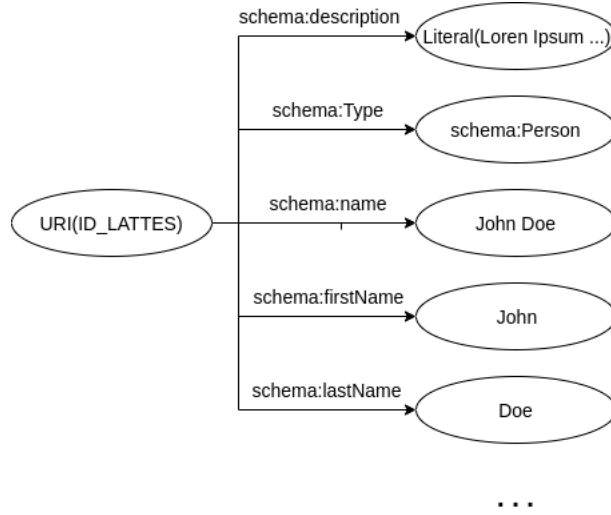


Figure 7: Triples extracted from a Lattes CV page. This is based on the object shown in Figure 6

with RDF that supports both in memory database or persistent using plugins.

3.3 Connecting with external graphs

Establishing the triples based on the data from the Lattes Platform built a triple store graph that contained only the relation inside the graph, making it a "data island", isolated from all external world. The next step is to link this triple store with external sources. In this study, we used the vast dataset available at *Springer Nature* [12], which contains more than 8 million articles and more than 7 million persons, all concerning scholarly data.

The connection between our knowledge graph and the external datasets was done utilizing the *schema:sameAs* predicate. This predicate is a key part of the Semantic Web, as it is used to indicate that the subject of two resources is considered to be the same thing in multiple stores. Figure 8 presents an example of external link to connect RDF triples from different RDF datasets. The links are created as triples by using the *schema:sameAs* predicate.

In our study, we linked articles from the Lattes Platform with articles in the *Springer Nature* dataset by using their Digital Object Identifier (DOI) number, which is a unique alphanumeric label created to identify a digital object. DOI will never change even if the object itself changes its physical location. So, based on the article DOI, the Knowlattes Graph creates a triple using the predicate *sameAs* from the ontology to link one local article with its URI (based on DOI) from the external base, creating the triple [URI(Article), *schema:sameAs*, URI_EXTERNAL(Article)].

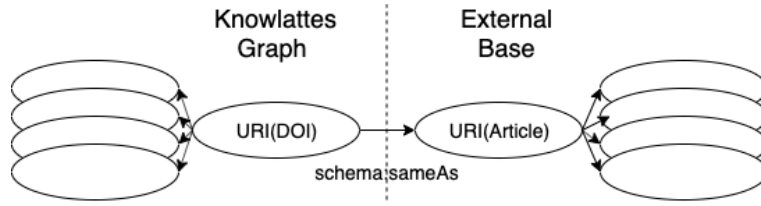


Figure 8: Example of an external link of data through the predicated *sameAs*

The algorithm 2 generally describes the implementation.

Algorithm 2: Linking with Springer Nature Datasets

Data: Triple Store, External Springer Nature Dataset

```

1 for publication in all publications do
2   if publication contains DOI then
3     | adds the triple (publication, schema:sameAs, spgrNature_link) ;
```

Result: Triple Store linked with external source

Based on the implemented procedure, publications in Lattes that contain DOI and exist on the SciGraph dataset can be connected by the predicate *sameAs*. Section 5 discusses how other entities from Knowlattes graph could have been connected with external sources.

3.4 Interface for querying the database

Given the structured triple store graph, one should be able to query it and get the expected results easily without the need to generate the graph each time. For this purpose, Knowlattes Graph implements a persistent triple store using the RDFLib plugin by exploring abstraction to store it on a SQL database.

