Towards Linking DBpedia's Bibliographic References to Bibliographic Repositories

David Nazarian and Nick Bassiliades

Department of Informatics, Aristotle University, Thessaloniki, Greece
{dnazarian,nbassili}@csd.auth.gr

Abstract. The widespread usage of semantic resources such as SPARQL endpoints and RDF data dumps by an ever growing number of users requires steps to be made in order to ensure the correctness of the provided data. DBpedia, a major node of the LOD cloud is a contributor of both types with its content deriving from Wikipedia. This paper presents our effort towards creating alternative links for the DBpedia's bibliographic references motivated by the "DBpedia citations & references challenge". We present the procedure of the link creation by utilizing a Java library that we have developed, called BibLinkCreator, which extracts data from the DBpedia's references RDF data dump provided during the competition, and other RDF data dumps (available for download or collected via APIs) relevant to literature, based on unique identifiers such as ISBN, and links citation URIs after matching identifiers and ensuring the similarity of other properties.

Keywords: Linking bibliographic references · Link discovery · Linked open data · Semantic web

1 Introduction

The web is evolving gradually into a machine-understandable platform through the introduction of semantics into it. From a Web of Documents, it is transforming into a Web of Data [4, 3] with a number of potential benefits for the users. Rich content is being made available by a variety of sources through RDF data dumps (datasets) and Simple Protocol and RDF Query Language (SPARQL) endpoints. The linking of such content creates a global data space [3, 7], a semantic cloud.

A huge amount of interconnected datasets are already available for free usage forming a cloud known as the Linked Open Data (LOD). Each of its nodes is a contribution of RDF content by individuals or organizations, linked to one or more datasets through RDF links. DBpedia, which extracts knowledge from Wikipedia, is one of its main nodes connected to a multitude of other through incoming and outgoing RDF links [9]. Many of these RDF datasets are made available by libraries from around the world providing metadata about digitized content such as books, serial publications (e.g. journals), maps, printed music and other library related material [4]. Such bibliographic datasets are being produced by mapping from existing library metadata stand-

adfa, p. 1, 2011.

ards such as the Machine-Readable Cataloging (MARC) and the Encoded Archival Description (EAD) [4].

Since the content of the LOD cloud is changing constantly, some of its links may become obsolete, pointing either to outdated resources or to nonexistent ones. In order to ensure the correctness of the provided data, maintenance of the links is an important issue [3]. Providing alternative outgoing links for a source dataset is one way to increase its connectedness and ensure that other paths will exist for its pointed resources. Of course, the role of metadata about the temporal, provenance and trust dimensions of LOD should not be ignored.

This paper will present our effort towards creating alternative links for the DBpedia's references. For this purpose, we have created a Java Library that is capable of extracting and preprocessing data from pairs of RDF repositories, and linking them based on a combination of key-based and similarity-based approaches [7]. For the data extraction, we have used a number of unique identifiers, such as ISBN, ISSN, DOI, LCCN etc., as keys. A link is then created after ensuring the similarity of content such as identifier, title and publication year.

By using a number of link destination repositories, downloaded or created after collecting data through APIs, we managed to create 1,084,445 alternative links for the DBpedia's bibliographic references, with 761,235 corresponding to distinct URIs.

The paper is organized as follows: section 2 discusses about related work done by the Semantic Web community, section 3 presents details about the source and destination RDF data dumps downloaded or collected via APIs, section 4 describes the BibLinkCreator library that we developed in order to create the links, section 5 presents the link creation results, and section 6 presents our conclusions and future steps.

2 Related Work

Although our main intention was the creation of links for the DBpedia's bibliographic references, it also led us to the development of a domain-specific bibliographic link creator library called BibLinkCreator which relies mainly on bibliographical identifiers. The use of such identifiers to link data is a common practice. An example of an application which exploits them is the RDF Book Mashup, which uses APIs from sources such as Amazon, Google and Yahoo in order to integrate information into Semantic Web. For this purpose, it uses the ISBN or the author name encoded into a URI sent during a lookup call to query data sources [2].

