EvdoGraph: A Knowledge Graph for the EVDOXUS Textbook Management Service for Greek Universities

Nick Bassiliades[a]
School of Informatics, Aristotle University of Thessaloniki, Greece

Keywords: Linked Open Data, Knowledge Graphs, Ontologies, DBpedia, SWI-Prolog.

Abstract: Evdoxus is a web information system for the management of the total ecosystem for the free provision of textbooks to the undergraduate students at the Greek Universities. Among its users are book publishers that register textbooks, faculty members that search for appropriate textbooks for their courses, administration of university departments that register the relevant textbooks for each module of the curricula (course), and finally, students that select one book per module that they attend. All the above information (except for which students selected which books) is freely available at the Evdoxus site in the form of HTML web pages. In this paper, we present how we extracted this information and converted it into an open Knowledge Graph in RDF that can be used to generate several interesting reports and answer statistical analysis questions in SPARQL. The KG is backed by a simple ontology which is aligned with some well-known ontologies. The extraction / conversion application has been developed using SWI-Prolog's XPath and Semantic Web libraries. The KG encompasses the Linked Open Data initiative by linking University instances with their corresponding DBpedia entries, employing the Wikipedia search engine and the DBpedia SPARQL endpoint.

1 INTRODUCTION

Evdoxus[1] (or Eudoxus) is an online service (web information system) for the management of the total ecosystem of the free provision of university textbooks to undergraduate students at the Greek Universities. It was launched in the academic year 2010-2011 and it offers: a) accurate online information about the textbooks that are available for each course / module; b) quick delivery of the books to the students; c) effective mechanisms for publishers' compensation; d) parallel distribution of free e-books and notes; e) public resources' abuse prevention; and f) more transparency and less bureaucracy. Its users include book publishers that register their textbooks, faculty members that search for appropriate textbooks for their courses, administration staff of departments that register the textbooks selected by the professors for each module of the curricula (course), and finally, students who, after registering at their University's Student Information System for the modules they will attend for each semester, they select one book per module that they attend. All the above information (except for which students selected which books) is freely available at the Evdoxus site in the form of HTML web pages.

Knowledge Graphs (KGs) are a powerful way of representing and integrating structured knowledge from either a single or various sources (Hogan, et al., 2022). They consist of nodes that represent entities, and edges that represent the relationships between them. KGs can be used to support various tasks, such as data analytics, information retrieval, question answering, recommendation, natural language understanding and computer vision. One of the key aspects of KGs is their use of ontologies to provide a formal representation of the entities and their relationships. Ontologies enable logical inference and reasoning over the KG, as well as consistency checking and validation. Ontologies also facilitate the interoperability and integration of different knowledge sources, by providing a common vocabulary and schema.

Linked open data (LOD), which is an older, alternative term for KGs, is a vision of making data on the web accessible, interoperable, and reusable. LOD is based on the principles of linked data, which use

[a] https://orcid.org/0000-0001-6035-1038
[1] https://eudoxus.gr/

Bassiliades, N.
EvdoGraph: A Knowledge Graph for the EVDOXUS Textbook Management Service for Greek Universities.
DOI: 10.5220/0012153600003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 2: KEOD*, pages 17-28
ISBN: 978-989-758-671-2; ISSN: 2184-3228

URIs to name and identify things, HTTP to allow these things to be looked up and dereferenced, and RDF to provide structured data using controlled vocabularies (Bizer, Heath, & Berners-Lee, 2009). LOD enables data from different sources and domains to be connected and queried in a semantic way. This can create new insights and value from the data, as well as foster innovation and collaboration. Tim Berners-Lee has proposed a 5-star scheme for grading the quality of open data on the web, where the highest ranking is given to LOD (Berners-Lee, 2009). According to this scheme, 5-star LOD meets the following criteria: a) Data is openly available in any format (1 star); b) Data is available as machine-readable structured data (2 stars); c) Data uses a non-proprietary format (3 stars); d) Data uses URIs to identify things (4 stars); e) Data is linked to other data to provide context (5 stars). Some examples of large LOD datasets are DBpedia[2], Wikidata[3], and GeoNames[4].

In this paper, we present how we have managed to extract all the information related to Greek universities, departments, study programs, courses and textbooks used in the courses, from the Evdoxus web site and then convert it into an open KG in RDF that can be used to generate several interesting reports and statistical analyses, through SPARQL queries. The KG is backed by a simple ontology which is aligned with some well-known ontologies. The extraction / conversion application has been developed using SWI-Prolog's XPath and Semantic Web libraries. The Evdoxus KG encompasses the LOD initiative (5 stars) by linking University instances with their corresponding DBpedia entries, employing the Wikipedia search engine and the DBpedia SPARQL endpoint.

In the rest of the paper, we briefly review related work in Section 2 and we present the KG generation methodology in Section 3. Section 4 presents various competency questions we have used to build the ontology and evaluate the KG, as well as some other interesting SPARQL queries. Finally, Section 5 concludes the paper with some future research directions.

2 RELATED WORK

There are several ontologies about Academia, including education, research, and publications (Stancin, Poscic, & Jaksic, 2020). The most influential ones can be found at the Linked Open Vocabularies (LOV) repository[5], a popular catalogue of reusable ontologies (Vandenbussche, Atemezing, Poveda-Villalón, & Vatant, 2017) that currently contains 782 ontologies in total. At LOV, we found 16 ontologies tagged with the keyword "Academy". However, only 6 of them deal with purely educational concepts, whereas the rest deal with research and publication activities. Out of the 6, we excluded those that do not have classes / properties in the English language (e.g., Education Ontology[6]), or they deal with country-specific educational systems (e.g., EduProgression Ontology[7]). In the following paragraphs, we briefly review the remaining ones, commenting also on how much they cover the needs of the Evdoxus KG.

