Searching for Researchers: an Ontology-based NoSQL Database System Approach and Practical Implementation

Mariana D. A. Salgueiro[1], Veronica dos Santos[1], André L. C. Rêgo[1], Daniel S. Guimarães[1], Jefferson B. Santos[2], Edward H. Haeusler[1], Marcos V. Villas[1] and Sérgio Lifschitz[1]

[1] Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
{msalgueiro,vdsantos}@inf.puc-rio.br, {andreregо,danielg}@aluno.puc-rio.br
{hermann,villas,sergio}@inf.puc-rio.br
[2] Escola Brasileira de Administração Pública e de Empresas
Fundação Getulio Vargas (FGV)
jefferson.santos@fgv.br

Abstract. This work presents the design and implementation of two web-based search systems, Busc@NIMA and Quem@PUC. Both systems allow the identification of research and development projects, besides existing competencies in laboratories and departments involving professors and researchers at PUC-Rio University. Our applications are based on a list of search-related terms that are matched to the dataset composed of PUC-Rio's Lattes CVs offered courses, information from administrative systems, and specific keywords that are input by the professors/researchers themselves. To integrate all the needed data, we consider multiple database and search technologies, such as XML, RDF, TripleStores, and Relational Databases. Search results include professor's name, academic papers, teaching activities, contact links, keywords, and laboratories of those involved with the subject represented by the set of keywords input. We describe the main features that show how our systems work.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: JIDM, SBBD, template

1. INTRODUCTION

How should one proceed when looking for experts in environmental studies? Or perhaps, more generically, how should one move when looking for experts in any domain? These questions can arise in many different ways: (i) when there is a need to look for an expert that could give an interview in a newspaper or (ii) even for students who want to deepen their studies and find those who can guide them. At every University, such as PUC-Rio and other well-known institutions, this kind of situation where people are looking for experts happens a lot.

One possibility would be to perform a web search with some search-related terms and add the expression "PUC-Rio." Another option would be to use the Lattes platform, powered by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), to search by Subject (title or production keywords) inside the researches' CVs. A third alternative would be to contact the "University Communication Area" and request the recommendation and contact details of the expert, which can also be difficult to find inside the university community.

Due to the research and academic information being spread across several sources, this project aims to bring them together in a single database and create a search engine capable of returning, through a keyword list, results in an organized way and with the proposal of doing this quickly and efficiently.

Identifying experts given a particular subject is also known as an expertise retrieval task. It is supported by information retrieval systems that can automatically rank individuals based on their expertise on a specific subject [Husain et al. 2019]. Identifying expert individuals from PUC-Rio can also help find research collaborators, paper reviewers, student supervisors, or to review literature from the point of view of a domain specialist. We target the academic experts' retrieval based on a scholarly knowledge graph in this work.

This article is an extended version of our previous article [Salgueiro et al. 2021]. It describes the development of an Information Retrieval System to discover professors/researchers from PUC-Rio based on search-related terms, in a specialized way named Busc@NIMA or in a generic way called Quem@PUC. Both applications access a single NoSQL database (a TripleStore + RDF system) that integrates academic and research information from different sources. Quem@PUC aims to answer questions like *Which researchers are related to subject Q?*. This relationship could be established through a research publication, a research project, or even an academic discipline from a graduate course. Busc@NIMA has the same objective in a narrower scope since the subject of interest Q belongs to the Environment domain, which is interdisciplinary. Through those tools available on the Web, it is possible to publicize research activities from the PUC-Rio community to be used by society in general, considering the particular interest of other researchers and journalists. Search results include professor's name, academic papers, teaching activities, contact links, keywords, and laboratories of those involved with the subject represented by the set of keywords input.

Busc@NIMA was a demand from Núcleo Interdisciplinar de Meio Ambiente (NIMA), a PUC-Rio center that encourages and develops projects and researches on issues related to the environment field. NIMA's coordinators wanted to facilitate and expand the public's access to the activities of professors and researchers in this field. Thus, the idea of the Busc@NIMA tool emerged to allow interested parties to identify the competencies in the environmental area existing at PUC-Rio, in its laboratories and departments, coordinated by members of the University's faculty and researchers.

Later on, we decided to develop Quem@PUC. Instead of focusing on the environmental field, we wanted to expand the possibilities of public access to do searches based on any topic of interest. This demand came naturally from the moment Busc@NIMA became publicly available. Therefore, Busc@NIMA is specialized in returning a list of professors/researchers who work on topics related to the environmental field, and Quem@PUC returns a list of all professors/researchers at PUC-Rio, as indicated in table I.

