Architecture to detect patterns from bibliographical data sources

Nombre del Primer Autor[1], *Nombre del Segundo Autor*[2], *Nombre del Tercer Autor*[1,3]

[1]Afiliación del primer autor, nombre de la universidad,
Dirección de la Universidad, ciudad, pais, código postal

[2]Afiliación del segundo autor, nombre de la universidad,
Dirección de la Universidad, ciudad, pais, código postal

[3]Afiliación del tercer autor, nombre de la universidad,
Dirección de la Universidad, ciudad, pais, código postal

Corresponding author: {primero,segundo}@universidad.edu, tercero@universidad2.edu

Fecha de recepción: 21 de Septiembre, 2014
Fecha de aceptación: 17 de Octubre, 2014

ABSTRACT

Increasingly, the use of scientific publications online is occurring more frequently. There is an extremely large number of scientific publications on the web. For researchers it is a challenging topic to pursue. Finding peers interested in collaborating on a certain topic or reviewing literature is challenging. We propose, a novel architecture to join multiple bibliography sources to detect patterns, enriching a data model using ontologies, vocabularies and Linked Data technologies. In our prototype implementation we use the components of this architecture to have a central repository with bibliographic resources and find similar knowledge areas ins the domain of Ecuadorian researchers.

Keywords: *Linked Data, KDD, Ecuador.*

1. INTRODUCTION

The number of publications that we can access almost instantaneously is rapidly increasing through using online resources such as search engines and digital libraries. This makes it more challenging for researchers to pursue a topic, review literature, track research history because the amount of information obtained is too extensive unless the user knows the name of the exact research paper that a researcher is looking for.

Currently, certain information about researchers and their bibliographic resources are scattered among various digital repositories, text files or bibliographic databases. When you need to propose projects with several researchers in a specific area belonging to different Institutions of Higher Education, raises questions such as: Who it works in similar lines of research? Or, how can you create a network of researchers in a common area when we do not know if they exist? In addition, defining the profile of a person in analysis, get their articles, know which one are the magazines that were accepted, among others, it is obligatory to access to multiple data sources. Given that, it is known that this process is manual, syntactic and different for each source of bibliographic resource available on the Web.

The expansion of this knowledge base will allow our academic community to have a centralized digital repository which has information of Ecuadorian researchers based in bibliographic resources. This project aims to encourage interagency collaboration and obtain as a result of this work a validated semantic repository, to locate researchers working in similar research areas and provide updated information accessible and reusable. Enhancing the generation of research networks with academic peers in the region it could provide a greater opportunity for cooperation and collaboration between the participating institutions.

The rest of this paper is organized in the following way: section 2 presents some related work. We outline the architecture proposed in section 3. In section 4 we give an initial prototype implementation using the components of the architecture to detect similar areas in the domain of Ecuadorian Researchers. Conclusions and future work are in section 5.

2. RELATED WORK

Previous studies on geoscience have shown that is posible to improve data retrieval, reuse, and integrate data repositories through the use of ontologies. According to [Krisnadhi et al. 2015], the Geolink project a part of EarthCube[1] that integrates seven repositories using Ontology Design Patterns (ODPs) [Gangemi 2005] defined manually. They have a set of ODPs as the overall scheme, rather than a monolithic ontology is used. To obtain data they executed federated queries. Conversely, in our proposal all sources form a single repository and we do not use federated queries because the response time is interminable. The data model Geolink is defined specifically for geodata, which it differs from our proposal covering several domains according to the bibliographic source.

Besides, Semantic Scholar is a project that have access to sources such as Arxiv, DBLP and CiteSeer where they take publications in pdf format to extract figures, tables, and captions, as [Bizer et al. 2009] explains. Subsequently in [Clark and Divvala 2015] work they process citations to find relationated papers, similar areas, similarity between abstracts. They are trying to cover the domain of Computer Science and offer a search engine to ease the review of literature. In contrast, we get a bibliographic resource which is enriched with data from several sources using ontologies and Linked Data Technologies.