The VIVO core ontology[8] focuses on the general domain of academia and researchers and their research-related activities and relationships. The ontology is independent of knowledge or creative domain (Corson-Rikert, et al., 2012). The VIVO ontology is mainly focused on the research and publications activities of a university and not education, so it partly covers the structure of the University and Courses / Modules, without however gluing Courses / Modules into Study Programs and also not hierarchically connecting the Academic Units to each other. For Books, it adheres to the BIBO ontology[9].

The Academic Institution Internal Structure Ontology (AIISO)[10] provides classes and properties to describe the internal organizational structure of an academic institution. AIISO captures the part of the Evdoxus ontology that deals with the structure of Universities, Departments, Study Programs and Courses. However, it does not cover Books and their relationship to Courses / Modules and some properties for Courses / Modules are also missing.

The Teaching Core Vocabulary Specification (TEACH)[11] is a lightweight vocabulary providing terms to enable teachers to relate things in their courses together. TEACH is based on practical requirements set by providing seminar and course descriptions as Linked Data. The TEACH ontology covers mostly the lower classes of the Evdoxus ontology, namely Courses/Modules and teaching material (e.g., Books). However, it does not cover the University organization in Universities / Departments.

Finally, the ReSIST Courseware Ontology[12] represents the various educational courses and resources

[2] https://www.dbpedia.org/
[3] https://www.wikidata.org/
[4] http://www.geonames.org/
[5] https://lov.linkeddata.es/dataset/lov
[6] https://schema.edu.ee/
[7] http://ns.inria.fr/semed/eduprogression/
[8] http://vivoweb.org/ontology/core
[9] https://www.dublincore.org/specifications/bibo/bibo/
[10] https://vocab.org/aiiso/schema-20080925.html
[11] https://lov.linkeddata.es/dataset/lov/vocabs/teach
[12] https://lov.linkeddata.es/dataset/lov/vocabs/crsw

within the ReSIST project. Its focus is mainly on the internal structure of a course / module and the activities / courseware involved, so it has too little overlap with the Evdoxus KG; thus, it was excluded from being aligned with the Evdoxus ontology.

One quite popular educational ontology, not included at LOV, is the Bowlogna ontology[13] that models an academic setting as proposed by the Bologna reform (Demartini, Enchev, Gapany, & Cudré-Mauroux, 2013), that gave rise to new administrative procedures at European universities and new concepts for the description of curricula. This ontology partly covers the concepts of the Evdoxus ontology but scarcely its properties. Nevertheless, we included it in the alignment process due to its impact in the development of educational ontologies.

Besides domain-dependent academic ontologies, there are several general-purpose ones, such as Schema.org, DBpedia and Wikidata, that include several classes / properties that may cover the Evdoxus ontology. Based on their popularity, we selected the above three to discuss here and include in the alignment process.

Schema.org is a collaborative, community activity that maintains schemas for structured data on the web. The Schema.org vocabulary[14] currently consists of 797 types and 1457 properties, covering entities, relationships between entities and actions, and can easily be extended. It is used by over 10 million sites to markup web pages and email messages. Many applications from Google, Microsoft, and others already use these vocabularies. The Schema.org vocabulary is so rich that actually covers the Evdoxus ontology even more than the aforementioned domain-specific ontologies. However, its alignment coverage is loose, meaning that it does not provide many equivalences, but rather subclasses and sub-properties.

The DBpedia Ontology[15] is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. The ontology currently covers 685 classes which form a subsumption hierarchy and are described by 2,795 different properties, being also one of the biggest general-purpose ontologies. However, it only minimally covers the Evdoxus ontology; we have included it in the alignment mainly because the DBpedia dataset is the central hub of the Linked Open Data cloud and we have linked the University instances of the Evdoxus KG with their DBpedia counterparts.

Wikidata (Vrandečić & Krötzsch, 2014) is the central data management platform of Wikipedia, capturing structured data on several subject domains, managing, among others, the information underlying Wikipedia and other Wikimedia projects. By the efforts of thousands of volunteers, the project has produced a large, open knowledge base with many interesting applications. The data is highly interlinked and connected to many other datasets. The Wikidata repository consists mostly of items and statements about these items. Items are used "to represent all the things in human knowledge, including topics, concepts, and objects", and are given a unique identifier, a label, and a description. Statements are used "for recording data about an item", and "consist of (at least) one property-value pair"; they serve to "connect items to each other, resulting in a linked data structure". To organize Wikidata's content, some items (classes) are used to classify other items through the "instance of" property. Further, classes are related through the "subclass of" taxonomic property, defining thus hierarchies of classes, from more general to more specific ones.

The Wikidata ontology (called WikiProject Ontology[16]) consists of a few upper-level classes and properties that aim to (a) support a broad semantic interoperability between notable ontologies like DOLCE, BFO, SUMO, Lemon, RDA, etc.; and (b) build consensus around the main branches of Wikidata core concept tree and how they relate to each other.

Most Academic KGs relate to scientific research and publications, such as the Microsoft Academic KG (Färber, 2019) or Open Research KG (Jaradeh, et al., 2019). There are few KGs related to the edu-cational aspect of academia, dealing with teaching and classroom resources, education management and educational technologies (Abu-Salih, 2021). However, none of the ones reported in the above survey relate to the content and the size of the Ev-doxus KG.