Table I. Busc@NIMA and Quem@PUC differences

Main difference between the tools		
Search Systems	Busc@NIMA	Quem@PUC
Main Goal	Identify members of the PUC-Rio community who work on **topics related to the environmental field**	Identify members of the PUC-Rio community who work with **any topic**
Search Question	Which researchers are related to topic Q from the environmental field?	Which researchers are related to topic Q?
Web URL	https://buscanima.biobd.inf.puc-rio.br/	https://quemnapuc.biobd.inf.puc-rio.br/

The remainder of this article is organized as follows: after this introduction and motivation, Section 2 presents some theoretical background on technical issues involved. Then, Section 3 describes our design solutions for the software development and Section 4 explains some of our tools's features. Related

works from UFMG and UFSC are described in section 5 . Section 6 concludes and points out some current and future works.

2. BACKGROUND

In most organizations, many information systems coexist and overlap in content and purpose, and at a university, like PUC-Rio, it is no different. In general, existing information systems were not designed to be integrated, and their corresponding databases should meet a specific context. We may build an information integration system (IIS) to provide uniform access to a set of heterogeneous and autonomous information sources and extract insights from integrated information. Such systems abstract the complexity of locating and accessing the information silos and resolving conflicts between sources. Several approaches have been proposed, emphasizing solutions to structural and technical problems [Ziegler and Dittrich 2007].

IIS, where data from multiple sources and formats are stored in a single central repository, requires extraction, transformation, and load (ETL) processes. Tools specialized in creating, maintaining, and monitoring automated ETL processes do not require programming skills. However, the semantic information integration must guarantee that they will preserve those data interpretation aspects through the integration process, and only related data may be combined. Such requirement demands domain knowledge to discover, understand and represent information.

Ontologies define controlled vocabularies that uniquely identify a set of concepts to avoid ambiguity in their interpretation. Therefore, an ontology can also be seen as a data model that formally defines the relationships between concepts. Ontologies aim to establish an organized and standardized relationship between entities, enabling data contextualization extracted from different sources and their interpretation.

RDF (Resource Description Framework) is a W3C graph data model that allows data to be shared, reused, and described by ontologies. It has features that facilitate data integration and interoperability. Data and metadata can be represented through a triple schema (subject, predicate, object) and stored in a TripleStore database. A TripleStore is a NoSQL graph database specialized in storing and manipulating RDF data through the SPARQL language. As a NoSQL data store, it also allows a flexible schema definition.

Information Retrieval Systems are designed to find digital objects stored in extensive collections that satisfy user information needs [Manning et al. 2008]. The query is usually specified in natural language through keywords representing the search intention. The search engine translates this query in its language model to return a list of digital objects. These digital objects can be documents, tables, graphics, videos, or images. Still, according to the object collection purpose and organization in more specific contexts, they may correspond to other resources like triples or subgraphs.

3. SYSTEM DESIGN AND TECHNOLOGIES

3.1 From Data Sources to Databases

Considering the system's purpose and the possibility of adding new data sources in the future, we have initially identified two non-functional requirements that guided the system architecture design: (1) a schemaless data model, and (2) the support for loading files in different formats, such as XML, RDF, CSV, and Excel spreadsheets. There is a variety of data sources and significant difficulty to establish a schematic model *a priori*. A data model without a rigid schema allows data to vary in structure. Thus, we decided to store the integrated data in a NoSQL system with a flexible schema.

We have chosen RDF as the target data model for the ETL process. Standardized, known, and publicly available domain ontologies were used during the data integration transformation step to preserve

the semantics of data extracted from different sources and facilitate data interpretation by the search tool. Resources (subjects and objects) must correspond to concepts present in the used ontologies and predicates to relationships and attributes. For example, we used *BIBO: Bibliographic Ontology Specification*[1], *BIO: A vocabulary for biographicalinformation*[2], *VIVO: An ontology for representing scholarship*[3], and *CCSO: Curriculum Course Syllabus Ontology*[4].

The Lattes platform, maintained by CNPq, is responsible for storing and making available Lattes Curriculum data of Brazilian researchers. CNPq provides a XML file for each Lattes Curriculum, as shown in figure 1. To help institutions to integrate data from the Lattes Platform into their internal information systems, CNPq provides Lattes Extractor[5]. Using this service, registered institutions can batch download the Lattes Curriculum of each researcher as an XML file. At PUC-Rio, the CCPA (Central Planning and Evaluation Coordination) is responsible for this service and provides monthly the set of files to feed our system's database.