There are many applications in which domain-specific data are collected and merged or linked in order to provide an easier way of access to information. URank is an application that collects data from various web sites containing information about university rankings, uniquely identifies the University entities by linking them to DBpedia LOD set and constructs a merged University ranking dataset [1]. The author in [6] presents an approach to create an active Linked Life Sciences Data Compendium which dynamically assembles queries retrieving data from multiple SPARQL endpoints, in order to easily navigate through different datasets.

Many link discovery frameworks have been developed and are available to facilitate the linking of datasets [12]. KnoFuss, LIMES and Silk are some of the well-known ones which include diverse capabilities. They are universal, not specific to a domain and incorporate different algorithms to reduce the search space for a given task.

3 Bibliographic Resources Used

Throughout the paper, the terms references and citations will be used interchangeably referring to the same concept. For the task of linking DBpedia's bibliographic references we used as a link source, the enwiki-20160305-citation-data[1] RDF data dump file provided during the "DBpedia citations & references challenge"[2] containing extracted metadata related to citation URIs from the English version articles of Wikipedia. In the majority of cases these URIs are actual web addresses (URLs) pointing to the cited resources. RDF data dumps and APIs providing the link destinations were chosen after analyzing the structure of the link source, and determining the portion of the citations for which an alternative link can be found. A total of five RDF data dumps and five APIs providing literature information about books, journals, periodicals, magazines etc. were chosen.

The greater portion of the citation URIs contained in the link source either refers to resources for which an alternative link does not exist or is difficult to be found. By alternative, we mean a content describing the same thing found elsewhere (in a different URI) not a related content. For example, there are many cases when a citation is being made to a website of an individual presenting his or her work which has not been published elsewhere and consequently no alternative link can exist. Many citations are being made to content such as YouTube videos not published elsewhere, or even to newspaper articles for which an alternative cannot be found easily. These reasons made us to concentrate on the portion of the link source which contains citations to literature available from many sources.

3.1 Link Source RDF Data Dump

As mentioned earlier the enwiki-20160305-citation-data file was used as a link source for the citations. It contains 97,468,830 triples which are reduced to 76,223,926 because of duplicates after inserting the file into a triplestore. The subjects of all the triples represent citation URIs with the majority of them (70.636.663 triples) representing actual URLs pointing to the cited resources, and the rest pointing to non-dereferenceable links of citation.dbpedia.org. There are 12,391,363 distinct citation subject URIs of different categories.

In order to distinguish which predicates are important and can be used during link creation, we sorted them by their counts and selected for examination the ones that

[1] http://downloads.dbpedia.org/temporary/citations/enwiki-20160305-citation-data.ttl.bz2
[2] http://wiki.dbpedia.org/blog/dbpedia-citations-references-challenge

had multiplicity greater than 5,000, since there are 3,620 predicates in total. From the 148 that resulted, only 16 predicates relevant to literature were selected for further examination as shown in Table 1. There are cases when a subject contains more than one of these predicates simultaneously.

Table 1. Literature relevant properties and their counts.

	Property name	**Count**		**Property name**	**Count**
1	journal	998,460	9	oclc	48,463
2	newspaper	664,742	10	issn	46,265
3	isbn	646,153	11	jstor	30,968
4	doi	584,056	12	magazine	27,833
5	pmid	347,067	13	encyclopedia	24,244
6	series	85,953	14	periodical	13,992
7	pmc	76,546	15	arxiv	13,624
8	bibcode	56,518	16	lccn	5,689

A search was conducted for each predicate and based on availability in online libraries and other criteria, only 11 of them were eventually selected. The predicates that were rejected are the following: newspaper, pmc, bibcode, jstor and encyclopedia. From the selected predicates, seven of them (isbn, issn, doi, pmid, oclc, arxiv and lccn) represent unique identifiers and the rest (journal, series, magazine and periodical) refer to titles of serial publications. The description of each identifier is given bellow:

1. **I**nternational **S**tandard **B**ook **N**umber (ISBN): A code that uniquely identifies each edition of a book. Two forms exist, the ISBN10 and the ISBN13. A conversion algorithm exists in order to convert an ISBN10 to an ISBN13.
2. **I**nternational **S**tandard **S**erial **N**umber (ISSN): An eight-digit code that uniquely identifies a serial publication such as a journal, a magazine etc. To each successive edition of the serial publication, the same ISSN is assigned.
3. **D**igital **O**bject **I**dentifier (DOI): A unique identifier that is assigned to a variety of things such as books, websites, articles etc. It is a string consisted of two parts, a prefix and a suffix separated by a forward slash.
4. **P**ub**M**ed **id**entifier (PMID): A sequentially assigned integer number to content that can be queried and retrieved via the PubMed search engine.
5. **O**nline **C**omputer **L**ibrary **C**enter (OCLC) control number: A sequentially assigned integer number to records of WorldCat.
6. ArXiv identifier: A unique identifier assigned to content submitted to arXiv.org.
7. **L**ibrary of **C**ongress **C**ontrol **N**umber (LCCN): A unique identifier assigned to records of the United States Library of Congress.

3.2 Link Destination RDF Data Dumps/APIs

Five link destination RDF Data Dump files were selected after the examination of the content provided by a number of libraries and literature relevant sources. We discovered many of these files through the datahub[3] website which hosts information about a multitude of datasets, and through searching for RDF content directly from different publishers and libraries.

In order for a file to be selected, a part of its content had to be described through one or many unique identifiers such as those described in section 3.1, and had to have dereferenceable subject URIs. There are many libraries providing a rich RDF content but lacking an identifier such as an ISBN. For example, TEL[4] (The European Library) which allows online access to resources of many European national libraries provides its RDF content based on an ontology that lacks a property related to ISBN, even though it contains information about books. There are also cases of libraries whose content is provided without dereferenceable URIs and consequently a link cannot be created by using them. Eventually we selected data dumps provided by DBLP, Springer, **B**iblioteca **N**acional de **E**spaña (BNE), **B**ritish **N**ational **B**ibliography (BNB) and **D**eutsche **N**ational**b**ibliografie (DNB).

There are many literature-related websites that provide their material through APIs in different serializations and formats, allowing individual or groups of items to be queried in most of the cases through unique identifiers, keywords or by other search criteria. Each website has its own policy in regard to the terms that apply to its API usage. Consequently, there can be limitations on the number of daily requests made, the number of items that can be queried in a single request, the interval between consecutive requests and other parameters. Although usually the provided content is not in a semantic form, it can be mapped to vocabularies in order to incorporate them into semantic applications.

Five APIs were selected based on a number of criteria such as, the types of the unique identifiers that could be queried, their limitations, their content, their responsiveness etc. We discovered many of them through a list of APIs[5] provided by MIT Libraries and through searching directly from publishers. Four of these are as described previously, and one of them provides data about individual items directly in different semantic forms. The selected link destination API providers are: arXiv, HathiTrust, Open Library, PubMed and WorldCat.

4 Bibliographic Link Creator Library

For the purpose of citation linking, a Java library called BibLinkCreator[6] (Bibliographic Link Creator) was developed. It is a domain-specific library which can be used to link URIs from any pair of bibliographic RDF datasets, if they contain

[3] https://datahub.io
[4] http://www.theeuropeanlibrary.org
[5] http://libguides.mit.edu/apis
[6] https://github.com/DavidNazarian/BibLinkCreator

metadata about unique identifiers such as those described in section 3.1, titles and publication years (optional in some cases). In order to communicate with these datasets which must be stored in a triplestore[7], it utilizes the Sesame API. Its main functionality is provided through a data extractor and a data linker class which are described below.

The abstract link creation procedure comprised of data extraction from two repositories (sources) followed by the data linking is illustrated in Fig. 1. The source X represents the repository providing the subject URIs of the links, while the source Y represents the repository providing the object URIs of the links. The parameters provided to the Data Extractor represent other inputs, such as SPARQL queries, user-defined string replacement data etc. The identifier and data collection steps include their preprocessing and validation. The identifier set C represents the distinct identifiers that are present in both repositories.