The last similar work that we have found at the time of writing is [Alfraidi 2015] which retrieves publications from Google Schoolar to find relationships based in the citations or references which gives an insight of the hierarchy distribution of publications around a given topic. The relationships are visualized in a 2D graph where nodes represent documents and links represent the citations or references between them. Although the user could see the papers connected in a topic or the most leading papers in the area. They cannot make a network of researchers that work in a certain area. As well some papers could be banned using just a source and there are research areas with a big amount of publications which can be difficult to visualize in a 2D graph.

3. GENERIC ARCHITECTURE

This section presents briefly the architecture proposed to use academic literature available on the web and find relationship between authors and their publications. Our approach relies on three different modules, namely: Data Extraction, Data Enrichment and Pattern Detection. The high-level modules of the architecture are illustrated in Figure 1 and their features will be explained along this section.

- We encompass different bibliographic *Data Sources*, which are heterogeneous. The access in some of this sources it is limited due to the restrictions of payment, subscriptions

[1] EarthCube is a community-led cyberinfrastructure initiative for the geosciences; http://earthcube.org/

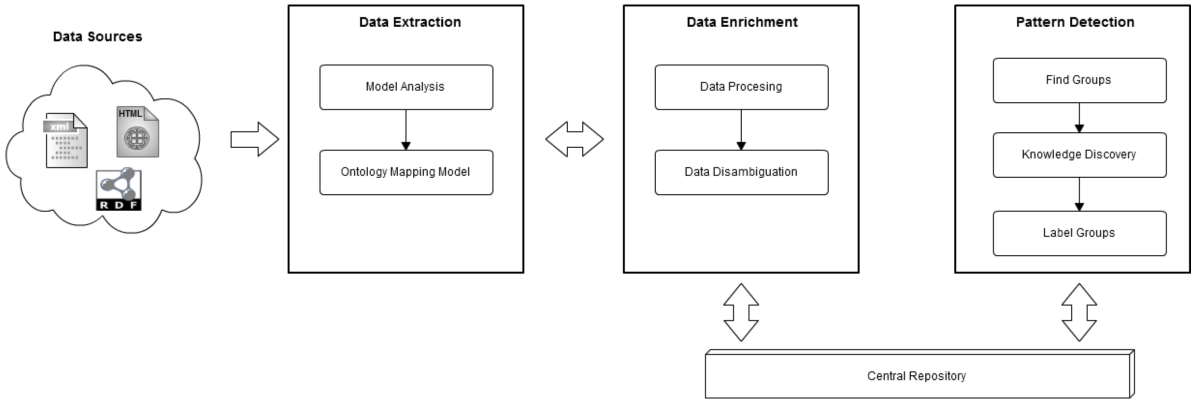

Figure 1. General architecture to detect patterns from bibliographic data sources.

and quota violations. We could get some descriptive information of academic literature such as title, abstract or keywords.

- The *Data Extraction* module manage heterogeneous data sources and extract it. This data is described using a bibliographic ontology and the *Model Analysis* component analyze the structure of each model. After that, the data described is mapped in the *Ontology Mapping Model* component with a common model.
- The *Data Enrichment* is the core of the system because in this module the data is processed and stored in the *Central Repository*. The main goal is to feed the *Central Repository* with information that runs through *Data Processing* component which put together all information extracted. After that, in the *Data Disambiguation* component the data is processed to remove inconsistent information.
- The *Pattern Detection* is the module which detect patterns from the data collected. Firstly, the whole data is taken from the *Central Repository* that is the input for the *Find Groups* component to detect some kind of association in the data set and group it. The *Knowledge Discovery* component it is used to extract knowledge from the associated groups. To speed up queries and have organized groups, each group is labeled in the *Label Groups* component. Finally, results are stored in the *Centralized Repository* for further queries.

4. PROTOTYPE IMPLEMENTATION

In this section, we describe the detailed aspects of our implementation. The modules described above have been used to join different bibliographical sources and detect similar knowledge areas in the domain of Ecuadorian researchers.