According to the XML schema file (XSD)[6] that describes these XML files, there are 5 elements in the first level subordinate to the root node CURRICULUM VITAE: General Data (DADOS GERAIS), Complementary Data (DADOS COMPLEMENTARES), Bibliographic Production (PRODUÇÃO BIBLIOGRÁFICA), Technical Production (PRODUÇÃO TÉCNICA) and Other Productions (OUTRAS PRODUÇÕES). An XSLT script is employed to transform each XML file into RDF using the selected ontologies as shown in figure 2. General speaking, XSLT (eXtensible Stylesheet Language Transformation) stands for a transformation of XML documents into other formats (like RDF, HTML, CSV, ...) using a styling language for XML. But this transformation also considers each element's semantics and attributes to map to the correspondent one from the selected ontologies. For example, table II shows some mappings of attributes.

CCPA also provides regularly dump of the latest administrative information as a csv file, another data source we use. The administrative informations are the Lattes ID, an administrative registration unique number, name (foaf:name), mail box (foaf:mbox), and home page (foaf:homepage) of each employee of PUC-Rio. As new dump is available, the repository is cleaned up, transformation is executed and data is loaded into the repository.

To convert administrative information to our RDF model, we use an ETL tool called Linked Pipes[7][Klímek and Škoda 2017]. It is a lightweight tool specialized in triplifying, i.e., the process of transforming data from any structured or semi-structured format (relational, XML, CSV) into a triple format from the RDF graph model and RDF manipulation using SPARQL. Linked Pipes has several built-in components, pipeline creation is carried out through a graphical interface, it does not require knowledge of low-level programming languages, make use of metadata to configure its components and allow execution and monitoring of processes history. It can be installed on separate servers from the application and database since it allows connection to remote repositories, as a source or destination of data. Also, in order to join Lattes data and administrative data, the ETL process uses the Lattes ID of each researcher as a unique key. This pipeline use the "Tabular" component available in Linked Pipes. It is a CSV to RDF converter that follows the standard of the W3C: Generating RDF from Tabular Data on the Web[8].

Another data source included in our database are the courses offered at PUC-Rio. Through the

[1] http://purl.org/ontology/bibo/
[2] http://purl.org/vocab/bio/0.1/
[3] https://duraspace.org/vivo/
[4] https://w3id.org/ccso/ccso#
[5] http://memoria.cnpq.br/web/portal-lattes/extracoes-de-dados
[6] http://memoria.cnpq.br/c/document_library/get_file?uuid=772309c0-fb72-4c6a-8c88-64b0ba46ae5d&groupId=313759
[7] https://etl.linkedpipes.com/
[8] https://www.w3.org/TR/csv2rdf/

Table II. Lattes attributes mappings

Lattes attributes	Ontology's predicates
NOME-COMPLETO-DO-AUTOR	foaf:name
E-MAIL	foaf:mbox
HOME-PAGE	foaf:homepage
NRO-ID-CNPQ	foaf:identifier
NOME-PARA-CITACAO	foaf:citationName
ANO-DO-TRABALHO ANO-DO-TEXTO ANO-DO-ARTIGO DADOS-BASICOS-DO-LIVRO/@ANO DADOS-BASICOS-DO-CAPITULO/@ANO	dcterms:issued
NOME-DA-EDITORA	dcterms:publisher
TITULO-DO-TRABALHO TITULO-DOS-ANAIS-OU-PROCEEDINGS TITULO-DO-PERIODICO-OU-REVISTA TITULO-DO-JORNAL-OU-REVISTA TITULO-DO-TEXTO TITULO-DO-ARTIGO-INGLES TITULO-DO-LIVRO	dc:title
PAGINA-INICIAL	bibo:pageStart
PAGINA-FINAL	bibo:pageEnd
ISSN	bibo:issn
NUMERO-IDENTIFICADOR	bibo:identifier
ISBN	bibo:isbn
DOI	bibo:doi
TEXTO-RESUMO-CV-RH TEXTO-RESUMO-CV-RHEN	bio:biography

```
<ARTIGO-PUBLICADO SEQUENCIA-PRODUCAO="154" ORDEM-IMPORTANCIA="">
    <DADOS-BASICOS-DO-ARTIGO NATUREZA="COMPLETO" TITULO-DO-ARTIGO="Nomenclatural novelties in
    Miconieae (Melastomataceae): new synonym and typifications" ANO-DO-ARTIGO="2020"
    PAIS-DE-PUBLICACAO="" IDIOMA="Português" MEIO-DE-DIVULGACAO="MEIO_DIGITAL"
    HOME-PAGE-DO-TRABALHO="[doi:10.11646/phytotaxa.443.2.5]" FLAG-RELEVANCIA="NAO" DOI="10.11646/
    phytotaxa.443.2.5" TITULO-DO-ARTIGO-INGLES="" FLAG-DIVULGACAO-CIENTIFICA="NAO"/>
    <DETALHAMENTO-DO-ARTIGO TITULO-DO-PERIODICO-OU-REVISTA="Phytotaxa (on-line)" ISSN="11793163"
    VOLUME="443" FASCICULO="" SERIE="2" PAGINA-INICIAL="179" PAGINA-FINAL="188"
    LOCAL-DE-PUBLICACAO=""/>
</ARTIGO-PUBLICADO>
```

