Using Linked Open Data to Bootstrap a Knowledge Base of Classical Texts

Matteo Romanello[1] and Michele Pasin[2]

[1] École Polytechnique Fédérale de Lausanne, Route Cantonale, 1015 Lausanne, Switzerland matteo.romanello@epfl.ch
[2] Springer Nature, The Campus, 4 Crinan Street, London N1 9XW, UK
michele.pasin@springernature.com

Abstract. We describe a domain-specific knowledge base aimed at supporting the extraction of bibliographic references in the domain of Classics. In particular, we deal with references to canonical works of the Greek and Latin literature by providing a model that represents key aspects of this domain such as names and abbreviations of authors, the canonical structure of classical works, and links to related web resources. Finally, we show how the availability of linked data in the emerging Graph of Ancient World Data has helped bootstrapping the creation of our knowledge base.

1 Introduction

Knowledge bases are essential resources for many Natural Language Processing tasks – such as for example the disambiguation of named entities or of word senses – as they provide algorithms with some surrogate of the knowledge needed to handle and capture certain aspects of our natural language.

The resource discussed in this paper is a domain-specific knowledge base aimed at supporting the extraction of bibliographic references in the domain of Classics. In particular, we deal with references to canonical works of the Greek and Latin literature, one out of many kinds of references to be found within publications in this field (e.g. references to fragmentary texts, inscriptions, papyri, manuscripts, coins and museum objects). One peculiarity of canonical references is that, by definition, they transcend (i.e. abstract from) specific editions or translations of a text.

This knowledge base contains various types of information that are needed to extract and disambiguate canonical references, such as:

1. names (and abbreviations) of ancient authors;
2. titles (and abbreviations) of ancient works;
3. unique identifiers of authors, works and citable passages of these works;
4. links to the Wikipedia pages of ancient authors;
5. information about the canonical citation structure of ancient works.

Although there exist several online resources from which this sort of information can be gathered – such as the Perseus Catalog[3] and the Classical Works Knowledge Base (CWKB)[4] – our knowledge base makes up for the lack of a single resource to support this information extraction task, that is suitable for use in NLP applications as well as to publish the extracted references using Semantic Web standards.

The creation of our knowledge was informed by the following principles:

1. it should be based on interoperable standards so as to increase the chances of being reused in other contexts;
2. it should be easy to use programmatically;
3. it should be linked as much as possible to other available resources as they provide complementary information about other facets of the data;
4. it should be easy to edit, maintain and update in the future.

Devising a technical solution that fulfills all of these principles is not entirely trivial as some of these principles may seem to contradict each other (e.g. "based on interoperable standards" and "easy to use programmatically"). In this paper we present our proposed solution, describe how it was implemented and explain how it is used in practice.[5] We also show how the availability of linked data in the emerging Graph of Ancient World Data [4] has helped bootstrapping the creation of our knowlegde base.

2 Motivation and Background

Much of the Graph of Ancient World Data (GAWD) is emerging as a community of practice has developed that values the use of shared controlled vocabularies based on URIs to refer to 'things' [3].

As emerges also from the GAWD cloud diagram[6], the Pleiades gazetteer has played a key role in the growth of LOD for the ancient world, with Pelagios building upon it to connect even more resources. In its current phase Pelagios has become *de facto* a LOD-facilitator: its community platform, Pelagios Commons, is the go-to place for those who are keen to apply Pelagios' model/philosophy to other areas of the study of the ancient world.

After geographical data, something similar is taking place with regards to the recently developed time-gazetters PeriodO and Chronontology, which have started to enable the interlinking of datasets based on shared references to time periods[7].

[3] Perseus Catalog, `http://catalog.perseus.org`.
[4] Classical Works Knowledge Base, `http://www.cwkb.org`.
[5] The HuCit Knowledge Base can be explored via a Linked Open Data (LOD) front-end available at `purl.org/hucit/kb/`.
[6] Graph of Ancient World Data by Régis Robineau (19/06/2012), `http://bsa.biblio.univ-lille3.fr/doc/gawd/gawd.html`.
[7] Pelagios commons – Time Working group, `http://commons.pelagios.org/groups/time-events-working-group/`.