The data saved by the Data Extractor is used by the Data Linker for the link creation. The identifier category determines the query category that will be used in order to retrieve data from the two sources and the similarity checks that will be employed. The parameters provided to the Data Linker include inputs such as source repository names (aliases), the string similarity thresholds, the year maximum absolute difference threshold etc.

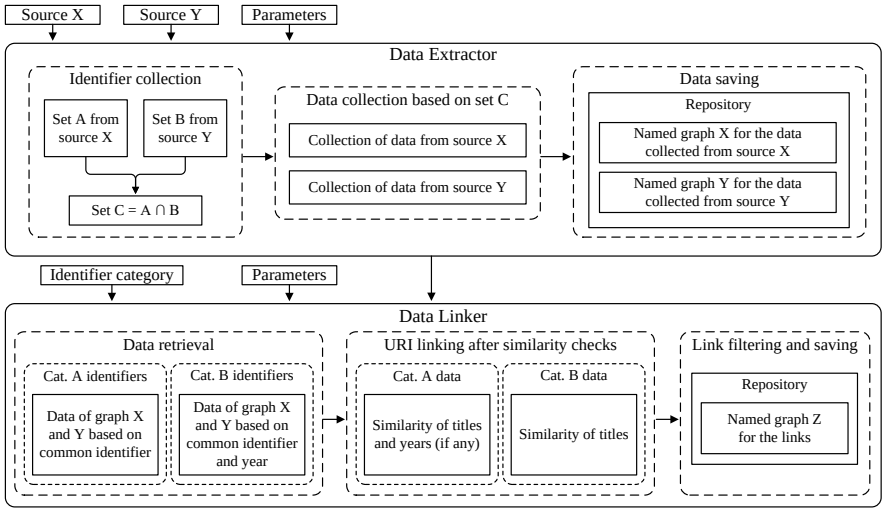

Fig. 1. Abstract link creation procedure.

4.1 Data Extractor

Since data from different sources can differ even though they describe the same entities, a mechanism is needed in order to make them uniform. The data extractor's main purpose is to extract, preprocess, validate and save specific data from the provided

[7] We have used the Ontotext GraphDB 7 Free edition http://ontotext.com/products/graphdb

semantic repositories, which will be later used by the data linker for the link creation. There are three categories of data that are being extracted, unique identifiers, titles and publication years.

Unique identifiers depending on their structure may be converted in other forms and may be validated if a validation algorithm exists for them. For example, preprocessing and validation steps for an extracted ISBN are the following:

1. Hyphens and other non-numeric characters are removed, except the letter x.
2. Then, it is checked for validity based on the ISBN algorithm and is kept or rejected accordingly.
3. If it represents an ISBN10 then it is converted to an ISBN13 because there are cases when the same ISBN is represented with different forms in different sources.

Extracted titles are strings representing the titles of books, specific editions of serial publications etc, which may contain letters (upper- and lower-case), numbers, and punctuation marks such as brackets, quotation marks, dashes and other. Since the same title can be present in different sources with different capitalization rules and with alternative punctuation marks, the preprocessing step converts all of their characters to lowercase and removes from them a number of punctuation marks (leaving single spaces between words). A 0-length resulting string is considered invalid.

Extracted publication years represent the years when books, specific editions of serial publications and other literature were published. Usually they are provided either as a string, a date or a gYear. During the preprocessing, all the non-numeric characters are removed from a string representation of the year, whereas from a date representation of year, only the year is kept, resulting in a string of four digits in both cases. A resulting string of length different than four is considered invalid.

The default preprocessing steps for each of these data categories can be preceded by user-defined string replacements. Sometimes this is necessary for removing parts of strings which if not removed can lead to corrupted results or as a step to increase the quality of the data. For example, there are cases when a prefix such as "ISBN10:" or "ISBN13:" exists in front of ISBN strings. If not removed, the default preprocessing (described previously by the three steps) would leave an undesired "10" or "13" in them leading to invalid ISBNs. To prevent it, a string replacement list containing both of these prefixes would ensure that they are replaced, for instance by an empty string.