Fig. 1. An XML file of a given Lattes Curriculum

website Micro Horário `https://fliplink.io/XN5jR`, it is possible to download the courses available in the current semester. The file downloaded is in CSV format. The courses informations are discipline code (ccso:code), name (ccso:csName) and professor's name (ccso:hasInstructor). To convert the data to our RDF model, we also use Linked Pipes.

To complement courses' data, we perform web scrapping on PUC-Rio's page to collect the knowledge body of the disciplines. A Python script was developed, loading all disciplines codes, and acessing *https://www.puc-rio.br/ferramentas/ementas/ementa.aspx?cd=XXXX* (where *XXXX* is the discipline code) using the library Requests. Afterwards, utilizing the library BeautifulSoup, the script navigates through the HTML pages until it finds the knowledge body of the disciplines. Considering accessing each webpage to collect the knowledge body and finally storing the data into AllegroGraph, the execution time of the script is around an hour. When modelling the triples for disciplines, the following were used: ccso:KnowledgeBody for knowledge bodies and ccso:code for the discipline codes.

Until recently, those three data sources (Lattes Curriculum, administrative information and dis-

```
<rdf:Description rdf:about="#P154">
  <rdf:type rdf:resource="http://purl.org/ontology/bibo/Article"/>
  <dc:title>Nomenclatural novelties in Miconieae (Melastomataceae): new synonym and typifications</dc:title>
  <dcterms:issued>2020</dcterms:issued>
  <dc:language>Português</dc:language>
  <foaf:homepage>[doi:10.11646/phytotaxa.443.2.5]</foaf:homepage>
  <bibo:doi>10.11646/phytotaxa.443.2.5</bibo:doi>
  <dcterms:isPartOf>
    <rdf:Description>
      <rdf:type rdf:resource="http://purl.org/ontology/bibo/Journal"/>
      <bibo:issn>11793163</bibo:issn>
      <dc:title>Phytotaxa (on-line)</dc:title>
    </rdf:Description>
  </dcterms:isPartOf>
  <bibo:pageStart>179</bibo:pageStart>
  <bibo:pageEnd>188</bibo:pageEnd>
  <bibo:volume>443</bibo:volume>
</rdf:Description>
```

Fig. 2. An example of an XML file for a Lattes Curriculum Vitae converted to RDF with the help of ontologies

ciplines) had proven to be sufficient for the system's purpose. However, a demand for information beyond those was soon noticed. To solve this, we built an "User Area" to enrich the database, which allows a professor/researcher to enter two new pieces of information: keywords and laboratories. The first consists of search terms that the professor/researcher would also like to be recognized for, complementing the information already stored, that can be entered into the system directly by them. The modelling used for keywords utilizes the predicate SIO:000001 ("is related to"). The second also consists of search terms involving R&D laboratories related or coordinated by the professor, such as, for example, equipment, service provision and other technical terms. Thereby, the laboratory triples were modeled so that they contemplate the name of the laboratory, it's URL, and associated information, as mentioned above.

AllegroGraph was used as the database system for centralized storage of data originating from other systems. It is a multi-model NoSQL database (Document in JSON, JSON-LD, and Graph in RDF). We have chosen AllegroGraph because it allows the most significant number of triples (five million) per repository, compared to other available options in a free version. Due to this limitation and also to facilitate periodically load process, triples were separated in five repositories: two for CV Lattes, one for administrative information, one for disciplines, other for syllabus and one for researcher's inputted data.

In order to optimize triple retrieval, AllegroGraph automatically performs the creation of seven indices. Each index uses a sort order of triple elements, for example, the index spogi classifies the subject(s) first, then the predicate, object, graph, and finally the id. The initial set of repository's indexes is: spogi, posgi, ospgi, gspoi, gposi, gospi and i. It is possible to create or remove indexes, for example, if the database is not subdivided into datasets (graphs), indexes starting with "g" will never be used and can be deleted. Eliminating unnecessary indexes speeds up data load. In addition, the database supports native free text (literals) indexing on triple objects. It is necessary to create *freetext* indexes where it is possible to specify which predicates should be considered, the stopwords to be removed, and the insensitive accent configuration.

