The Computer Science Ontology: A Comprehensive Automatically-Generated Taxonomy of Research Areas

The Computer Science Ontology: A Comprehensive Automatically-Generated Taxonomy of Research Areas

Angelo A. Salatino[1], Thiviyan Thanapalasingam[1], Andrea Mannocci[1], Aliaksandr Birukou[2], Francesco Osborne[1], Enrico Motta[1]

[1]Knowledge Media Institute, The Open University, MK7 6AA, Milton Keynes, UK
{firstname.lastname}@open.ac.uk
[2]Springer-Verlag GmbH, Tiergartenstrasse 17, 69121 Heidelberg, Germany
aliaksandr.birukou@springer.com

Abstract. Ontologies of research areas are important tools for characterising, exploring, and analysing the research landscape. Some fields of research are comprehensively described by large-scale taxonomies, e.g., MeSH in Biology and PhySH in Physics. Conversely, current Computer Science taxonomies are coarse-grained and tend to evolve slowly. For instance, the ACM classification scheme contains only about 2K research topics and the last version dates back to 2012. In this paper, we introduce the Computer Science Ontology (CSO), a large-scale, automatically generated ontology of research areas, which includes about 14K topics and 162K semantic relationships. It was created by applying the Klink-2 algorithm on a very large dataset of 16M scientific articles. CSO presents two main advantages over the alternatives: i) it includes a very large number of topics that do not appear in other classifications, and ii) it can be updated automatically by running Klink-2 on recent corpora of publications. CSO powers several tools adopted by the editorial team at Springer Nature and has been used to enable a variety of solutions, such as classifying research publications, detecting research communities, and predicting research trends. To facilitate the uptake of CSO, we have also released the *CSO Classifier*, a tool for automatically classifying research papers, and the *CSO Portal*, a web application that enables users to download, explore, and provide granular feedback on CSO. Users can use the portal to navigate and visualise sections of the ontology, rate topics and

relationships, and suggest missing ones. The portal will support the publication of and access to regular new releases of CSO, with the aim of providing a comprehensive resource to the various research communities engaged with scholarly data.

Keywords: Scholarly Data, Ontology Learning, Bibliographic Data, Scholarly Ontologies.

1 Introduction

Ontologies have proved to be powerful solutions to represent domain knowledge, integrate data from different sources, and support a variety of semantic applications [1–5]. In the scholarly domain, ontologies are often used to facilitate the integration of large datasets of research data [6], the exploration of the academic landscape [7], information extraction from scientific articles [8], and so on. Specifically, ontologies that describe research topics and their relationships are invaluable tools for helping to make sense of the research dynamics [7], classify publications [3], characterise [9] and identify [10] research communities, and forecast research trends [11] and technology adoption [12].

Some fields of research are well described by large-scale and up-to-date taxonomies, e.g., MeSH in Biology and PhySH in Physics. Conversely, current Computer Science taxonomies are coarse-grained and tend to evolve slowly. For instance, the current version of the ACM classification scheme, containing only about 2K research topics, dates back to 2012, when it superseded its 1998 release.

In this paper, we present the *Computer Science Ontology (CSO)*, a large-scale, granular, and automatically generated ontology of research areas which includes 14,164 topics and 162,121 semantic relationships. CSO was created by applying the Klink-2 algorithm on a dataset of 16M scientific articles, primarily in the field of Computer Science [13]. CSO presents two main advantages over alternative classifications: i) it includes a very large number of topics that do not appear in other classifications, and ii) it can be updated

automatically by running Klink-2 on recent corpora of publications. In particular, its fine-grained representation of research topics is essential for characterising the content of research papers at the granular level at which researchers typically operate. For instance, CSO characterises the Semantic Web according to 34 sub-topics, such as Linked Data, Semantic Web Services, Ontology Matching, SPARQL, OWL, SWRL, and many others. Conversely, the ACM classification simply contains three related concepts: "Semantic web description languages", "Resource Description Framework (RDF)", and "Web Ontology Language (OWL)".

While CSO was officially launched on 10 January 2019 with a joint press release[1] from the Open University and Springer Nature, we have been releasing smaller versions of CSO since 2012 with the aim of fostering reproducibility of relevant research papers [13–15]. However, we did not announce its release and advertised it publicly, as we were aiming at increasing its quality and coverage first. During this period, CSO has supported a range of applications and approaches for community detection, trend forecasting, and paper classification [10, 11, 16]. In particular, CSO powers two tools currently used by the editorial team at Springer Nature: Smart Topic Miner [3] and Smart Book Recommender [17]. The first is a semi-automatic tool for annotating Springer Nature books by means of topics drawn from both CSO and the Springer Nature editorial classification system. The latter is an ontology-based recommender system that suggests the most appropriate books, journals, and conference proceedings in the Springer Nature catalogue, to be marketed at specific scientific events.