In addition to space and time, references to ancient texts are undoubtedly another dimension that could be leveraged to expand the GAWD cloud, as the cited primary sources are often an area where existing datasets do overlap. However, what is still missing to realise this potential are resolvable URIs for all citable sections of canonical texts. We do have – thanks to the CTS protocol developed for the Homer Multitext project – a scheme of unique identifiers that can be used to identify those citable units of texts, i.e. the CTS Uniform Resources Names (URNs). This protocol was implemented also by the Perseus catalog, meaning that each canonical author and work can now be looked up by its CTS URN in the catalog.[2]

The only thing that is currently missing – which is one of the aims of our knowledge base – is to have fully resolvable URIs for all citable passages of classical texts, linked to other resources like Perseus (catalog and library) and CWKB. One of the advantages of having such URIs in place is that we will then be able to use them in combination with ontologies like the FRBR-aligned Bibliographic Ontology (FaBiO) and the Citation Typing Ontology (CiTO) [6] for publishing citation data on the Semantic Web.

3 Uses of the Knowledge Base

The knowledge base is one of four components of the system described in [7] to extract automatically canonical references from text (see Fig. 1). The other components are a Citation Extractor (2) that takes care of identifying the citation components within the stream of text; a Citation Matcher (2) that attempts to disambiguate the cited ancient work against the knowledge base and, in turn, relies on a Citation Parser (4) to normalise the reference scope into a format that can be readily embedded into a CTS URN.[8]

The extraction and disambiguation of canonical references was modelled as a three-step process consisting of the following steps:

1. **Extraction of named entities**: a) names of ancient authors (e.g. Virgilio); b) titles of works (e.g. Aeneid) and c) references to specific text passages (e.g. Virg., Aen. 12.10 f.).
2. **Detection of relations between entities**: since a reference is represented as a relation between two entities (i.e. the author name/work title and the reference scope), the canonical references are reconstructed from the entities found in the text. For example, the reference "12.10 f." is expressed as a relation between the entity identifying the cited text (in this case "Virg. Aen.") and the entity indicating the citation scope ("12.10 f."), namely the precise text passage being cited.

[8] Knowledge Base, Citation Extractor and CitationParser are openly available as Python libraries respectively at `https://github.com/mromanello/CitationExtractor`, `https://github.com/mromanello/CitationParser` and `https://github.com/mromanello/hucit_kb` (the CitationMatcher component is part of the CitationExtractor).

3. **Disambiguation of named entities and relations**: determining which authors, works and passages are referred to in the text is done by assigning a unique identifier to each entity and relation. The reference in the example above, for instance, will be assigned the URN "urn:cts:latinLit:phi0690.phi003:12.10-12.11". This identifier is built by concatenating the URN for the cited work (urn:cts:latinLit:phi0690.phi003 for Virgil's Aeneid) with a normalised value representing the cited passage (12.10-12.11 which stands for book 12, lines 10 and 11).

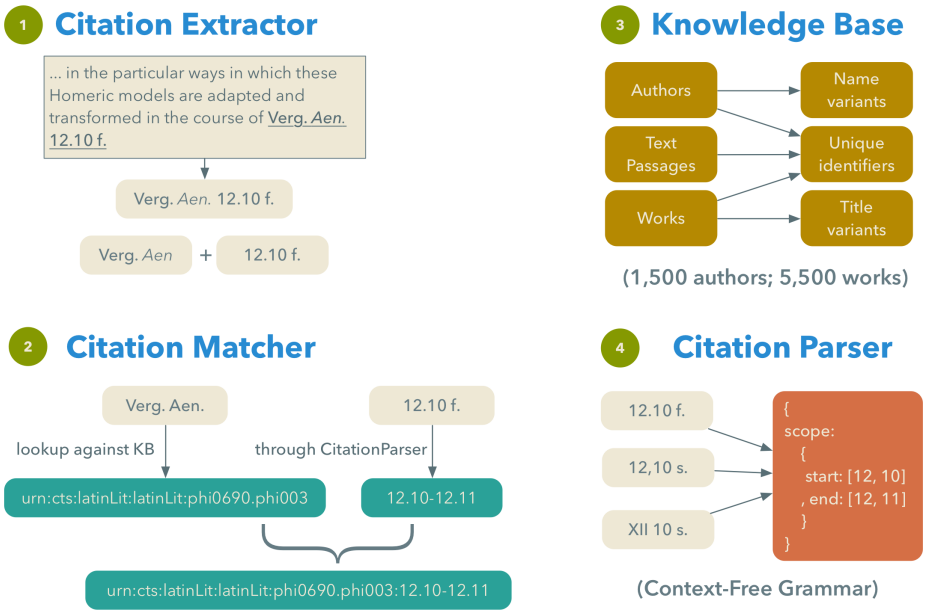

Fig. 1. The four software components used for the extraction of canonical references.