Complex graph patterns (CGP) [Angles et al. 2018] queries were built using SPARQL language to retrieve resources of interest about researchers/professors. Keyword matching in queries was performed using the (*magic property*) *fti:match* operator, which allows the query to use previous created *freetext* index. Due to repository size limitations, it was necessary to use a federation mechanism to execute the queries in order to return a result that involves all related repositories created in the database. This feature allows AllegroGraph to automatically distribute SPARQL queries among the target repositories, local or external ones, and combine the results transparently for the application.

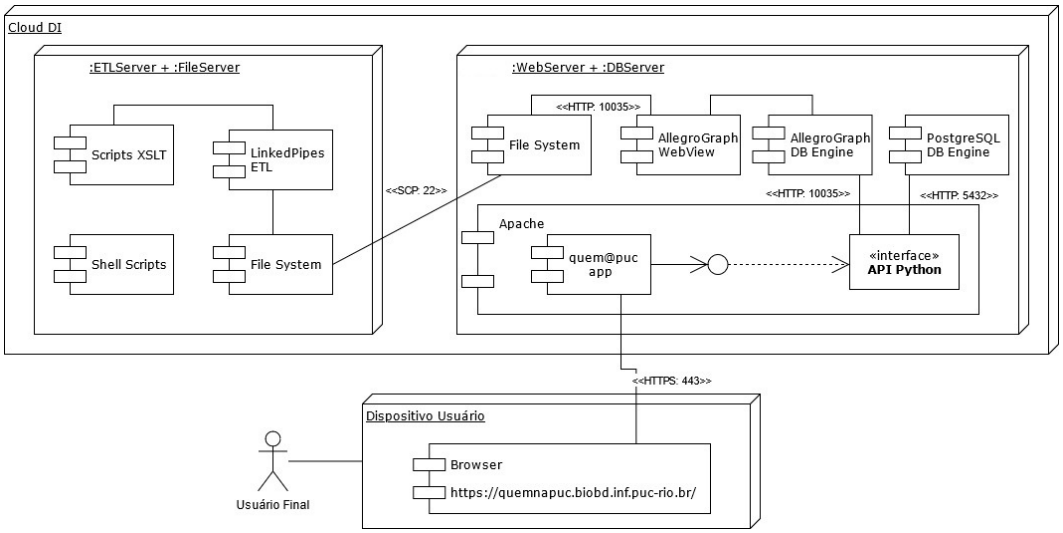

Fig. 3. UML Deployment Diagram

Additionally, PostgreSQL DBMS was also used to record search activity to analyze user behavior and search terms, as well as store professor's/researcher's information provided by PUC, such as Lattes identifiers and e-mails. The log data allows statistics generation such as the most searched terms, most selected researchers/professors, etc. Also, in case of eventual problems, the data can provide information for quick fixes. Since log data, Lattes identifiers and e-mails are well defined and not subject to constant changes in the schema, we decided to adopt a Relational DBMS to meet this requirement.

3.2 System Architecture

The application was built using Python programming language in conjunction with the Flask microframework, allowing fast prototyping of web applications. The AllegroGraph Python API provides methods to create, query, and maintain RDF data and manage stored triples.

The entire system is divided into two servers: one containing the ETL tool (ETL Server), where all ETL pipelines are done, and the other is the application server (Web Server + DB Server), where the databases are hosted, and the search engine runs, managed with the Apache tool (see figure 3).

To use the User Area as shown in figure 4, first the professor/researcher has to input the Lattes ID and the system sends an email message to the mailbox associated with this ID according to the administrative data source. The researchers register themselves and define a password for future authentication. Afterwards, the systems allows them to input new information that is indexed and incorporate in the search.

The source code of Busc@NIMA and Quem@PUC are identical, except for one environment variable ("MODO_NIMA"), responsible for making the system decide if it should execute in mode Quem@PUC, or Busc@NIMA. For example, in case the environment variable is present, the system utilizes three different HTML files, and performs two distinct SPARQL queries. In table III, we present a piece from the HTML file containing the initial page, and shows how it is possible to change the title of the website utilizing the functionality from Jinja2, included in Flask.