To the best of our knowledge there are no services like Evdoxus nor any datasets similar to the Evdoxus KG. Searching at the LOD Cloud repository[17], we have found 12 datasets related to the university domain, 4 of them being University reading lists[18], which is quite close to the Evdoxus KG. The largest of them is supposed to consist of 4 million triples. However, none of these LOD datasets is accessible anymore.

[13] https://gist.github.com/lsarni
[14] https://schema.org/docs/schemas.html
[15] http://mappings.dbpedia.org/server/ontology/classes/
[16] https://www.wikidata.org/wiki/Wikidata:WikiProject_Ontology
[17] https://lod-cloud.net
[18] https://lod-cloud.net/datasets?search=reading

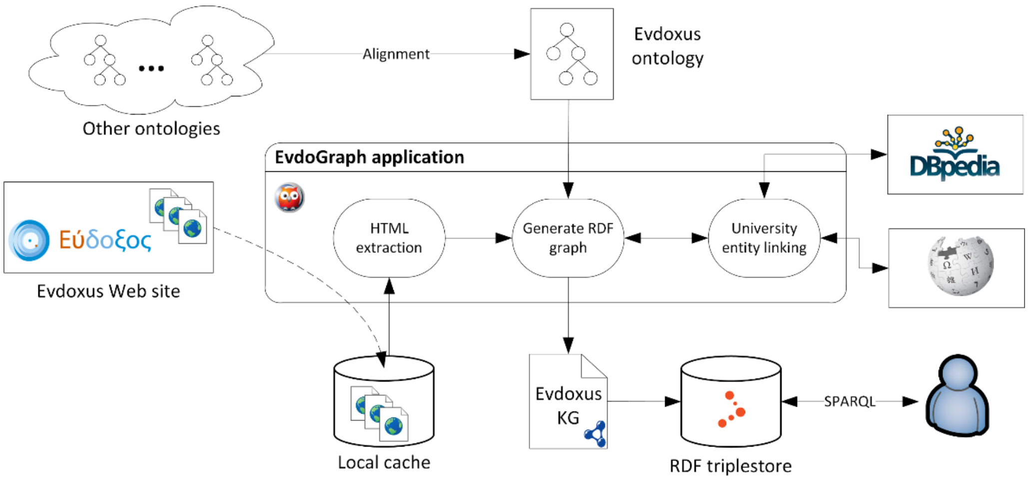

Figure 1: Knowledge Graph generation workflow.

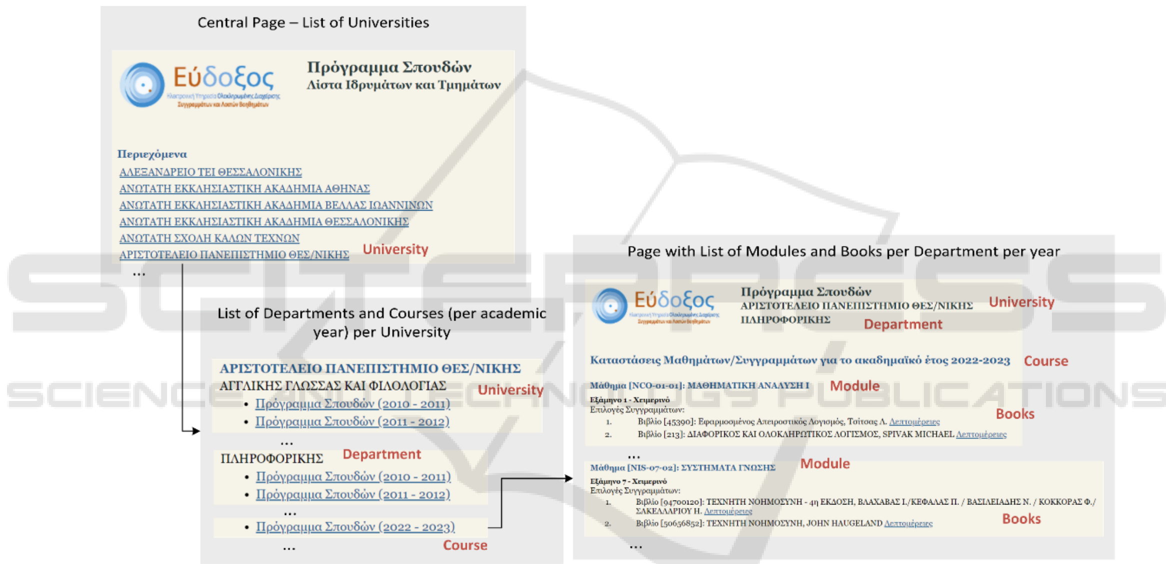

Figure 2: Structure of the Evdoxus site.

Furthermore, the data portal of the Greek Government contains one dataset (CSV and JSON) about statistics of requests (from students) and deliveries for books made through the Evdoxus system[19], without any details about Departments, Courses, Modules, and specific books, whereas the old data portal[20] does contain some partial older datasets, in tabular format.

3 METHODOLOGY

In Figure 1 we can see the workflow of our methodology for creating the Evdoxus KG. Central role plays the EvdoGraph application which has been developed [21] in SWI-Prolog (Wielemaker, Schrijvers, Triska, & Lager, 2012). The web pages that constitute the Evdoxus web site repository (Figure 2) are first cached in the local disk, to speed up the graph generation process later. This is done by extracting from the entry page[22], which contains all the Greek Universities (46), all their departments (732) and all their study programs throughout the years Evdoxus operates (13 years, from 2010 to 2023), all the linked pages, namely the pages that contain all the courses of the curricula (per year), plus the textbooks suggested per course, a total of 9516 pages.