With a persistent store, a web API was built using Python Flask framework where a SPARQL query can be written on the search field, queried over the Knowlatter Graph base and, as result, tabular data is returned and displayed as presented in Figure 9. The webpage to query the sample used on the paper can be found on a live demo located⁷

4 Deployed Analyses

This section provides some analyses of what could be some potential uses of our data. To this end, we applied our framework to generate a KG from a sample equivalent to 1000 randomly chosen Lattes pages. The size of the sample in this evaluation

⁷<https://knowlattes.herokuapp.com/>

Knowlattes Graph

```
SELECT ?author ?project ?year ?type
WHERE {
  ?project schema:author ?uri .
  ?uri schema:givenName ?author .
  ?project schema:datePublished ?year .
  ?project schema:type ?type
}
LIMIT 10
```

Results

Stéphano	10.1016/j.biopha.2019.108733	2019	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.1002/cbin.11162	2019	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.1016/j.bbrc.2019.05.103	2019	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.7150/ijms.23150	2018	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.1590/1414-431x20176146	2017	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.3390/ijms16046855	2015	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.1186/s12872-015-0156-4	2015	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.1159/000358642	2014	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.1249/mss.0b013e318217e8b6	2011	http://schema.org/version/latest/schema.nt#ScholarlyArticle
Stéphano	10.1590/s0066-782x2009001100004	2009	http://schema.org/version/latest/schema.nt#ScholarlyArticle

Figure 9: Knowlattes Graph web interface made for executing SPARQL queries and retrieve information

was chosen due to processing power and memory limitations. Nevertheless, the insights obtained from our study based on this sample can be extrapolated for the entire dataset.

4.1 Most common predicates

In order to identify the type of information contained in our graph, we must look at its edges, in other words, its predicates. This metric provides not only an overview of the type of data contained in our triple store, but also how it is connected to Scigraph’s dataset.

The query on Listing 1 returns the top 10 predicates by frequency. Figure 10 shows the output obtained from using this query in our KnowLattes Query Maker.

These results indicate two main findings:

1. Our sample dataset of 1000 Lattes pages presents more than 28000 URI, for example, as all the URI have a `schema:type`
2. There is an average of 12.7 articles per author that contains a valid link with a Springer Nature dataset article.

Listing 1: Querying the top 10 predicates on the sampled Knowlattes Graph

```

1  SELECT ?predicate (COUNT(*) AS ?frequency)
2  WHERE {?subject ?predicate ?object}
3  GROUP BY ?predicate
4  ORDER BY DESC (?frequency)
5  LIMIT 10

```

4.2 Average Number of Publications per Author

Another important aspect of the scholarly community is understanding through which media type information is being published. Through that, authors would acquire valuable knowledge of how best to contribute with their peers to maximize the reach of their studies.

The query on Listing 2 looks at the objects of triples considering the edges of *schema:type* predicates from our knowledge graph. The predicates define the nature of the triples present in our sample data. Figure 11 shows the output obtained from using this query in our KnowLattes Query Maker.

Results

http://schema.org/version/latest/schema.nt#type	28016
http://schema.org/version/latest/schema.nt#name	28006
http://schema.org/version/latest/schema.nt#author	26298
http://schema.org/version/latest/schema.nt#datePublished	21093
http://schema.org/version/latest/schema.nt#knows	19974
http://schema.org/version/latest/schema.nt#genre	16732
http://schema.org/version/latest/schema.nt#sameAs	12658
http://schema.org/version/latest/schema.nt#description	5610
http://schema.org/version/latest/schema.nt#year	5572
http://schema.org/version/latest/schema.nt#editor	4004

Figure 10: Top 10 predicates in KnowLattes Graph

Listing 2: Querying the number of publication per author

```

1  SELECT ?object (COUNT(*) AS ?frequency)
2  WHERE {?subject schema:type ?object}
3  GROUP BY ?object
4  ORDER BY DESC (?frequency)

```

Results

http://schema.org/version/latest/schema.nt#ScholarlyArticle	12658
http://schema.org/version/latest/schema.nt#ResearchProject	5539
http://schema.org/version/latest/schema.nt#TechArticle	4066
http://schema.org/version/latest/schema.nt#Book	3897
http://schema.org/version/latest/schema.nt#EducationalOrganization	945
http://schema.org/version/latest/schema.nt#Person	891
http://schema.org/version/latest/schema.nt#Language	20

Figure 11: Objects with predicate *schema:type*

We can indicate the following aspects from these results:

1. On average, an author has 25 publications, out of that, 12 being an scholarly article, 5 a research project, 4 tech article and 3 books/chapters contributions.
2. Out of the 26160 entities analyzed, the most common are, respectively, scholarly articles (12658, 48%), research projects (5539, 21%), technical articles (4066, 16%) and books (3897, 15%).