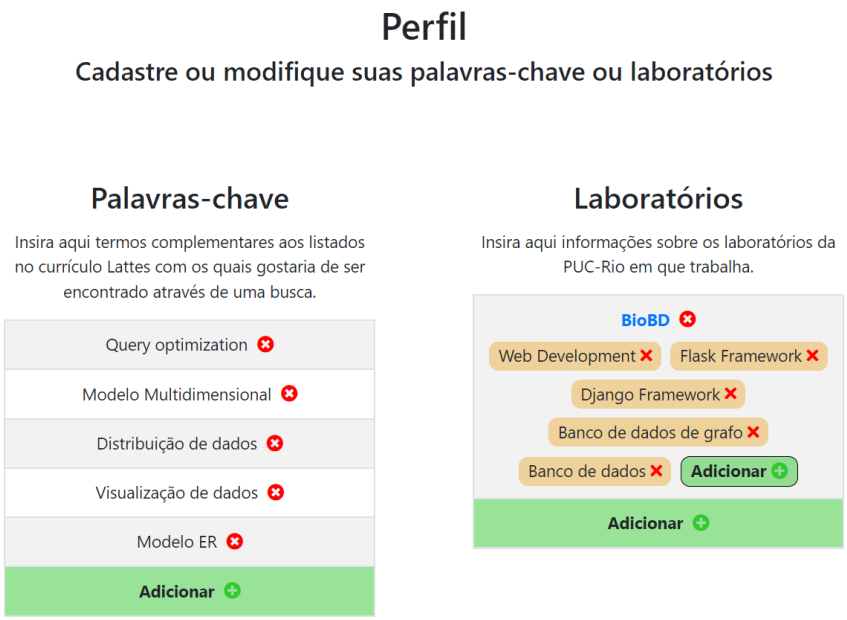

Fig. 4. User Area: keyword and laboratories

Table III. Environment variable for Quem@PUC and Busc@NIMA

```
{% if modo_nima %}
<h1 class="titulo azul-marinho">Busc<i class="fas fa-search" style="font-size: 0.6em;"></i>NIMA</h1>
{% else %}
<h1 class="titulo azul-marinho"><i class="fas fa-search" style="font-size: 0.85em;"></i>uem@PUC</h1>
{% endif %}
```

3.3 Classification using Machine Learning

Besides the environment variable mentioned, what differentiates Quem@PUC and Busc@NIMA is the fact that Busc@NIMA is specialized in retrieving information in the environmental field. To classify the environmental field data, Machine Learning techniques were used, more specifically, a deep neural network. Firstly, manual classification of a large number of academic papers was made, that is, a large number of productions were collected, and one by one each production were individually classified, saying whether it is related to the environmental field or not. To classify manually, the titles of each production were collected, and three individuals manually classified if it was related to the environmental topics, if it was not, or if it was not possible to know from the title. If two individuals agreed on the same classification, this production and classification were stored, if not, this production was discarded.

After acquiring a database of approximately 1200 productions manually classified, text processing was performed in the titles, utilizing the functionality CountVectorizer from Python's library scikit-learn. This processing transforms a text into a sparse matrix/array. Afterwards, this matrix/array is utilised to train a deep neural network, containing a deep layer with ten nodes, and an exit node. 80% of the productions were used to train the neural network, and the remainder to evaluate the accuracy of the neural network, which was around 86% accuracy. In the future, this neural network will be used

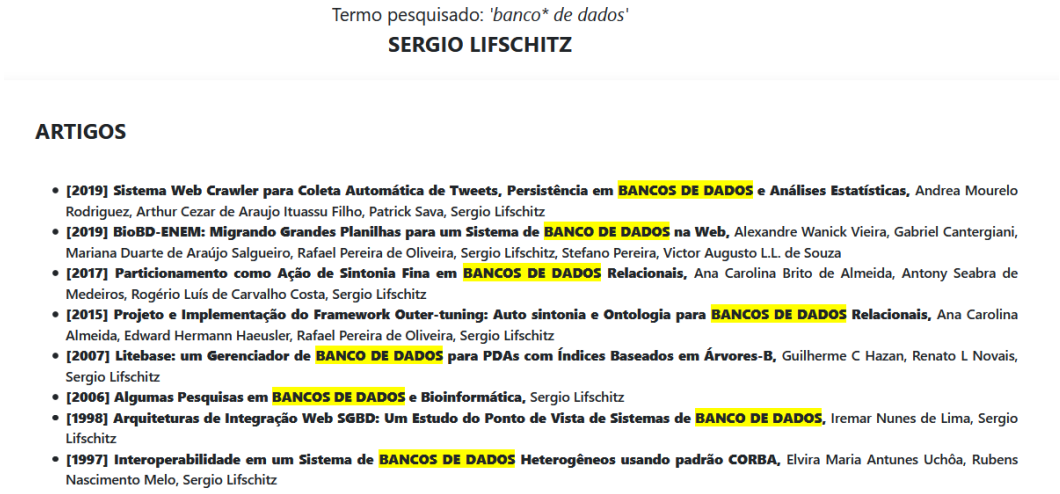

Fig. 5. Result of use of wildcards

to classify all productions in AllegroGraph's database, in order to display in the Busc@NIMA system only the productions that it claims to be related to the environment with a high certainty rate.