[19] https://data.gov.gr/datasets/grnet_eudoxus/
[20] https://repository.data.gov.gr/dataset?tags=ΕΥΔΟΞΟΣ
[21] https://github.com/nbassili/EvdoGraph
[22] https://service.eudoxus.gr/public/departments

Then, all these HTML pages are used for extracting information about Universities, Departments, Study Programs per year (called “courses” in Evdoxus), Courses per Study Program (called “modules”), and finally, Books (textbooks) suggested per module. Notice that only the printed books are extracted and not any additionally suggested eBooks.

The extraction is based on the http, sgml and xpath libraries of SWI-Prolog. Notice that since these pages are automatically generated from the Evdoxus internal DB, they follow a consistent template and thus their parsing and information extraction is 100% successful. This information is used for generating the KG, in RDF, inside SWI-Prolog’s knowledge base, using the semweb library of SWI-Prolog.

The KG is structured according to the Evdoxus ontology we have developed[23], in RDF Schema. The ontology is minimal, with 7 classes and 10 properties (Table 1), inspired by the hierarchical information structure of the Evdoxus site (Figure 2). Figure 3 shows the class hierarchy of the Evdoxus ontology, while Figure 4 shows the relationships between the concrete ontology classes. Notice that classes AcademicEntity and LearningEntity are abstract ones, i.e., they do not have direct instances but rather serve as property placeholders for inheritance purposes.

The University class instances are linked to their corresponding DBpedia entries (where possible), via a methodology that combines searching in Wikipedia, using the search API, and DBpedia, using its SPARQL endpoint, discussed in Section 3.1.

The Evdoxus ontology has been aligned with various external ontologies for interoperability purposes (VIVO, AIISO, teach, Bowlogna, Schema.org, DBpedia, and Wikidata). The alignments are shown in Table 1, both for classes (using rdfs:subClassOf or owl:equivalentClass) and properties (using rdfs:subPropertyOf, owl:equivalentProperty or owl:inverseOf).

Table 2 shows alignment statistics for the various ontologies. By “strict” alignments, we denote alignments through owl:equivalentClass and owl:equivalentProperty, whereas by “loose” alignments, we mean alignments through rdfs:subClassOf, rdfs:subPropertyOf and owl:inverseOf. It is evident that VIVO provides the most strict alignments, whereas Wikidata, Schema.org and AIISO provide the most total aligments.

In Listing, we quote an example of the Evdoxus KG that includes a University (Aristotle University of Thessaloniki), a Department (Informatics), a Course (BSc of Informatics, 2022-2023), a Module (Knowledge Systems) and one of the suggested Books (Vlahavas, Kefalas, Bassiliades, Kokkoras, & Sakellariou, 2020).

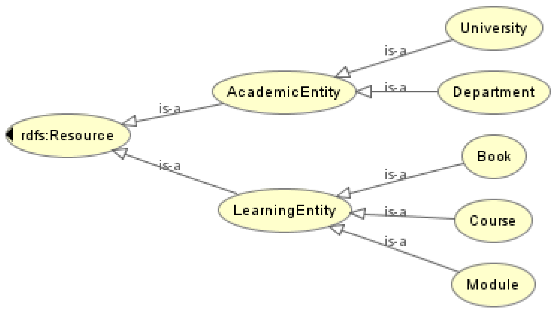

Figure 3: Class hierarchy of the Evdoxus ontology.

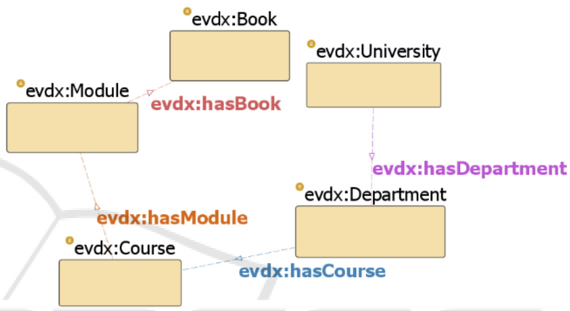

Figure 4: Relationships between concrete classes.

The whole Evdoxus KG is, finally, stored at a triplestore (GraphDB) and can be found and queried here[24]. Table 3 shows statistics about the Evdoxus KG. Notice that there might be more books registered at Evdoxus; however, in the Evdoxus KG we only include those that are being used in some module. Books are uniquely identified via their Evdoxus code. Also, modules are uniquely identified within Departments via their code. However, there is no single identification scheme across departments.

The entire extraction (from the local cache) and generation of the KG takes ~230 sec on a i7-11700 @ 2.50GHz PC, with 16GB memory, using SWI-Prolog (64 bits, v. 9.0.3), whereas storing the KG at the SSD takes ~28 sec (~3.9 million triples). Loading the KG at GraphDB, using RDFS Plus semantics (total ~17 million triples), takes ~9 min.

[23] https://w3id.org/evdoxus

[24] http://lod.csd.auth.gr:7200/sparql; repository Evdoxus.

Table 1: Alignments of the Evdoxus Ontology with external ontologies.