4.1. Data Sources

The different data sources represent repositories that contains information about authors and scientific publications of different areas. The sources about authors are distributed in different Institutions of Higher Education which can be accessed through DSpace[2]. The scientific publications are extracted from bibliographic sources such as Microsoft Academics, Google Scholar, DBLP and Scopus that make available their data via APIs, web pages or files. The data vary in their content either because each source has a different structure or the access to data is restricted. For example, in the case of Scopus, we can make a maximum of 5000 querys by IP,

[2] DSpace is the software of choice for academic, non-profit, and commercial organizations building open digital repositories; http://www.dspace.org

otherwise the source blocks access for seven days. Scopus has the following characteristics: data affiliation of authors, tables, graphs of publications, authors study areas, etc. While DBLP or Microsoft academics do not have these properties. Therefore we see that it is necessary to make a unification of these variety of data models in a central repository that contains literature from different disciplines.

It is necessary to process the data sources referred above to understand the structure and access of the data, because each API is different. These tasks are described in detail in the next subsection.

4.2. Data Extraction.

The *data extraction* module is responsible for collecting and processing bibliographic data resources from the sources mentioned in the previous section (Microsoft Academics, Google Scholar, DBLP and Scopus). The data extracted are analyzed in order to define a structure using documentation of source or web scraping techniques. The data is described and mapped to a common model, using a bibliography ontology and Linked Data Techniques. We use the bibliographic data sources to cover different scenarios and find the main problems involved in making the extraction and enrichment of bibliographic resources. Every time a new source is added, you must perform an analysis of the source model and a mapping to a common data model, these two processes are abstracted into two components described below.

4.2.1. Model Analysis.

The different bibliographic sources provide their resources with a logical structure or data model different having the same type of information. Bibliographic resources are not ruled by a standard or comprehensive model encompassing all properties as authors, appointments, conferences, knowledge areas, etc. Some features such as DOI, ISBN, format bibliographic references of resources are ruled by International Standard Bibliographic Description (ISBD)[Barbarić 2014], ISO 690[3]. Functional Requirements for Bibliographic Records (FRBR) [O'Neill 2002] recommended a new approach to cataloging based on an entity-relationship model a bibliographic resource. However it is not enough if we need a common data model to facilitate the processing of scientific publications.

In Figure 2 you can see the data model between two bibliographic databases: Microsoft Academics and Springer Open Access API, that represent the diversity of models of data between bibliographic databases. This heterogeneity of models represents the challenge of integrating various sources. Therefore before adding a new data source to the architecture must perform an analysis of its structure with respect to models already being used for the purpose of assigning correspondences between new source model and model common data.

In some cases a source do not publish your model data, then we use works that describe an model, for example [Ley 2009] have described the model data of DBLP. Another form is make xml requests sending as parameters the author names to source for inquire into data structure. The result of this component is data with defined model, after is mapped to a common model that is described below.

[3] ISO standard for bibliographic referencing in documents of all sorts.

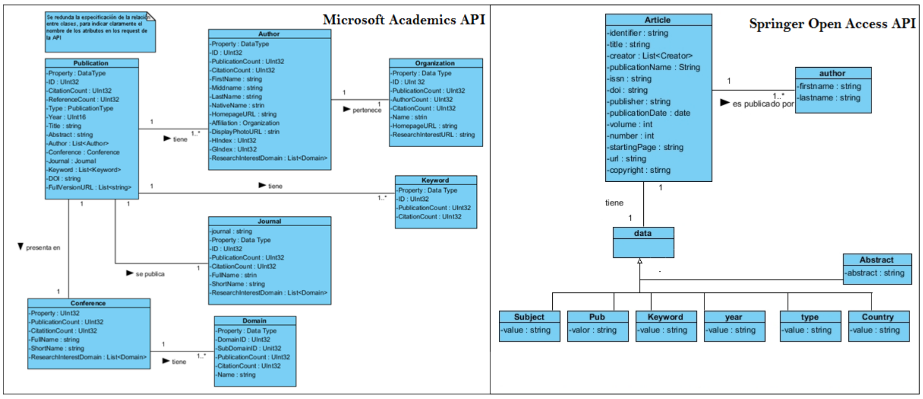

Figure 2. Data Models of Microsoft Academics API and Springer Open Access API.