3.1 Linking References to Full-text

The first use of the knowledge base – and perhaps the most important at least from an end-user perspective – is the linking of extracted references with their corresponding full text passage. Since canonical references by definition transcend (i.e. abstract from) specific editions or translations of the text, the linking of a reference to its full text needs to enable the reader to select the very edition or translation she is after from those available online. To this end, we rely on two external services: the Perseus Digital Library for openly available editions and translations, and CWKB for texts whose access requires an institutional

subscription (e.g. the Thesaurus Linguae Graecae). In particular, the latter provides the ability of resolving links in a context-aware fashion: if the user selects to read the TLG text of a passage and her institution has access to it, the service will redirect the browser directly to the full-text.

3.2 Generation of Dictionaries

The second use case is the generation of dictionaries – i.e. lists – of names, titles and respective abbreviations that are used at various stages of the extraction of canonical references. The dictionaries of abbreviations are employed in the process of splitting texts up into sentences and then into tokens. Their use allows us to prevent some errors that are commonly caused by the presence of punctuation within abbreviations. Such dictionaries are of particular importance for the extraction of information – i.e. author names, work titles and canonical references – from texts written in several European languages as they enable the citation extraction system to relate different spelling variants to the same entity.

3.3 Disambiguation of References

Information contained in the Wikipedia page of a given ancient author can be used to help the automatic disambiguation of canonical references, a technique that is used in almost any Named Entity Disambiguation system (cfr. [9]). The rationale behind this is that the words and entities contained in the Wikipedia page of a given author have some overlap with the context where a reference to that author appears. Typically, this information is leveraged by computing a similarity score between the document where the reference is found and the Wikipedia page of every disambiguation candidate, usually extracted from a knowledge base by using some heuristics. This score is then used in combination with other features to establish the ranking of the disambiguation candidates, whose aim is to rank the correct entity as first.

4 Knowledge Base Implementation

4.1 Data Model

The rationale for developing the knowledge base's data model was to re-use as much as possible already existing and widely adopted ontologies, and to extend them by means of new classes and properties only when absolutely necessary.

The first two ontologies that form the backbone of the HuCit knowledge base are CIDOC-CRM and FRBRoo.[9] The CIDOC-CRM is a conceptual model that was born as a metadata standard for the archive and museum world, and proved to be suitable to represent information in many different domains. The subset of CIDOC-CRM classes and properties used by the knowledge base is limited

[9] See respectively `http://www.cidoc-crm.org/` and `http://www.cidoc-crm.org/frbroo/`.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix hucit: <http://purl.org/net/hucit#> .
@prefix ecrm: <http://erlangen-crm.org/current/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix efrbroo: <http://erlangen-crm.org/efrbroo/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://purl.org/hucit/kb/authors/678>
    a efrbroo:F10_Person ;
    ecrm:P1_is_identified_by
     ↪  <http://purl.org/net/hucit-kb/authors/678#cts_urn>,
     ↪  <http://purl.org/net/hucit-kb/authors/678#name> ;
    owl:sameAs
     ↪  <http://data.perseus.org/catalog/urn:cts:latinLit:phi0690/>,
     ↪  <http://cwkb.org/author/id/678/>,
     ↪  <http://viaf.org/viaf/8194433> .

<http://purl.org/hucit/kb/authors/678#name>
    a efrbroo:F12_Name ;
    ecrm:P139_has_alternative_form
     ↪  <http://purl.org/hucit/kb/authors/678#abbr> ;
    rdfs:label "P. Vergilius Maro"@la, "P. Virgilius Maro"@la, "Publio
     ↪  Virgilio Marone"@it, "Publio Virgilio Marón"@es, "Publius
     ↪  Vergilius Maro"@la, "Publius Virgilius Maro"@la, "Vergil",
     ↪  "Virgil"@en, "Virgile"@fr .

<http://purl.org/hucit/kb/authors/678#abbr>
    ecrm:P2_has_type <http://purl.org/hucit/kb/types/abbreviation> ;
    a ecrm:E41_Appellation ;
    rdfs:label "Verg.", "Virg." .