Evdoxus Ontology		Alignments
Classes	evdx:AcademicEntity	-
	evdx:LearningEntity	-
	evdx:University	owl:equivalentClass vivo:University, dbo:University, wikidata:Q3918; rdfs:subClassOf aiiso:Institution, schema:CollegeOrUniversity
	evdx:Department	owl:equivalentClass vivo:AcademicDepartment, aiiso:Department, bow:Department, wikidata:Q2467461; rdfs:subClassOf schema:School
	evdx:Course	owl:equivalentClass aiiso:Programme, teach:StudyProgram, bow:Study_Program; rdfs:subClassOf schema:EducationalOccupationalProgram, wikidata:Q207137
	evdx:Module	owl:equivalentClass vivo:Course, aiiso:Course, teach:Course, schema:Course, bow:Module, wikidata:Q600134
	evdx:Book	owl:equivalentClass bibo:Book, schema:Book, dbo:Book, wikidata:Q571; rdfs:subClassOf teach:Material
Properties	evdx:hasDepartment	rdfs:subPropertyOf aiiso:organization, schema:department, wikidata:P527
	evdx:hasCourse	rdfs:subPropertyOf aiiso:teaches; owl:inverseOf schema:provider, wikidata:P137
	evdx:hasModule	rdfs:subPropertyOf aiiso:knowledgeGrouping, wikidata:P527; owl:inverseOf teach:studyProgram; owl:equivalentProperty schema:hasCourse
	evdx:hasBook	rdfs:subPropertyOf teach:reading; owl:inverseOf wikidata:P366
	evdx:hasCode	rdfs:subPropertyOf aiiso:code, wikidata:P3295
	evdx:hasURL	owl:equivalentProperty vCard:hasURL, schema:url; rdfs:subPropertyOf wikidata:P2699
	evdx:name	owl:equivalentProperty vcard:hasOrganizationName; rdfs:subPropertyOf foaf:name, bow:hasName, wikidata:P2561
	evdx:semester	rdfs:subPropertyOf teach:academicTerm
	evdx:title	owl:equivalentProperty vCard:title; rdfs:subPropertyOf teach:hasTitle, bow:hasName, wikidata:P1476
	evdx:year	rdfs:subPropertyOf bow:beginsToApplyOnDate, schema:startDate, dbo:startYear, wikidata:P571

Listing 1: Excerpt from the Evdoxus KG.

```
evdx:university_8
        a evdx:University ;
        evdx:hasDepartment ... ,   evdx:dept_1596 ,   ... ;
        evdx:name "ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣ/ΝΙΚΗΣ" ;
        owl:sameAs dbr:Aristotle_University_of_Thessaloniki ,
                dbpedia-el:Αριστοτέλειο_Πανεπιστήμιο_Θεσσαλονίκης .
evdx:dept_1596
        a evdx:Department ;
        evdx:hasCode "1596" ;
        evdx:hasCourse ... ,
                evdx:course_1596_2021 , evdx:course_1596_2022 ;
        evdx:name "ΠΛΗΡΟΦΟΡΙΚΗΣ" .
evdx:course_1596_2022
        a evdx:Course ;
        evdx:hasModule ... ,
                evdx:module_1596_2022_61 ,
                ... ;
        evdx:hasURL "https://service.eudoxus.gr/public/departments/courses/1596/2022"^^xsd:anyURI ;
        evdx:title "Πρόγραμμα Σπουδών (2022 - 2023)" ;
        evdx:year 2022 .
evdx:module_1596_2022_61
        a evdx:Module ;
        evdx:hasBook evdx:book_50656852 ,
```

```
            evdx:book_94700120 ;
        evdx:hasCode "NIS-07-02" ;
        evdx:semester 7 ;
        evdx:title "ΣΥΣΤΗΜΑΤΑ ΓΝΩΣΗΣ" .
evdx:book_94700120
        a evdx:Book ;
        evdx:hasCode "94700120" ;
        evdx:hasURL "https://service.eudoxus.gr/search/#a/id:94700120/0"^^xsd:anyURI ;
        evdx:title "ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ - 4η ΕΚΔΟΣΗ, ΒΛΑΧΑΒΑΣ Ι./ΚΕΦΑΛΑΣ Π. / ΒΑΣΙΛΕΙΑΔΗΣ Ν. / ΚΟΚΚΟΡΑΣ Φ./ ΣΑΚΕΛΛΑΡΙΟΥ Η." .
```

Algorithm 1: Function link-university.

```
Function link-university(UniversityName)
      Result := xpath(Wikipedia-search(UniversityName), //ul[@class='mw-search-results']/li[1])
      AltNames := { xpath(Result, //a/@title),
                    xpath(Result, //span[@class='searchalttitle']/a[@class='mw-redirect}/@title) }
      ELWikiURL := xpath(Result, //a/@href)
      ELDBpediaURI := deref(ELWikiURL)
      ENDBpediaURI := find-en-dbpedia(ELWikiURL, ELDBpediaURI)
      AltNames := AltNames ∪ find-alt-names(ENDBpediaURI)
      if max_{x ∈ AltNames} (str_dist(x, UniversityName)) > 0.75
            then Return({ELDBpediaURI, ENDBpediaURI })
            else Return(∅)
```

Algorithm 2: Function find-en-dbpedia.

```
Function find-en-dbpedia(ELWikiURL, ELDBpediaURI)
      R := Execute-sparql("https://dbpedia.org/sparql",
            'select ?u where { ?u rdf:type dbo:University ;  owl:sameAs <ELDBPediaURL> .  }')
      if R=∅
            then R := deref(xpath(ELWikiURL, //a[@class='interlanguage-link-target' and @hreflang='en']//@href))
      Return(R)
```

Algorithm 3: Function find-alt-names.

```
Function find-alt-names(ENDBpediaURI)
      If ENDBpediaURI≠∅
            then Result := Execute-sparql(("https://dbpedia.org/sparql",
                  'select distinct str(?n) where { <ENDBPediaURI> (rdfs:label |foaf:name | dbp:nativeName) ?n . }')
            else Result := ∅
      Return(Result)
```

Table 2: Alignment statistics.