4.2.2. Mapping Ontology Model.

In this component we have each data source with a different model that structures the data in a common model. This component find a correspondence between the properties of source model and an common data model is using techniques of *Metrics Similarity* [Charikar 2002]. This models are annotated using RDF[4] with an structure based in triples. This process of mapping is manual because the diversity of structures of source data models. The common model is illustrated in the Figure 3 in which you can observe the mapping between the data model of a source and common data model that we have defined. An alternative suggested by [Ortiz Vivar and Segarra Flores 2015] is that we can use to do automatic annotation of services that offer the bibliography databases to enhance the mapping process that now is manual.

The common model proposed is described using the ontology (BIBO)[Giasson and D'Arcus 2009], which is an ontology is used to describe bibliographic entities as books, magazines, etc. The authors are described using ontology FOAF (Friend of a Friend), it is an ontology used to describe people, their activities and relationships with other people and objects [Brickley and Miller 2012].

The data that we have has repeat entities and it is ambiguous to determine publications assigned to an author, in some cases author attributes are similar. So the data must be processed to be then stored, which is detailed below.

4.3. Data Enrichment

The module of *Data Enrichment* unifies all data of publications and authors in a central repository using ontologies. We find characteristics between publications and authors assignment correspondences through a component of *Data Process*. We have various entities of a same author or publication and this represent a problem of inconsistency, for this reason we have an component called *Data Disambiguation* that solve this problem which is described below.

We decide that is necessary to have materialized data authors and publications in a repository to find correspondences between these locally, other option is to recover the publications when a user needs them. The time between making a request to an external source and the mapping takes an average of twenty seconds. Therefore we have a unit repository to offer high availability and speed bibliographic resources to consult the publications of a specific author

[4] Resource Description Framework; https://www.w3.org/RDF/

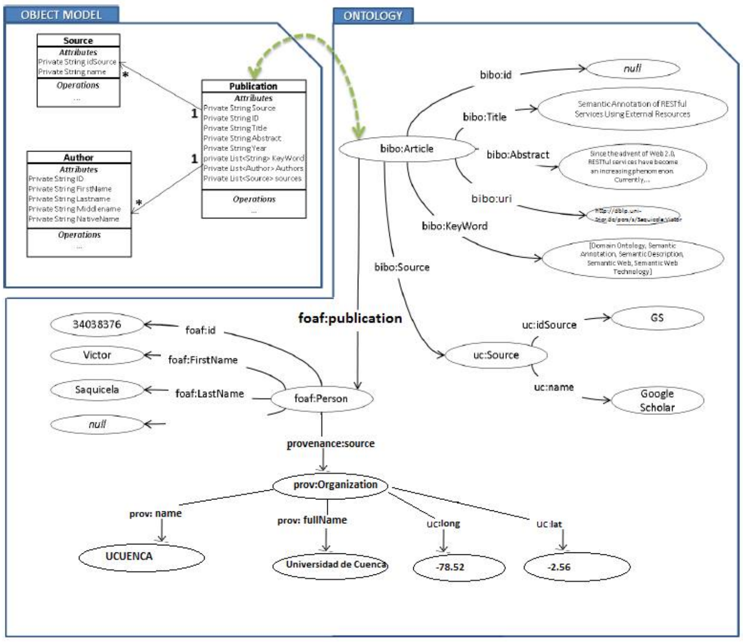

Figure 3. Common data model publications.

using SPARQL Endpoints[5].

4.4. Data Disambiguation

The *Data Extracted* module have repeated publications from different sources or publications assigned erroneously. It is necessary to discover authors that are the same entity between our various sources. The prototype developed allows define a single record of an author in a central repository using characteristics of the author and characteristics of their publications, taking advantage of ontological descriptions and algorithms similarity metrics. We have defined a priority among bibliographical sources according to the quality of the data. For example the most reliable source is Scopus, because it is consistent to searches such as *Juan Pablo Carvallo Vega* and *Juan Pablo Carvallo Ochoa*, identifying in most cases the difference. Other data sources such as DBLP not keep complete records of the author and only use the first name and lastname, causing scientific publications are assigned to other authors.