4.3 Utilizing the external dataset

In this analyses, we give a step by step example of how the connection to the external SciGraph dataset works, by querying articles with the *schema:sameAs* predicate and then, based on its URI, opening it in the Springer Nature platform.

We start by using our KnowLattes Query Maker to extract articles (their DOI's and their titles) by the author “Maria Lúcia Harada” that contains the *schema:sameAs* predicate, or, in other words, have a match in the SciGraph dataset. The query on Listing 3 presents the implemented analysis. Figure 12 presents the output of the results from query 3.

Listing 3: Searching for the authors name on Knowlattes

```

1 SELECT ?doi_sn ?title
2 WHERE {
3     ?doi schema:author ?uri .
4     ?uri schema:familyName "Lucia Harada" .
5     ?doi schema:name ?title .
6     ?doi schema:sameAs ?doi_sn
7 }
8 ORDER BY ?doi
9 LIMIT 5

```

As a matter of example, in Figure 12, we highlighted in orange one particular article, called *Diversity, Geographic Distribution and Conservation of Squirrel Monkeys, Saimiri (Primates, Cebidae), in the Floodplain Forests of Central Amazon*. This article contains the DOI *10.1007/s10764-013-9714-8*. Figure 13 presents the equivalent to this article in the SciGraph Data Explorer:

In our solution, by connecting this article to the SciGraph dataset expands the number of options that the user has to visualize and manipulate data. Figure 13 presents that in the SciGraph platform offers further data representation such as: a graph visualization for the data pertaining the article; a list of all the triples; and several formats to export the data for development use by including JSON-LD, N-triples, Turtle and RDF/XML.

Results


http://scigraph.springernature.com/pub.10.1007/s10238-005-0077-0	Molecular study of the tumour suppressor gene PTEN in gastric adenocarcinoma in Brazil
http://scigraph.springernature.com/pub.10.1007/s10238-010-0122-5	Survivin -31C/G polymorphism and gastric cancer risk in a Brazilian population
http://scigraph.springernature.com/pub.10.1007/s10764-013-9714-8	Diversity, Geographic Distribution and Conservation of Squirrel Monkeys, Saimiri (Primates, Cebidae), in the Floodplain Forests of Central Amazon
http://scigraph.springernature.com/pub.10.1007/s11738-011-0788-7	Identification of sequences expressed during compatible black pepper Fusarium solani f. sp. piperis interaction
http://scigraph.springernature.com/pub.10.1007/s13277-013-0742-y	Analysis of the methylation patterns of the p16 INK4A , p15 INK4B , and APC genes in gastric adenocarcinoma patients from a Brazilian population

Figure 12: Output from query on Listing 3.

SN SciGraph Data Explorer 2019
Getting Started
Datasets
Releases
Downloads
License
FAQ

Search
Go

YOU ARE HERE: HOME / ARTICLES / <http://scigraph.springernature.com/pub.10.1007/s10764-013-9714-8>



Diversity, Geographic Distribution and Conservation of Squirrel Monkeys, Saimiri (Primates, Cebidae), in the Floodplain Forests of Central Amazon
View Full Text

Ontology type: schema:ScholarlyArticle

Overview
Identifiers
Visual
JSON-LD
Triples
Developers

Article Info

DATE
2013-10

AUTHORS
[Fernanda Pozzan Paim](#), [José de Sousa e Silva Júnior](#), [João Valsecchi](#), [Maria Lúcia Harada](#), [Helder Lima de Queiroz](#)

Journal

TITLE
[International Journal of Primatology](#)

ISSUE
5

VOLUME
34

Figure 13: Corresponding article in the SciGraph Data Explorer platform

5 Discussion

Knowlattes Graph framework aims to discover, study and be an entrance guide on the web semantics related to scholarly data for the Brazilian platform. It presents the capabilities of: 1) parsing a lattes HTML page; 2) extract all the needed information; 3) build a triple store graph; and 4) link the article to external datasets. We conducted a case from information from Lattes platform by exploring 1000 pages. Our deployed analyses showed that our framework is able to get insights on the data. With a greater scope, it would be able to provide further information such as: the distance of two researchers; recommendations for the researcher when finding a subject; and even finding the researches that publish similar topics.