In order to extract the three kinds of data described previously (identifiers, titles and publication years), a SPARQL query for each of the data sources is employed. Each query must expose a number of variables with specific names or aliases. There are two query categories depending on the identifier involved during the data extraction. The first (category-A) includes those that identify a specific literature item, such as a book, and the second (category-B) includes those that identify a range of literature items, such as a journal.

The identifiers extracted from the link source repository (set A in Fig. 1) are used to filter the data that will be extracted from the link destination repository (set B in Fig. 1), thus only absolutely relevant data are extracted (set C in Fig. 1).

4.2 Data Linker

The purpose of the data linker is to link URIs from two sources (identified as X and Y in Fig. 1) based on the similarity of identifiers, titles and years, by using a provided URI representing the type of the link, such as owl:sameAs [5], rdfs:seeAlso [11] etc. The data of the two sources are retrieved from the repository where the extracted data were saved. Depending on the identifier used to retrieve the data, there are two predefined SPARQL query categories directly corresponding to the ones described in the previous section.

The first category (category-A) data are retrieved from the two sources by using a common unique identifier variable name. This results into retrieving only the data accompanied with matched identifiers from both sources. The second category (category-B) data are retrieved from the two sources by using a common identifier and a common year variable name for both sources. This ensures that only editions of serial publications published during the same year are retrieved from both sources, since there are cases when the same title is being reused for editions of different years (e.g. recurrent conferences).

The usage of identifiers alone to link the data proved to be insufficient, because there are cases when by human error they are assigned to irrelevant URIs. These kinds of errors are considerably higher in datasets produced by community effort. Identifiers combined with titles and years ensure that the metadata of the two sources are relevant to each other and increase confidence on the correctness of the result.

The string similarity measures that have been incorporated into the library, to compare the titles are the Jaccard, Dice, Overlap and Cosine coefficient [10]. All of them take as an input two strings, and return a real number ranging from 0 (no similarity) to 1 (absolute similarity). They can be parameterized in regard to the string segmentation type which will be used to create the comparison sets for each input.

To compare the titles retrieved by the category-B queries, a stricter threshold should be used in contrary to those of the category-A. This is because, even different editions of serial publications published during the same year and identified by the same ISSN can have very similar titles. For example, "The History of Poland" and the "The History of Portugal" are different editions of the "The Greenwood Histories of the Modern Nations" (ISSN: 1096-2905), both published in the year 2000, and have very similar titles. A low threshold would wrongly deem these different titles similar, whereas a high threshold would correctly differentiate them.

After data from the two sources have been found to refer to the same entity, the last step before the link creation is to check whether the subject and object URIs of the created links are different. For URIs referring to doi.org, there is an optional capability of resolving their redirected addresses in order to ensure that even after the redirection the URIs remain different. This is done by retrieving metadata[8] from doi.org in JSON format which contain the redirection information.

[8] http://www.doi.org/factsheets/DOIProxy.html

5 Results

By utilizing the BibLinkCreator library, we managed to create 1,084,445 alternative links for the DBpedia's bibliographic references with 761,235 corresponding to distinct subjects. This covers 761,235 / 12,391,363 = 6.14% of the distinct subjects found in the entire file and 761,235 / 1,719,223 = 44.3% of the distinct subjects that contain one of the 11 selected predicates described in section 3.1. Notice that although the coverage seems to be small, the majority of the uncovered links are not about bibliographic references but other stuff, such as YouTube videos, newspaper articles, etc., as mentioned in section 3. In order to have more accurate results, we excluded the subjects containing a "chapter" predicate, because usually its presence indicates that a subject URI is referring to a specific chapters of literature item, and thus cover 761,235 / 1,646,484 = 46.23%.

For the link category-A, we used the Overlap coefficient as a string similarity measure, parameterized with a threshold ≥ 80% for the title comparisons, when year data have been retrieved, and ≥ 90% otherwise. The years have been compared with 1-year maximum absolute difference threshold. For the link category-B, the Dice coefficient, which is equivalent to the F1 score, has been used with a threshold ≥ 97% for the title comparisons. Both similarity coefficients have been parameterized with character level shingles of k = 2. We chose those parameter values by experimenting with different string similarity measures, thresholds, shingle types and shingle sizes, but mostly taking into account the nature of the data.