[1] Press Release: Springer Nature and The Open University launch a unique Computer Science Ontology (CSO) - https://group.springernature.com/gp/group/media/press-releases/springer-nature-and-the-open-university-launch-a-unique/16386730

We are now publicly releasing the Computer Science Ontology, to ensure that the wider scientific community can take advantage of it and use it as a comprehensive and granular semantic resource to support the development of novel applications in the scholarly domain. To facilitate its uptake, we have also released the *CSO Classifier*, a tool for automatically classifying research papers, and the *CSO Portal*, a web application that enables users to download, explore, and provide granular feedback on CSO. The portal offers three different interfaces for exploring the ontology and visualizing the network of relationships between topics. It also allows users to rate both topics and relationships between topics, as well as suggesting new topics and relationships. This feedback from the community will then be used in the context of generating new versions of CSO. Indeed, we plan to release regularly new versions of CSO, which will incorporate both user feedback as well as new knowledge extracted from the latest scholarly publications.

This paper is an extended version of the work published in [18]. The main novel contributions include:

- A revised version of the ontology that focus on the branches directly under Computer Science and a few other relevant roots.
- The generation of 27,803 *sameAs* and *relatedLink* relationships linking CSO to five Knowledge Bases (DBpedia [19], Wikidata [20], YAGO [21], Freebase [22], Cyc [23]) and to two web sites containing additional information about research topics: Wikipedia and Microsoft Academic[2].
- New features added to the CSO Portal, such as, a tool for finding paths between topics and a dashboard for assisting the CSO *steering committee* in curating the ontology.
- A more comprehensive discussion of the usage of CSO.

[2] Microsoft Academic - https://academic.microsoft.com

- An analysis of the queries to the CSO portal, showing some preliminary trends in terms of geographical distribution of the users and preferred formats.

The paper is structured as follows. In Section 2, we discuss the related work, pointing out the existing gaps. In Section 3, we present the Computer Science Ontology and discuss its generation, the alignment with external resources, and the strategy for updating it. Section 4 describes the CSO Classifier [24], a tool for automatically classifying research articles according to CSO. Section 5 shows both applications and research efforts that make use of CSO. In Section 6, we discuss the CSO Portal and the relevant use cases. Finally, in Section 7 we summarise the main conclusions and outline future directions of work.

2 Related work

Ontologies and taxonomies of research topics can support a variety of crucial tasks, such as integrating heterogenous datasets [6], assisting users in exploring digital libraries [25], producing scholarly analytics [26], and forecasting research dynamics [3, 11]. Since generating these knowledge bases manually requires a large number of experts and is an expensive and lengthy process, in the last years we saw the emergence of several methods for producing them (semi-) automatically from a set of relevant documents. In this section, we will first focus on current available ontologies of research topics and then discuss the approaches to automatically generate them.

2.1 Ontologies and taxonomies of research areas

In the field of Computer Science, the best-known taxonomy is the ACM Computing Classification System[3], developed and maintained by the Association for Computing

[3] The ACM Computing Classification System - http://www.acm.org/publications/class-2012.

Machinery (ACM). However, this taxonomy suffers from several limitations: in particular, it contains only about 2K research topics and it is developed manually. This is an extremely slow and expensive process and, as a result, its last version dates back to 2012. Hence, while the ACM taxonomy has been adopted by many publishers, in practice it lacks both depth and breadth and its releases quickly go out of date due to the rapidly changing nature of the research landscape.

In the field of Physics and Astronomy, the most popular solution used to be the Physics and Astronomy Classification Scheme (PACS)[4], replaced in 2016 by the Physics Subject Headings (PhySH)[5]. PACS used to associate alphanumerical codes to each subject heading to indicate their position within the hierarchy. However, this setup made its maintenance quite complex and the American Institute of Physics (AIP) discontinued it in 2010. Afterwards, the American Physical Society (APS) developed PhySH, a new classification scheme that has the advantage of being crowdsourced with the support of authors, reviewers, editors and organisers of scientific conferences, so that it is constantly updated with new terms.

The Mathematics Subject Classification (MSC)[6] is the main taxonomy used in the field of Mathematics. This scheme is maintained by Mathematical Reviews[7] and zbMATH[8] and it is adopted by many mathematics journals. It consists of 63 macro-areas classified with two digits: each of them is further refined into over 5K three- and five-digit classifications representing their sub-areas. The last version dates back to 2010 and typically a new official version is released every ten years.

---

[4] Physics and Astronomy Classification Scheme - https://publishing.aip.org/publishing/pacs.
[5] PhySH - Physics Subject Headings - https://physh.aps.org/about.
[6] 2010 Mathematics Subject Classification - https://mathscinet.ams.org/msc/msc2010.html.
[7] Mathematical reviews - http://www.ams.org/mr-database.
[8] zbMATH - https://zbmath.org.