4. SYSTEM FEATURES

The system retrieves information given any list of keywords Q. The system shows results for exact matches given the input keywords. It returns items associated with professors/researchers that contain precisely the word(s) informed. In this pattern match operation, accented or uppercase characters are considered equivalent to unaccented or lowercase characters. The system performs an exact match with all the words input, in any order, that is, the logical operator to perform the matching is *AND*. Depending on how generic the keyword input is, it may imply a larger system's response time. We rank [Baeza-Yates and Ribeiro-Neto 1999] the results based on which professor/researcher has the most matches with the keyword input.

For example, if the user enters the words "web" and "semântica" the match will identify publications such as "Semântica na Web: Uma Estratégia para o Alinhamento Taxônomico de Ontologias" and "Web Semântica: O Futuro da Internet" since both words occur in the titles, even if in a different order from the one informed.

If the user searches for the word "catador", for example, the system will perform the correspondence with elements that contain the words "catador" (singular), such as the publication whose title is "Construindo identidades: catador herói ou sobrevivente da perversa forma de catação", but will not match with titles of publications that contain the word "catadores" (plural). The system also allows an approximate search variation concerning the spelling of the words used. In this case, the user can make use of two *wildcards* character options. The question mark (?) matches any single character, and the asterisk (*) matches none to many characters at the associated position. In figure 5, we present an example wildcard in the expression *banco* de dados*.

Another feature included, due to the fact that many publications are international and have english tittle, is searched term(s) translation to English as shown in figure 6. The system suggests an English translation for a Portuguese keyword. If the user accepts the suggestion, the query will retrieve results merging both Portuguese and English items. To perform language detection and term translation, Google's Cloud Translate is used.

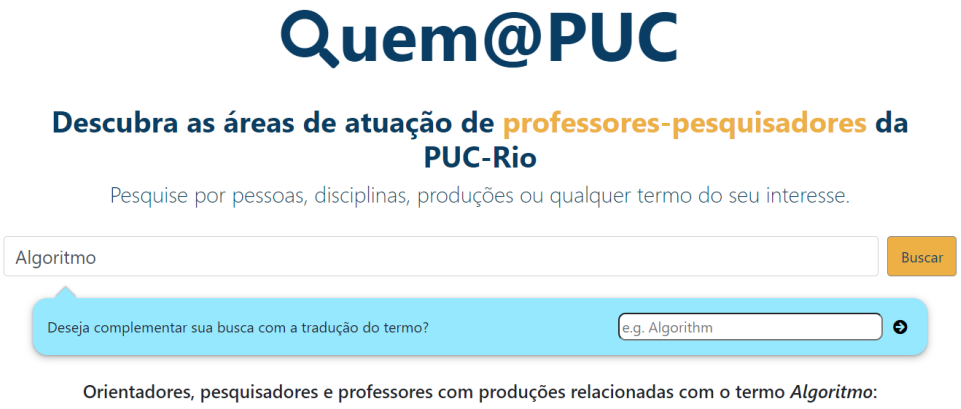

Fig. 6. Translation functionality

This service provides an API in Python to do a fast and efficient translation or detection. It was chosen because of its high quality, and no cost for up to 300000 uses per month. As each term uses up to 3 times the API, there is virtually no API expense. To suggest the translated term, two different strategies are used. The first strategy consists of utilizing the detect language function from Google's API. In case the API identifies English, it utilizes the API a second time to translate it to Portuguese, and the output is the suggested translated term.

However, if the detected language is not English nor Portuguese, or the certainty of the API is too low ($< 30\%$), the second strategy comes into action. The second strategy is similar to a brute force method. The system utilizes the API two times: one to translate the user's term from English to Portuguese, and another for translating it from Portuguese to English. Then, if and only if one of the attempted translations was successful, the output term is suggested. If both translations were successful, it means the term is ambiguous, and the suggested translation is probably incorrect. Therefore, the system does not suggest any translation. The same occurs when none of the translations were successful (for a translation to be successful, the source term and the translated term have to be different. For example, ["Algoritmo" –> "Algorithm"] is a successful translation, but ["Allegro" –> "Allegro"] is not).