Ontology	Alignments		
	Strict	Loose	Total
VIVO	7	0	7
AIISO	3	6	9
TEACH	2	5	7
Bowlogna	3	3	6
Schema.org	4	6	10
DBpedia	2	1	3
Wikidata	4	10	14

Table 3: Evdoxus KG statistics.

Class	Instance count
University (linked to DBpedia)	46 (43)
Department	732
Course	9516
Module	535143
Book	40529

Table 4: Description of auxiliary functions.

Function	Description
Wikipedia-search(InputString)	Returns the HTML page contents of the Wikipedia search results
xpath(HTMLPage,Xpath-expr)	Returns the content of the HTML page according to the Xpath-expr
deref(WikiURL)	Replaces “http://el.wikipedia.org/wiki/” with “http://el.dbpedia.org/resource/”, or “https://en.wikipedia.org/wiki/” with “http://dbpedia.org/resource/”

3.1 DBpedia Linking

All the University class instances of the KG are linked to their DBpedia counterparts, if possible. The pseudocode for our linking methodology is shown at Algorithm 1, Algorithm 2 and Algorithm 3. Table 4 describes the input/output of some trivial auxiliary predicates used in the previous algorithms. Our methodology is based on previous experience in linking University instances with DBpedia (Bassiliades, 2014).

Initially, the full University name is used to search at Wikipedia the corresponding lemma, using its search API (Line 2 at Algorithm 1, see example[25] for Aristotle University of Thessaloniki). Then, the first returned result is extracted from the HTML result page (using Xpath, Lines 2-5). The URL of the result Wikipedia lemma page is de-referenced to its corresponding DBpedia URI (Line 6). Since University names are in Greek, the resulting lemmas are from the Greek Wikipedia, and the corresponding DBpedia instances are from the Greek DBpedia[26]. Using the SPARQL endpoint of the English DBpedia, the corresponding URI at the English DBpedia is retrieved (using owl:sameAs, Line 7 at Algorithm 1 and Lines 2-3 at Algorithm 2). Alternatively, the URI at the English DBpedia can be discovered by de-referencing the URI of the English Wikipedia, which is usually “hidden” in title of the Greek Wikipedia lemma (p-lang-btn, Lines 4-5 at Algorithm 2). Both URIs are linked to the Evdoxus KG University entry, as shown in the example at Listing .

All the above are the straightforward steps, which usually discover the correct Wikipedia lemmas and DBpedia entries. In order to check if the retrieved information is the correct one, the title of the retrieved Wikipedia entry is compared with the University name extracted from the Evdoxus site, using the string-matching metric of (Stoilos, Stamou, & Kollias, 2005), with a threshold of 0.75 (Line 9 at Algorithm 1), discovered via experimentation. If the matched lemma / entity has a smaller string similarity, then alternative University names for comparison are sought. One such source is alternative or redirected Wikipedia lemmas; a few years ago, Greek Universities have undergone a partial reformation (Polytechnics either reformed to or merged into universities or they have joined with existing Universities). In this case, alternative or redirected Wikipedia lemma titles contain the older University title (Line 4 at Algorithm 1), which is still mentioned at the Evdoxus site. Another source of alternative names is DBpedia itself; the dereferenced DBpedia URI is queried for alternative University titles (Line 8 at Algorithm 1 and Algorithm 3). If none of the above alternative names has a similarity score greater than the threshold, then the University entity is not linked to DBpedia (Lines 9-11 at Algorithm 1). This occurs for only 3 out of the 46 University instances (Table 3). In general, our methodology achieves 100% precision, since all the linked DBpedia entries are correct, and 100% recall, since the non-linked University entries do not have a DBpedia/Wikipedia entry. The confirmation was performed via manual inspection.

4 COMPETENCY QUESTIONS

One of the main reasons for building the Evdoxus Ontology and KG was to be able to generate various reports and statistics concerning the Universities, Departments and Modules (Courses) that a certain textbook is used, either in one academic year or throughout the years, alone or in comparison to other competitive textbooks, etc. Initially, these reports / statistics were generated (for personal use) through a Prolog application called EvdoStats[27]. The predicates that generate these reports / statistics play the role of competency questions that the Evdoxus Ontology and KG should be able to answer.

Table 5 shows some the competency questions handled by the Evdoxus KG.

[25] https://el.wikipedia.org/w/index.php?fulltext=1&search=**ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣ/ΝΙΚΗΣ**&ns0=1

[26] http://el.dbpedia.org/

[27] https://github.com/nbassili/EvdoStats

Table 5: Competency questions.

CQ1	Return all modules that use the book, for a specific year, along with the department and the university.
CQ2	Return how many modules and all module names (in a string), that the book is used, for a specific year, along with the department and the university, grouped by the department.
CQ3	Return how many modules, departments and universities use the book, for a specific year.
CQ4	Return how many modules, departments and universities use the book, per year, for a range of years.
CQ5	Which departments (including details about university/modules) additionally use the book in a subsequent academic year, compared to a previous one.
CQ6	Which universities (including details about departments/modules) additionally use the book in a subsequent academic year, compared to a previous one.
CQ7	Which modules (including details about departments/universities) additionally use the book in a subsequent academic year, compared to a previous one.
CQ8	Return comparison details and statistics for multiple books, for a specific academic year.
CQ9	Which modules (including details about departments/universities) use only the first book and not the second, for a specific academic year.
CQ10	Which departments (including details about university/modules) use only the first book and not the second, for a specific academic year.
CQ11	Which universities (including details about departments/modules) use only the first book and not the second, for a specific academic year.

Table 6: Results for competency question CQ4.