<http://purl.org/hucit/kb/authors/678#cts_urn>
    ecrm:P2_has_type <http://purl.org/hucit/kb/types/CTS_URN> ;
    a ecrm:E42_Identifier ;
    rdfs:label "urn:cts:latinLit:phi0690" .
```

Fig. 2. Knowledge base example: the record for Virgil expressed as Turtle RDF.

to those that represent things like names, titles, abbreviations and for ancient authors and works (for an example, see Fig. 2). It is worth noting, however, that we try as much as possible to harmonise our use of CIDOC-CRM with the adoption of other essential standards, like the CTS protocol, that exist outside of the CRM world. For instance, we make extensive use of CTS URNs, which are declared as instances of CIDOC-CRM's `E42_Identifier` having a specific `E55_Type`.

FRBRoo is an implementation of the FRBR model, aligned with the CIDOC-CRM [5]. The FRBR model has been widely adopted in the field of Digital Clas-

sics as its hierarchy is suitable to describe the kind of bibliographic information scholars in this field deal with [1]. The FRBRoo classes used by our knowledge base are those concerned with the representation of authorship, namely the fact that a given conceptual work was created by someone at a certain point in time (e.g. Ovid's creation of the *Metamorphoses*).

The third and last ontology involved is the Humanities Citation Ontology (HuCit). This ontology was developed as a lightweight extension of CIDOC-CRM and FRBRoo aimed specifically at formalising the canonical text structures that are used to cite classical texts (see [8]; [7], pp. 85-100). This ontology allows us to instantiate any single citable unit of a canonical text (e.g. all lines in all books of Homer's *Iliad*), an ability of essential importance when representing canonical citations. Fig. 3 shows how the canonical text structure of Virgil's *Aeneid* can be modelled, and how it relates to a canonical citation.

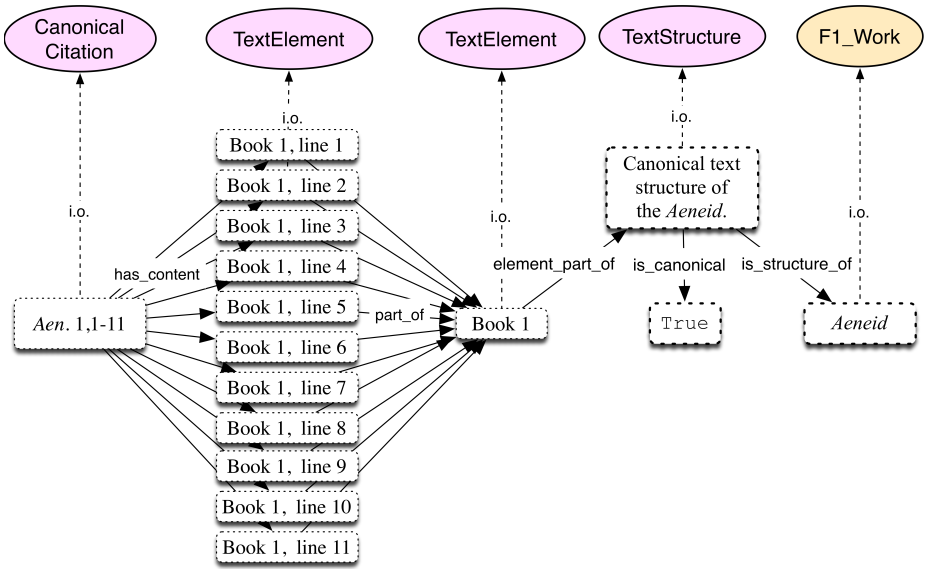

Fig. 3. Modelling the content of a canonical citation with HuCit's classes TextElement and TextStructure, and FRBRoo's F1_Work.

4.2 Populating the Knowledge Base

Our knowledge base is populated from (and linked to) three main datasources:

1. the Classical Works Knowledge Base (CWKB);
2. the Perseus Digital Library and Catalog;
3. Wikidata.

Firstly, the CWKB was used to perform the initial import of ancient authors and works into the knowledge base. Since CWKB has recently begun to publish its data as LOD, it was possible to harvest programmatically its content. Each author and work in our resources is aligned to CWKB by means of an `owl:sameAs` property; by doing so, it will be possible to benefit from new data that will be added to this resource in the future. Thanks to the CWKB it was possible to add to the knowledge base a substantial amount of authority data (see Table 1), to which others were added at later stages. The distribution of author names and work titles is rather uneven, as the high variance of their distribution reported in table 1 confirms; alongside authors and works with only one name/title variant each, there are others with a much higher number of lexical information attached.

	Total	Min	Max	Mean	Variance
Author names	4842	1	27	3.13	9.81
Author abbreviations	774	0	2	0.50	0.26
Work titles	10354	1	31	1.99	6.42
Work abbreviations	2377	0	3	0.46	0.57

Table 1. Basic statistics about the lexical information contained in the knowledge base for ancient authors (n=1548) as well as their works (n=5199).

Secondly, the Perseus Digital Library was used to import information concerning the canonical text structures according to which ancient works are cited in the scholarship (e.g. Homer's *Iliad* division into books and lines). CWKB records are linked to Perseus in two ways: first, by means of `owl:sameAs` links pointing to author or work records in the Catalog; and, second, by means of a `dcterms:identifier` (in the Dublin Core vocabulary) recording the CTS URN of an author/work (e.g. `urn:cts:greekLit:tlg0012.tlg001` for Homer's *Iliad*). It is worth noting that, currently, links to Perseus are available only for a subset of the CWKB records, and thus of our knowledge base. Increasing this coverage as much as possible is one of the goals of our project for the next future.