After selecting a professor/researcher, it is possible to read their biography, access their contact page, their Lattes Curriculum, the productions and lectures, and the keywords and laboratories, if any as shown in figure 7. When selecting any production categories, the items containing the searched word appear first, and then the articles released in the years 2018, 2019, and 2020 appear ordered in reverse chronological order.

In addition, the system enables to search for a professor/researcher name directly, in which the user can enter the full name or part of it. We were able to develop a way to understand when the user is searching for a keyword or the professor/researcher's name, bringing more semantics to the project as presented in figure 8. According to the triple predicate, the SPARQL query returns if the keyword matches with a person name or a publication title.

Fig. 7. Selection of professor/researcher

Fig. 8. Result of desambiguation

5. RELATED WORKS

Somos UFMG[9] was developed to facilitate UFMG competencies identification. This mapping is important to boost the interaction of scientific and technological research areas with the general public and public and private institutions. The tool makes it possible to retrieve the researchers, their expertise, and scientific and intellectual production. The system also retrieves data on Units, Departments, and laboratory infrastructure for the university and presents an indicators panel about

[9] http://somos.ufmg.br/

the faculty and their literary production. Most of the information about the researchers is extracted from the Lattes Platform.

When typing a keyword referring to a subject of interest, the system presents the result grouped according to the type of the corresponding element. For example, when typing the word *Sustainability*, the results are separated into two sets: Keywords and Specialty Hierarchy, both associated directly with the researchers. In another example, using the word *Nursing*, the result was grouped into Keywords, Specialty Hierarchy, Academic Units, and Departments. It seems that the system uses a syntactic match approach on literal values that matches the element name, be it a person or a department or any other type of element.

The differences between Somos UFMG and Quem@PUC are: Quem@PUC retrieves the courses taught by professors in the current semester, as well as their knowledge bodies; professors/researchers can enter keywords that refer to them and information about their laboratories through a private area; possibility of complementing the search with the term translated from Portuguese to English or vice-versa; and the productions retrieved are classified not only into articles but also into biographies, books and chapters.

Another example is the IPU[10] which was developed to enable query of data from Lattes curricula registered by UFSC members. Data extraction is performed every fifteen days. Based on this dataset, the system allows for the generation of graphs that present the leading indicators of the intellectual production of its researchers, in addition to the search for specific data found in CV Lattes. In addition, the system has the functionality of downloading data as files in Excel-compatible format for other uses. However, the system use is restricted to the UFSC community through login and password registration and because of that we are not able to check the system features.

6. CONCLUSIONS

The publication of this tool represents an efficiency gain in identifying professors or researchers from the PUC-Rio community. However, we cannot deal with the ambiguity of words. Most systems usually focus on the terms involved in the searches (syntax) and not on their meanings (semantics). To overcome this limitation, it will be necessary to better incorporate semantic elements in the search engine to understand the users' intentions behind their searches. Among the possibilities to carry out this incorporation, there is a controlled list of keywords, thesaurus, domain ontologies, language models, word embeddings, and even knowledge graphs that are being studied at the present moment.

In future versions, it is also planned to extract more information from CVs, such as research and development projects and lines of research. Other data sources can be incorporated, like the researcher's Web pages. The machine learning model is yet to be improved.

REFERENCES

Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., and Vrgoč, D. Foundations of Modern Query Languages for Graph Databases. *ACM Computing Surveys* 50 (5): 1–40, sep, 2018.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

Husain, O., Salim, N., Alias, R. A., Abdelsalam, S., and Hassan, A. Expert finding systems: A systematic review. *Applied Sciences* 9 (20), 2019.

Klímek, J. and Škoda, P. Linkedpipes etl in use: Practical publication and consumption of linked data. In *Proceedings of the 19th International Conference on Information Integration and Web-Based Applications & Services*. iiWAS '17. Association for Computing Machinery, New York, NY, USA, pp. 441–445, 2017.

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[10] `http://ipu.sistemas.ufsc.br/`

Salgueiro, M., dos Santos, V., Rêgo, A., Guimarães, D., Haeusler, E., dos Santos, J., Villas, M., and Lifschitz, S. Quem@puc - a tool to find researchers at puc-rio. In *Anais Estendidos do XXXVI Simpósio Brasileiro de Bancos de Dados*. SBC, Porto Alegre, RS, Brasil, pp. 93–98, 2021.

Ziegler, P. and Dittrich, K. R. Data integration – problems, approaches, and perspectives. In *Conceptual Modelling in Information Systems Engineering*, J. Krogstie, A. L. Opdahl, and S. Brinkkemper (Eds.). Springer, pp. 39–58, 2007.