In the component *Data Disambiguation* handles the problem of data inconsistency, using the authors publications in our repository and publications that the source returns, if correspondences between them are states that there is consistency between authors and their publications, this process is illustrated in Figure 4 also this component helps us when we need to add new publications of an author. The work present in [Varadharajalu et al. 2011] have an algorithm to disambiguation using the affiliation, email, Url, Organization, Co-Authors. Making in context disambiguate authors and publications. In our case is necessary to have more information about the author to extend the disambiguation algorithm and can desert resources that don have a relation.

Until now we have a central repository using Ontologies and Linked Data but it is necessary to extract knowledge from this data. In the following section we show how the pattern detection module was applied to detect similar areas between researchers.

[5] Services that accept SPARQL queries and return results

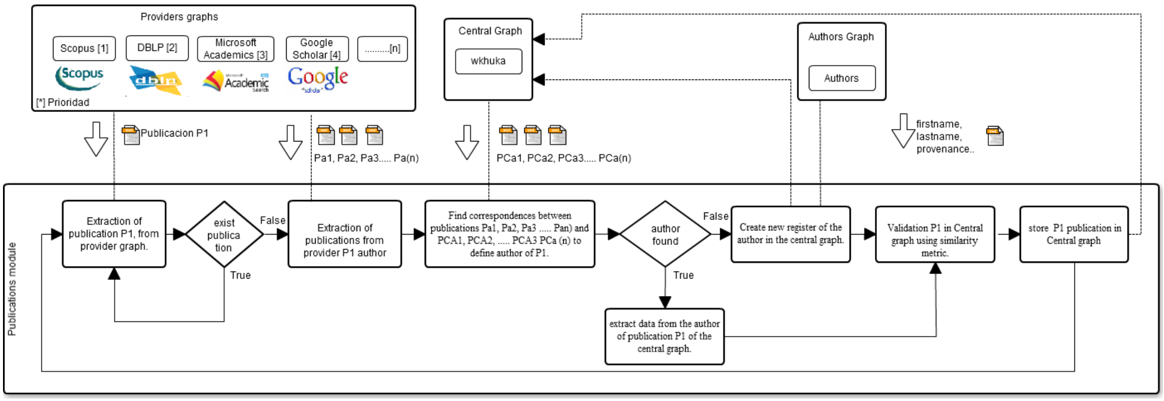
Providers graphs
Scopus [1]
DBLP [2]
Microsoft Academics [3]
Google Scholar [4]
..........[n]
[*] Prioridad
Central Graph
wkhuka
Authors Graph
Authors
Publicacion P1
Pa1, Pa2, Pa3..... Pa(n)
PCa1, PCa2, PCa3..... PCa(n)
firstname, lastname, provenance..
Publications module
Extraction of publication P1, from provider graph.
exist publica tion
False
True
Extraction of publications from provider P1 author
Find correspondences between publications Pa1, Pa2, Pa3 ..... Pan) and PCA1, PCA2, ... PCA3 PCa (n) to define author of P1.
author found
False
True
Create new register of the author in the central graph.
Validation P1 in Central graph using similarity metric.
store P1 publication in Central graph
extract data from the author of publication P1 of the central graph.

Figure 4. Disambiguation Process.

4.5. Pattern Detection

In this section, we outline the three components of the module to detect patterns in the data stored. It uses Apache Mahout to execute algorithms of machine learning. We choose mahout for the ability to deal with massive datasets, it is a scalable Java library and we could profit of the distributed computation, because it is built upon Apache Hadoop. The whole implementation is open sourced and available on our GitHub repository[6].

4.5.1. Find Groups

Broadly, keywords of academic literature talk about a certain topic area or methodology. Detecting similar areas based in the keywords . It could help us to detect researchers with interests in common and open up an opportunity to generate new research projects. Boosting interagency collaborative work and form cooperative research groups.