At the present stage of development, when linking with external platforms, our current implementation deals with articles that contain DOI. However, it would be a further improvement to link researchers and even other types of resources present in the KG. Linking researchers among multiple bases have the challenge of discovering the correct subject on the foreign base that is the same one on Knowlattes. Usually, as an example, by using the authors' name of an article can produce more than one match in external datasets. The creation of links based on string processing and matching is a challenging task. This represents a big obstacle for creating links via strings, *i.e.*, sequences of characters. By only using the name string is difficult to know to which author the link should be established, or if it should be established for more than one.

In addition to the linking based on the article's DOI, future work could establish links based on other entities, such as the author's names, research projects and book titles. To accomplish this, the study of link discovery techniques such as the Silk [22] can be a tool to explore and find relationships between entities within different data sources. In this sense, we should specify which types of RDF links should be discovered between data sources as well as which conditions entities must fulfill to be interlinked.

Future investigations could tackle both the further implementation of Knowlattes and the content behind the KG. When it comes to implementation, some improvements could be made on how the data is stored, for instance, by using storages that are optimized for graphs. This would make our system further scalable than it currently is as our current implementation uses an abstraction over a non-optimized SQL database. Furthermore, our Query Maker could offer an API so that other services could more easily and intuitively utilize our dataset.

6 Conclusion

This work designed and implemented the Knowlattes Graph framework to extract and interlink scholarly data. This study addressed the scope of the Brazilian scholarly community, in particular, its most famous platform, Lattes. We provided a solution to move forward into making its data more available and connected to other established RDF datasets. We implemented an algorithm that parses data from the Lattes Platform, transforms it into a RDF triplestore as semantically structured data. Our solution connects created triples to an external dataset and then makes it publicly available through a web interface. This allows the knowledge graph to be queried. The results from the deployed analyses based on the Lattes information provided insights into how the data is distributed considering information about publications and researchers. We showed the feasibility in linking such data to external RDF graphs. Future work involves addressing additional features of the framework to link other types of entities.

References

- [1] Dbpedia, accessed in 06/08/2019. <https://wiki.dbpedia.org/>.
- [2] From strings to things, accessed in 27/08/2019. <https://medium.com/@sderymail/challenges-of-knowledge-graph-part-1-d9ffe9e35214>.
- [3] Google knowledge graph, accessed in 27/08/2019. <https://developers.google.com/knowledge-graph/>.
- [4] Lattes Platform, accessed in 19/08/2019. <http://lattes.cnpq.br/>.
- [5] Owl - w3, accessed in 21/08/2019. <https://www.w3.org/2001/sw/wiki/OWL>.
- [6] Rdf - w3, accessed in 06/08/2019. <https://www.w3.org/RDF/>.
- [7] Rdflib documentation, accessed in 02/09/2019. <https://rdflib.readthedocs.io/en/stable/>.
- [8] Scigraph website, accessed in 21/08/2019. <https://www.springernature.com/br/researchers/scigraph>.
- [9] Semantic web - w3, accessed in 06/08/2019. <https://www.w3.org/standards/semanticweb/>.
- [10] Semantic web - wikipedia, accessed in 06/12/2018. https://en.wikipedia.org/wiki/Semantic_Web.