The accuracy of the results cannot be evaluated using automated methods since URL content comparisons are involved. A sampling of the results showed that inaccuracies are negligible relative to the number of the links found and are largely due to inaccurate metadata of the DBpedia's citations. Proportionally, the majority of inaccuracies occur in the category-B links, because of cases when a URI describing a serial publication is provided instead of a URI describing a specific edition of that serial publication, even though the provided metadata describe the specific edition.

As a predicate for the created links we have used rdfs:seeAlso instead of owl:sameAs because a) it is more "safe" considering the uncertainties that can exist in the results, b) because there are links pointing to metadata URIs and c) because the resources involved in the link creation from the two sources do not share all the same properties as needed for owl:sameAs [8, 5]. The created links are available to be queried through different named graphs from a SPARQL endpoint[9].

6 Conclusions and Future Work

We have presented our effort towards the link creation procedure for the bibliographic references of DBpedia which is one of the main nodes of the LOD cloud by utilizing a bibliographic link creator library called BibLinkCreator that we developed for this

[9] http://lod.csd.auth.gr:7200/sparql

purpose. A combination of key-based and similarity-based approaches were used, because of errors present in the data or the sources.

Future work can include the use of more link destination data sources (RDF data dumps and APIs) in order to create new links, using already existing links in the metadata of the destination data sources to increase the number of alternative links, the incorporation of the ability to use SPARQL endpoints and other communication APIs such as Jena to retrieve data, and the incorporation of Machine Learning techniques in order to reduce the user effort into specifying the parameters needed for the two link categories discussed in the section 4.2. Finally, a proper evaluation for accuracy and completeness should be performed by crowdsourcing methods, as well as a comparison with general-purpose tools, such as SILK, LIMS, etc., in order to be used as a baseline of comparison for the performance of our tool.

References

1. Bassiliades, N.: Collecting University Rankings for Comparison Using Web Extraction and Entity Linking Techniques. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI 2014. vol. 469, Springer, Heidelberg (2014)
2. Bizer, C., Cyganiak, R., Gauß, T.: The RDF Book Mashup: From Web APIs to a Web of Data. In: Auer, S., Bizer, C., Heath, T., Aastrand Grimnes, G. (eds.) SFSW 2007. vol. 248, CEUR Workshop Proceedings, Aachen (2007)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. IJSWIS 5(3), 1-22 (2009)
4. Godby, C.J., Wang, S., Mixter, J.K.: Library Linked Data in the Cloud: OCLC's Experiments with New Models of Resource Description. Morgan & Claypool, San Rafael (2015)
5. Halpin, H., Hayes, P.J.: When owl:sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web. In: Bizer, C., Berners-Lee, T., Hausenblas, M. (eds.) LDOW 2010. vol. 628, CEUR Workshop Proceedings, Aachen (2010)
6. Hasnain, A.: Improving discovery in Life Sciences Linked Open Data Cloud. In: Ciravegna, F., Vidal, M.E., (eds.) ISWC 2015. vol. 1491, CEUR Workshop Proceedings, Aachen (2015)
7. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool, San Rafael (2011)
8. Jaffri, A., Glaser, H., Millard, I.C.: Managing URI Synonymity to Enable Consistent Reference on the Semantic Web. In: Bouquet, P., Halpin, H., Stoermer, H., Tummarello, G. (eds.) IRSW 2008. vol. 422, CEUR Workshop Proceedings, Aachen (2008)
9. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web 6(2), 167-195 (2015)
10. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
11. Matinfar, F., Nematbakhsh, M.: Specializing RDFS: See Also in Semantic Web. In: IJWesT 2012. vol. 3, no. 1, AIRCC, Chennai (2012)
12. Nentwig, M., Hartung, M., Ngonga Ngomo, A.C., Rahm, E.: A Survey of Current Link Discovery Frameworks. Semantic Web 8(3), 419-436 (2017)