Since the canonical divisions of texts are encoded as markup elements within the digital editions and translations contained in Perseus, it is possible to leverage such information in order to instantiate the relevant HuCit classes (i.e. TextStructure and TextElement). This operation can be fully automated given that the content in Perseus is accessible programmatically by using its CTS API. The process involves gathering two pieces of information for each work contained in the knowledge base and in Perseus: first, information about the hierarchical structure of the canonical text divisions (e.g. the book/line structure of the *Iliad*); second, a list of all the citable elements that make up such a structure (e.g. a list of all the books and lines in the *Iliad*). Once all citable elements of a text have been instantiated and imported into the knowledge base, they can be used e.g. as the subject or the object of RDF statements.

Thirdly, we have been adding – whenever possible – `owl:sameAs` links pointing to VIAF and Wikidata records of ancient authors. We started with those authors that are linked to the Perseus Catalog and for which the Catalog provides a VIAF identifier. The main reason for doing this is that is to help the disambiguation of references, as described above. By following the chain of links from Perseus to VIAF and then to Wikidata, we were able to query Wikidata's SPARQL endpoint to get the links to the Wikipedia pages in the languages we are interested in (French, German, English, Italian and Spanish).

Finally, the knowledge base is being constantly updated with new variant forms and abbreviations as they are encountered while extracting canonical references from text corpora like JSTOR or *L'Anneée Philologique*.

4.3 Interfaces

Although the choice of CIDOC-CRM and FRBRoo for the data model certainly makes our data more interoperable and increases the chances of them being reused in other contexts, it imposes certain affordances when it comes to use the knowledge base from within an NLP application.

An apt example of these affordances is provided by authorship, a concept that CIDOC-CRM treats as an event: an author is the subject involved in the event that leads to the creation of a given work. As a result, retrieving the works by a certain author implies a) finding all creation events in which this author was involved and b) finding all the works that were created in such events. Since the knowledge base is stored in a triple store, this and many other queries will have to be written in SPARQL, leading the application to grow in complexity.

In order to keep the knowledge base as much as possible easy to use programmatically, without having to give up the advantages of CIDOC-CRM mentioned above, we used a Python Object RDF Mapper library called SuRF [10]. SuRF works similarly to an Object-relation Mapper with the difference that, instead of mapping a relation database to instances of Python objects, it maps a triple store to such objects. This allows us to interact programmatically with the knowledge base (see Fig. 4), and to hide away certain complexities of the underlying data model.

The following interfaces are currently available:

- a SPARQL endpoint to a Virtuoso triple store;
- a LOD interface, provided by the package Pubby, which makes the URIs of resources contained in the triple store resolvable to various formats (HTML, RDF/XML, RDF/Turtle);
- a Command Line Interface (CLI), aimed at easing the task of adding new information to the knowledge base.[11]

[10] SuRF – Object RDF Mapper, `http://pythonhosted.org/SuRF/`.

[11] All code and data for the knowledge base can be found at `https://github.com/mromanello/hucit_kb`.