?year	?Univs	?Depts	?Mods
2019	20	42	59
2020	21	49	73
2021	20	54	83
2022	21	58	83

Table 7: Results for competency question CQ8.

?Book	?Univs	?Depts	?Mods
"94700120"	21	58	83
"102070469, 13909"	19	56	82

Listing 2 and Listing 3 show the SPARQL queries implementing the competency questions CQ4 and CQ8, respectively, whereas Table 6 and Table 7 show the corresponding results. Notice that there are situations where multiple editions of the same textbook are available through Evdoxus. To this end, in order to correctly calculate book statistics, the VALUES construct is used to aggregate different editions of the same book into a single result. E.g. at Listing 2, the book codes "94700120" and "12867416" denote the 4th (most recent) and the 3rd editions, respectively, of the book at reference (Vlahavas, Kefalas, Bassiliades, Kokkoras, & Sakellariou, 2020), whereas at Listing 3, the book codes "102070469" and "13909" correspond to the Greek translation of the 4th and 2nd editions, respectively, of the book at reference (Russell & Norvig, 2020). The SPARQL queries for all the competency questions can be found at the GitHub repository for EvdoGraph[28].

In addition to the competency questions, which are inspired by the EvdoStats application and revolve around reports and statistics about a single or a couple of books, the Evdoxus KG can provide answers to more general queries. For example, in Listing 4 two such queries are shown; the left one (AQ1) returns the names of the departments (and the name of their university) that are not "active" at Evdoxus during a specific academic year, i.e., they have a course that does not have any modules. The latter is true mainly for departments that have been discontinued in previous years, but the Evdoxus site still includes them.

The query on the right (AQ2) reports the top-6 books, concerning the number of departments and modules that they are used in, sorted in descending order, first on departments and then on modules. Results are shown in Table 8. Such reports are not easy to be generated by the EvdoStats application because the latter is based on searching for specific book codes in the course pages and not on collecting information about all books.

[28] https://github.com/nbassili/EvdoGraph

Listing 2: SPARQL query for competency question CQ4.

```
PREFIX evdx: <https://w3id.org/evdoxus#>
select ?year (count(DISTINCT ?u) as ?Univs) (count(DISTINCT ?d) as ?Depts) (count(DISTINCT ?m) as ?Mods)
where {
    ?s a evdx:Book .
    VALUES ?code { "94700120" "12867416" }
    ?s evdx:hasCode ?code .
    ?m a evdx:Module ;  evdx:hasBook ?s .
    ?c a evdx:Course ;  evdx:year ?year.
    FILTER ((?year>=2019) && (?year<2023)) .
    ?c evdx:hasModule ?m .
    ?d a evdx:Department ;  evdx:hasCourse ?c .
    ?u a evdx:University ;  evdx:hasDepartment ?d .
} group by ?year
    order by ?year
```

Listing 3: SPARQL query for competency question CQ8.

```
PREFIX evdx: <https://w3id.org/evdoxus#>
select (group_concat(DISTINCT ?code;separator=", ") as ?Book) (count(DISTINCT ?u) as ?Univs) (count(DISTINCT
?d) as ?Depts) (count(DISTINCT ?m) as ?Mods) where {
    ?s a evdx:Book .
    VALUES (?bcount ?code) { (1 "94700120") (1 "12867416") (2 "102070469") (2 "13909")}
    ?s evdx:hasCode ?code .
    ?m a evdx:Module ;  evdx:hasBook ?s .
    ?c a evdx:Course ;  evdx:year 2022 ;  evdx:hasModule ?m .
    ?d a evdx:Department ;  evdx:hasCourse ?c .
    ?u a evdx:University ;  evdx:hasDepartment ?d .
} group by ?bcount
```


Listing 4: Additional SPARQL queries.

AQ1

```
PREFIX evdx: <https://w3id.org/evdoxus#>
select ?dn ?un where {
    ?c a evdx:Course ; evdx:year 2022 .
    ?d evdx:hasCourse ?c ; evdx:name ?dn .
    ?u evdx:hasDepartment ?d ; evdx:name ?un
.
    FILTER NOT EXISTS {
        ?c evdx:hasModule ?m .
    }
}
```

AQ2

```
PREFIX evdx: <https://w3id.org/evdoxus#>
select ?book (count(DISTINCT ?d) as ?Depts) (count(DISTINCT
?m) as ?Mods) where {
    ?s a evdx:Book ;  evdx:title ?book .
    ?m a evdx:Module ;  evdx:hasBook ?s .
    ?c a evdx:Course ;  evdx:year 2022 ;  evdx:hasModule ?m .
    ?d a evdx:Department ;  evdx:hasCourse ?c .
} group by ?s ?book
    order by desc(?Depts) desc(?Mods) limit 6
```

Table 8: Results for additional question AQ2.

?book	?Depts	?Mods
THOMAS ΑΠΕΙΡΟΣΤΙΚΟΣ ΛΟΓΙΣΜΟΣ, [George B. Thomas], Jr., Joel Hass, …	79	142
Εισαγωγή στην πληροφορική, Evans Alan, Martin Kendall, Poatsy Mary Anne (Συγγρ.) …	74	96
Πώς γίνεται μια επιστημονική εργασία;, Ζαφειρόπουλος Κώστας	70	102
Επιχειρηματικότητα και μικρές Επιχειρήσεις 2η Έκδοση, David Deakins, Mark Freel	62	73
ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ - 4η ΕΚΔΟΣΗ, ΒΛΑΧΑΒΑΣ Ι./ΚΕΦΑΛΑΣ Π./ ΒΑΣΙΛΕΙΑΔΗΣ …	58	83
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ, BEHROUZ FOROUZAN	58	72

5 CONCLUSIONS

In this paper, we have presented the methodology for generating the Evdoxus Knowledge Graph, that consists of information about the structure of Greek Universities, including their Departments, Study Programs, Courses, and the textbooks that are used and freely provided to the undergraduate students. This information was extracted from the Evdoxus site, an online system for the management of the total ecosystem for the free provision of textbooks to the undergraduate students at the Greek Universities. The extraction / conversion application, called EvdoGraph, has been developed using SWI-Prolog. The KG is using the vocabulary of a simple ontology we have developed, which has been also aligned with some well-known ontologies for interoperability. Moreover, the KG fully endorses the Linked Open Data initiative by linking University class instances with their corresponding DBpedia entries. The final result is a quite rich KG with almost 4 million explicit triples that is freely available through a SPARQL endpoint.