Firstly, we disjoin our data stored in the *Central Repository*, because we just need to process the keywords to group the most common areas. Other fields such as author or title of the publication are stored in a separate file. Both files are converted in a specific Hadoop file format that is SequenceFile[7]. Those files stores key/value pairs, where in the first file the key is a unique identifier and a bunch of keywords that belongs to a paper are stored as a value. Same happens in the second file with the difference in the value pair that stores the remaining fields such as author and publication.

We use techniques of text clustering [Andrews and Fox 2007] for the *find groups* module. Before to clustering the data into Mahout It is necessary to do some procedures . Data has been preprocessed to convert text in numerical values, but not all the keywords have the same relevance. The weighting technique used to magnify the most important words is Term frequency-inverse document frequency (TF-IDF). The weighted values are used to generate the Vector Space Model (VSM) where words are dimensions. The problem with this VSM generated is that words are entirely independent each other and it is not always true. Sometimes words have some kind of dependency such as *Semantic* with *Web*. In order to achieve this dependency we use collocations [Manning and Schütze 1999]. At the time of writing, we are executing our experiment using bi-grams and an Euclidean norm (2-norm), which can change. In future experiments, it will be interesting to generate vectors using Latent Semantic Indexing

[6] `https://github.com/cuent/kodar.git`

[7] Mahout also use Sequence files to manage input and outputs of MapReduce and store temporary files.

(LSI) or apply a log-likelihood to take words that mostly have the chance to go together. So in the long run, we have our vectors completed to start clustering.

We start with the vectors generated to execute K-Means algorithm in Mahout. It was executed using a Cosine distance measure as the similarity measure. RandomSeedGenerator[8] was used to seed the initial centroids. The experiment were set to 100 maximum of interations and the value of k varies according to the number of data extracted from the different bibliographic databases. Once the algorithm finishes we have our similar areas based in a bunch of keywords. The dilemma its how to detect researchers networks using the group of keywords generated, this work its done in the following section.

4.5.2. Knowledge Discovery

Once detected the groups of similar areas. We could extract some knowledge from the groups stablished such as what researchers could be interested to work together based in the areas they are working on.

We have developed a MapReduce model to accomplish it as you can observe in the Figure 5. First, we sort our vectors accord to a unique identifier. After that we merge each vector clustered with their original keyword. In our final job, we merge the resulting file of the first stage (Sort & Join) and the additional file containing the remaining fields (title, author). Then, we get a file with all the original fields, plus a field showing the cluster that each row belongs.

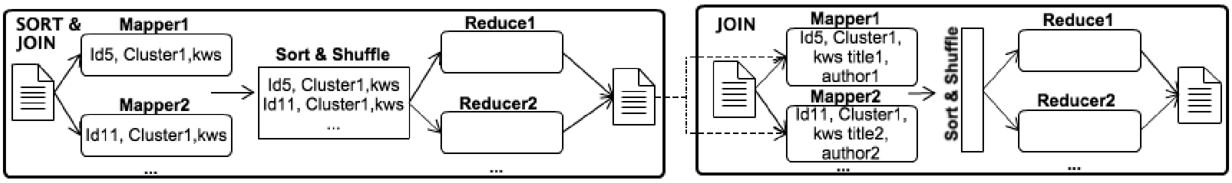

Figure 5. MapReduce model

There is an association between authors and their publications in each group. Our next step is know what a group is about. So in the next section, we label each cluster according with their keywords.

4.5.3. Label Groups

Search engines could increase performance in searches by finding a general topic area based in the words that belongs to a cluster. We can respond to specific queries (i.e.: show all researchers working in a specific area or all subareas belonging to a general topic area).

We use WordNet[9] [Miller 1995] to find synonyms, hypernyms, hyponyms and the concept of a word for all keywords in a cluster. It helps to find a common meaning in the way that words could occur together and find similar meanings. In other words, with the group of word set up we could find a concept or a topic for each cluster.