```
>>> from pkg_resources import *
>>> from knowledge_base import KnowledgeBase

# Initialise the KB to use a remote Virtuoso triple store as back-end
# (an example configuration file is shipped with the library).
>>> conf = resource_filename('knowledge_base','config/virtuoso.ini')
>>> kb = KnowledgeBase(conf)

# Search for records with label `Omero`
>>> search_results = kb.search('Omero')
>>> print len(search_results)
1

# Find out how many ancient authors are contained in the KB
>>> all_authors = kb.get_authors()
>>> print len(all_authors)
1548

# And how many works in total?
>>> print sum([len(author.get_works()) for author in all_authors])
5199
```

Fig. 4. Example of interacting with the knowledge base by using its Python API.

5 Conclusions and Further Work

In this paper we have presented a domain-specific knowledge base aimed at supporting the extraction of bibliographic references in the domain of Classics. In addition to lexical information about ancient authors and works (variants spellings, abbreviations), this knowledge base will contain a record for any citable passage of canonical texts, thus making it possible to use it also in order to publish the extracted citation data by means of existing ontologies such as CiTO.

Since the main intended use of the knowledge base is within NLP applications, we developed a solution that neatly separates the data model – i.e. the ontologies used to represent the data – from the code library used to access and query the knowledge base. Such a solution has the advantage of hiding the complexities of the data model when accessing the contents of the knowledge base. In our specific case, this allowed us to build upon the CIDOC-CRM to model the data, while hiding its complexity – or even just specific design patterns it enforces – at the level of the code interfaces.

Furthermore, we have shown how the availability of LOD about classical texts, part of the so-called *Graph of Ancient World Data*, has enabled us to bootstrap the creation of our knowledge base. Existing links between CWKB and Perseus, as well as between Perseus and VIAF, greatly eased our task of populating the knowledge base for a limited number of ancient authors and works. At the same time, the external resources we are linking to will potentially be

able to aggregate information from our knowledge base. For example, by using the VIAF URI for Homer, an external service could derive a list of publications where Homeric works are cited, simply by following chains of `owl:sameAs` relations.

Further developments of the knowledge base in the near future will be aimed at increasing its coverage both in breadth and depth. While the number of classical authors and works does not grow, we aim to add links to VIAF, Wikidata and Perseus for as many entries as possible, and to continue enrich the knowledge base with lexical information. Also, a web user interface is planned so as to make it easier to engage a wider community of users in growing collaboratively the knowledge base.

References

1. Babeu, A., Bamman, D., Crane, G., Kummer, R., Weaver, G.: Named Entity Identification and Cyberinfrastructure. In: Kovács, L., Fuhr, N., Meghini, C. (eds.) Research and Advanced Technology for Digital Libraries, pp. 259–270. Springer (2007), http://dx.doi.org/10.1007/978-3-540-74851-9_22
2. Crane, G., Almas, B., Babeu, A., Cerrato, L., Krohn, A., Baumgart, F., Berti, M., Franzini, G., Stoyanova, S.: Cataloging for a Billion Word Library of Greek and Latin. In: Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage. pp. 83–88. DATeCH '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2595188.2595190
3. Elliott, T., Heath, S., Muccigrosso, J.: Prologue and Introduction. ISAW Papers 7(1) (2014), http://dlib.nyu.edu/awdl/isaw/isaw-papers/7/elliott-heath-muccigrosso/
4. Isaksen, L., Simon, R., Barker, E.T., de Soto Cañamares, P.: Pelagios and the emerging graph of ancient world data. In: Proceedings of the 2014 ACM conference on Web science - WebSci '14. pp. 197–201. ACM Press, New York, New York, USA (2014), http://dl.acm.org/citation.cfm?doid=2615569.2615693
5. Le Boeuf, P.: A Strange Model Named FRBROO. Cataloging & Classification Quarterly 50(5-7), 422–438 (2012), http://dx.doi.org/10.1080/01639374.2012.679222
6. Peroni, S., Shotton, D.: FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. Web Semantics: Science, Services and Agents on the World Wide Web 17, 33–43 (2012)
7. Romanello, M.: From Index Locorum to Citation Network: an Approach to the Automatic Extraction of Canonical References and its Applications to the Study of Classical Texts. Ph.D. thesis, King's College London (2015), http://hdl.handle.net/11858/00-1780-0000-002A-4537-A
8. Romanello, M., Pasin, M.: Citations and Annotations in Classics : Old Problems and New Perspectives. In: DH-CASE '13 Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: metadata, vocabularies and techniques in the Digital Humanities. ACM, New York, NY, USA (2013), http://dx.doi.org/10.1145/2517978.2517981
9. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering 27(2), 443–460 (feb 2015)