The possible uses for the KG are countless. In the paper we have demonstrated several competency questions that can be answered via SPARQL queries that generate detailed reports or aggregate statistical analyses concerning the "performance" (popularity among the Greek Universities) of either one book or several books in comparison. More ideas for using the KG could be for marketing purposes, i.e., publishers could have an instant clear picture of the University market in order to strategically decide for new books or promotion targets, or faculty researchers could analyse the Greek Higher Education landscape, i.e., analyse what kind of courses are taught at various disciplines, or compare study programs at different Universities / Departments. And, of course, according to (European Data Portal, 2020) opening up official information can support technological innovation and economic growth by enabling third parties to develop new kinds of digital applications and services.

Ideas for future work could include the more fine-grained treatment of textbooks, as currently their title is actually the whole citation of the book. This will allow statistics about authors and publishers, as well as possibility to further link the KG to external bibliographic LOD datasets. Another option would be to link Study programs and modules to their syllabus description at various University repositories or open data APIs, such as the one of the Aristotle University of Thessaloniki[29]. Finally, the University / Department instances could be linked to more LOD datasets, such as Wikidata, even though this can already be indirectly (albeit partially) provided via DBpedia's interlinking to several other LOD datasets[30].

[29] https://ws-ext.it.auth.gr/swagger/

[30] http://wikidata.dbpedia.org/services-resources/interlinking

REFERENCES

Abu-Salih, B. (2021). Domain-specific knowledge graphs: A survey. *Journal of Network and Computer Applications, 185*. doi:10.1016/j.jnca.2021.103076

Bassiliades, N. (2014). Collecting University Rankings for Comparison Using Web Extraction and Entity Linking Techniques. In *ICT in Education, Research and Industrial Applications, CCIS* (Vol. 469, pp. 23-46). Springer-Verlag. doi:10.1007/978-3-319-13206-8_2

Berners-Lee, T. (2009). *Linked data*. Retrieved from WWW Consortium: http://www.w3.org/DesignIssues/LinkedData.html

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *International journal on semantic web and information systems, 5*(3), pp. 1-22. doi:10.4018/jswis.2009081901

Corson-Rikert, J., Mitchell, S., Lowe, B., Rejack, N., Ding, Y., & Guo, C. (2012). The VIVO Ontology. In *VIVO, Synthesis Lectures on Data, Semantics, and Knowledge* (pp. 15–33). Springer, Cham. doi:10.1007/978-3-031-79435-3_2

Demartini, G., Enchev, I., Gapany, J., & Cudré-Mauroux, P. (2013). The Bowlogna ontology: Fostering open curricula and agile knowledge bases for Europe's higher education landscape. *Semantic Web, 4*(1), pp. 53-63. doi:10.3233/SW-2012-0064

European Data Portal. (2020). *The Economic Impact of Open Data: Opportunities for value creation in Europe.* European Union. doi:10.2830/63132

Färber, M. (2019). The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data. *ISWC 2019, LNCS. 11779*, pp. 113-129. Springer. doi:10.1007/978-3-030-30796-7_8

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G., Gutierrez, C., . . . Zimmermann, A. (2022). Knowledge graphs. *ACM Computing Surveys, 54*(4), pp. 1–37. doi:10.1145/3447772

Jaradeh, M. Y., Oelen, A., Farfar, K. E., Prinz, M., D'Souza, J., Kismihók, G., . . . Auer, S. (2019). Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. *K-CAP 2019* (pp. 243-246). ACM. doi:10.1145/3360901.3364435

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach, 4th edition.* Pearson.

Stancin, K., Poscic, P., & Jaksic, D. (2020). Ontologies in education - state of the art. *Education and Information*

Technologies, 25(6), pp. 5301-5320. doi:10. 1007/s10639-020-10226-z
Stoilos, G., Stamou, G., & Kollias, S. (2005). A String Metric for Ontology Alignment. *ISWC 2005* (pp. 624-637). Springer. doi:10.1007/11574620_45
Vandenbussche, P.-Y., Atemezing, G. A., Poveda-Villalón, M., & Vatant, B. (2017). Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. *Semantic Web, 8*(3), pp. 437-452. doi:10.3233/SW-160213
Vlahavas, I., Kefalas, P., Bassiliades, N., Kokkoras, F., & Sakellariou, I. (2020). *Artificial Intelligence, 4th edition.* Thessaloniki: University of Macedonia Press.
Vrandečić, D., & Krötzsch, K. (2014, October). Wikidata: a free collaborative knowledgebase. *Communications ACM, 10*, pp. 78–85. doi:10.1145/2629489
Wielemaker, J., Schrijvers, T., Triska, M., & Lager, T. (2012). SWI-Prolog. *Theory and Practice of Logic Programming, 12*(1-2), pp. 67-96. doi:10.1017/ S1471068411000494.