We applied Collapsed Variational Bayes (CVB) algorithm [Blei et al. 2003] that is an

[8] It is used to generate random centroids

[9] It is a lexical database for the English language that is used for text analysis applications.

implementation for Latent Dirichlet Allocation (LDA) in Mahout. We use all the words generated by WordNet plus the title and keywords of each publication to find a broader topic based in multiple subtopics described by the keywords. We use Mahout RowId to convert Term Frequency (TF) vectors into a matrix. The CVB algorithm was executed with the following parameters: 1 for the number of latent topics and 20 maximum interactions. This job is applied to each cluster.

Results are imported in the Central Repository usign RDF. Figure 6 shows the concepts and relationships used to export the results. The full arrow symbolize a relationship between classes and the dash arrow symbolize a common relationship.

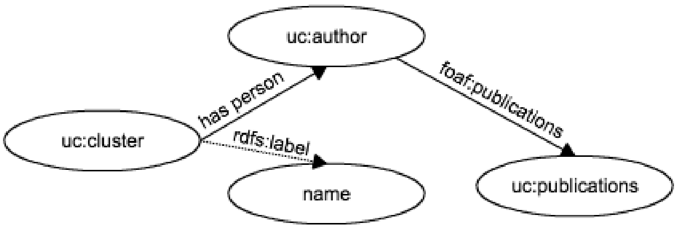

Figure 6. Concepts and relationships of RDF file.

5. Conclusion and Further Work

We have presented architecture that have a central repository with rich data from various bibliographic sources with a data model defined and described using ontologies that includes links to other data in other repositories.

Acknowledgements

En esta sección se agradece de manera cortés por la ayuda: científica, de redacción y técnica (equipo y otros materiales especiales) recibida de cualquier persona o institución. Además, en esta sección se expresa también un reconocimiento por la ayuda financiera externa (como subvenciones, contratos o becas) recibida tanto para la realización de la investigación como para la preparación del artículo. Debe ser breve.

References

Alfraidi, H. (2015). Interactive system for scientific publication visualization and similarity measurement based on citation network. Master's thesis, University of Ottawa.

Andrews, N. O. and Fox, E. A. (2007). Recent developments in document clustering. Virginia Tech. Department of Computer Science.

Barbarić, A. (2014). Isbd: International standard bibliographic description.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(30):993–1022.

Brickley, D. and Miller, L. (2012). Foaf vocabulary specification 0.98. *Namespace document*, 9.

Charikar, M. (2002). Similarity estimation techniques from rounding algorithms. In *STOC*.

Clark, C. and Divvala, S. (2015). Looking beyond text: Extracting figures, tables and captions from computer science papers.

Gangemi, A. (2005). *The Semantic Web – ISWC 2005: 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005. Proceedings*, chapter Ontology Design Patterns for Semantic Web Content, pages 262–276. Springer Berlin Heidelberg, Berlin, Heidelberg.

Giasson, F. and D'Arcus, B. (2009). Bibliographic ontology specification.

Krisnadhi, A. A., Hu, Y., Janowicz, K., Hitzler, P., Arko, R. A., Carbotte, S., Chandler, C., Cheatham, M., Fils, D., Finin, T., Ji, P., Jones, M. B., Karima, N., Lehnert, K., Mickle, A., Narock, T., OBrien, M., Raymond, L., Shepherd, A., Schildhauer, M., and Wiebe, P. (2015). The geolink framework for pattern-based linked data integration. In *SEMWEB*.

Ley, M. (2009). Dblp xml requests.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

O'Neill, E. T. (2002). Frbr: Functional requirements for bibliographic records. *Library resources & technical services*.

Ortiz Vivar, J. E. and Segarra Flores, J. L. (2015). Plataforma para la anotación semántica de servicio web restful sobre un bus de servicios.

Varadharajalu, A., Liu, W., and Wong, W. (2011). Author name disambiguation for ranking and clustering pubmed data using netclus. In *AI 2011: Advances in Artificial Intelligence*, pages 152–161. Springer.