- [11] Sparql - w3, accessed in 06/12/2018. <https://www.w3.org/TR/rdf-sparql-query/>.
- [12] Springer nature: Datasets at a glance, accessed in 22/10/2019. https://scigraph.springernature.com/explorer/datasets/data_at_a_glance/.
- [13] Tim berners-lee, linked data, accessed in 06/08/2019. <https://www.w3.org/DesignIssues/LinkedData.html>.
- [14] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5:1–22, 2009.
- [15] Luciano A. Digiampietri, Jesús P. Mena-Chalco, José J. Pérez-Alcázar, Esteban F. Tuesta, Karina V. Delgado, Rogério Mugnaini, Gabriela S. Silva, and Jamison J. S. Lima. Extração, caracterização e análises de dados de currículos lattes. 2015.
- [16] Diana Man. Ontologies in computer science. In *Didactica Mathematica*, pages 43–46, 2013.
- [17] Jesús Pascual Mena-Chalco and Roberto Marcondes Cesar Junior. scriptlattes: an open-source knowledge extraction system from the lattes platform. *Journal of the Brazilian Computer Society*, 15(4):31–39, December 2009.
- [18] Jesús Pascual Mena-Chalco and Roberto Marcondes Cesar Junior. scriptlattes: an open-source knowledge extraction system from the lattes platform. *Journal of the Brazilian Computer Society*, 15(4):31–39, Dec 2009.
- [19] Natalya F. Noy and Deborah L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, March 2001.
- [20] Pedro Szekely, Craig A. Knoblock, Jason Slepicka, Andrew Philpot, Amandeep Singh, Chengye Yin, Dipsy Kapoor, Prem Natarajan, Daniel Marcu, Kevin Knight, David Stallard, Subessware S. Karunamoorthy, Rajagopal Bojanapalli, Steven Minton, Brian Amanatullah, Todd Hughes, Mike Tamayo, David Flynt, Rachel Artiss, Shih-Fu Chang, Tao Chen, Gerald Hiebel, and Lidia Ferreira. Building and using a knowledge graph to combat human trafficking. In Marcelo Arenas, Oscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d’Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab, editors, *The Semantic Web - ISWC 2015*, pages 205–221, Cham, 2015. Springer International Publishing.
- [21] Sahar Vahdati, Guillermo Palma, Rahul Jyoti Nath, Christoph Lange, Sören Auer, and Maria-Esther Vidal. Unveiling scholarly communities over knowledge

- graphs. In Eva Méndez, Fabio Crestani, Cristina Ribeiro, Gabriel David, and João Correia Lopes, editors, *Digital Libraries for Open Knowledge*, pages 103–115, Cham, 2018. Springer International Publishing.
- [22] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk—a link discovery framework for the web of data. *Proceedings of the 2nd Linked Data on the Web Workshop*, 01 2009.

7 Appendix

Tables containing all the predicates in lattex pages

7.1 Researcher

Subject	Predicate	Object
URI(researcher)	<i>schema:type</i>	<i>schema:Person</i>
	<i>schema:name</i>	researcher.name
	<i>schema:familyName</i>	last(researcher.name)
	<i>schema:givenName</i>	first(researcher.name)
	<i>schema:gender</i>	researcher.gender
	<i>schema:knows</i>	[URI(researcher), ...]
	<i>schema:knowLanguage</i>	nomeIdioma
	<i>schema:alumniOf</i>	URI(institution)

7.2 Language

Subject	Predicate	Object
URI(nome_idioma)	<i>schema:type</i>	<i>schema:Language</i>

7.3 Academic Degree

Subject	Predicate	Object
URI(institution)	<i>schema:type</i>	<i>schema:EducationalOrganization</i>
	<i>schema:name</i>	institution.name
	<i>schema:hasCredential</i>	institution.academic_degree_type

7.4 Articles

Subject	Predicate	Object
URI(doi)	<i>schema:type</i>	<i>schema:ScholarlyArticle</i>
	<i>schema:name</i>	article.title
	<i>schema:datePublished</i>	article.year
	<i>schema:genre</i>	article.genre
	<i>schema:author</i>	[URI(researcher), ...]
	<i>schema:sameAs</i>	URI(SpringerPage)

7.5 Research

Subject	Predicate	Object
URI(research)	<i>schema:type</i>	<i>schema:ResearchProject</i>
	<i>schema:name</i>	research.name
	<i>schema:year</i>	research.year
	<i>schema:description</i>	research.description
	<i>schema:author</i>	[URI(researcher), ...]

7.6 Books and Chapters

Subject	Predicate	Object
URI(book_title)	<i>schema:type</i>	<i>schema:Book</i>
	<i>schema:name</i>	book.name
	<i>schema:datePublished</i>	book.year
	<i>schema:authors</i>	[URI(book_title), ...]
	<i>schema:isPartOf</i>	[URI(book_title), ...]
	<i>schema:editor</i>	book.publisher
	<i>schema:pagination</i>	book.pages

7.7 Tech Articles

Subject	Predicate	Object
URI(tech_article)	<i>schema:type</i>	<i>schema:TechArticle</i>
	<i>schema:name</i>	tech_article.title
	<i>schema:datePublished</i>	tech_article.year
	<i>schema:genre</i>	tech_article.genre
	<i>schema:author</i>	[URI(researcher), ...]