The Medical Subject Heading (MeSH)[9] [27] is the standard solution in the field of Medicine. It is maintained by the National Library of Medicine of the United States and it is constantly updated by collecting new terms as they appear in the scientific literature.

The JEL[10] classification scheme is the most used classification in the field of Economics. It was created by the Journal of Economic Literature of the American Economic Association. Its last major revision dates back to 1990, but in the last years there have been many incremental changes to reflect the latest developments in the field [28]. In the same field, we can also find the STW Thesaurus for Economics developed by ZBW - Leibniz Information Centre for Economics. This thesaurus contains almost 6,000 standardized subject headings and around 20,000 additional terms to support individual keywords.

The Library of Congress Classification[11] is a system of library classification that encompasses many areas of science. It was developed by the Library of Congress and it is used to classify books within large academic libraries in USA and several other countries. However, it is too shallow to support the characterisation of scientific research at a good level of granularity. For instance, the field of Computer Science is covered by only three topics: Electronic computers, Computer science, and Computer software.

Dimensions[12], a company that provides commercial solutions to support users in exploring the research landscape, uses Fields of Research (FoR)[13], a classification developed by the Australian Bureau of Statistics[14], which is included in the Australian and New Zealand Standard Research Classification (ANZSRC) alongside the Research Fields, Courses and Disciplines (RFCD) classification and Socio-Economic Objective (SEO)

---

[9] MeSH - Medical Subject Headings - https://www.nlm.nih.gov/mesh.
[10] Journal of Economic Literature - https://www.aeaweb.org/econlit/jelCodes.php.
[11] Library of Congress Classification: https://www.loc.gov/catdir/cpso/lcc.html.
[12] Dimensions.ai - https://www.dimensions.ai
[13] Implementation of FoR in Dimension.ai - https://web.archive.org/web/20190125122911/https://app.dimensions.ai/browse/publication/for?redirect_path=/discover/publication
[14] Australian Bureaus of Statistics - http://www.abs.gov.au

classification. Fields of Research has three hierarchical levels, namely Divisions (at the broadest level), Groups, and Fields (at the most fine-grained level). Divisions, Groups and Fields are assigned unique 2-digit, 4-digit, and 6-digit codes respectively. A common limitation of these taxonomies is that, being manually crafted and maintained by domain experts, they tend to evolve relatively slow and therefore become quickly outdated. To cope with this issue, some institutions (e.g., the American Physical Society) are crowdsourcing their classification scheme. However, the crowdsourcing strategy also suffers from limitations, such as trust and reliability [29].

2.2 Automatic generation of classification schemes

A complementary strategy is to automatically or semi-automatically generate these classifications using data driven methodologies. In the literature, we can find a variety of approaches for learning taxonomies or ontologies based on natural language processing [30], clustering techniques [31], statistical methods [32], and so on. For instance, Text2Onto [30] is a framework for learning ontologies from a collection of documents. This approach identifies synonyms, sub-/superclass hierarchies, etc. through the application of natural language processing techniques on the sentence structure, where phrases like “such as…” and “and other…” imply a hierarchy between terms. This method presents some similarities with the Klink-2 algorithm [13], but requires the full text of documents. TaxGen [31] is another approach to the automatic generation of a taxonomy from a corpus by means of a hierarchical agglomerative clustering algorithm and text mining techniques. The clustering algorithm first identifies the bottom clusters by observing the linguistic features in the documents, such as co-occurrences of words, names of people, organisations, domain terms and other significant words from the text. Then the clusters are aggregated creating higher-level clusters, which form the hierarchy. This strategy is similar to the one adopted by Klink-2 for inferring the *relatedEquivalent*

relationships. Another approach to automatically creating categorisation systems is the subsumption method [32], which computes the conditional probability for a keyword to be associated with another based on their co-occurrence. Given a pair of keywords, this system tries to understand whether there is a subsumption relationship between them, according to certain heuristics. Shen et al. [33] adopted a variation of this technique for generating the Fields of Study (FoS) for Microsoft Academic [33]. This classification includes both hand-crafted concepts (the first two levels) and topics automatically derived from Wikipedia. However, the taxonomy learning approach focuses on Wikipedia and does not take advantage of the metadata associated with research papers. Conversely, Klink-2 considers both academic publications and external sources. In addition, when mapping CSO to DBpedia (see Section 3.2) we found that only about 61% of CSO topics are represented in the online encyclopedia. It is also possible to combine ontology learning and a crowdsourcing strategy by developing approaches that take into account both statistical measures and user opinions [34, 35]. For instance, Wohlgenannt et al [34] combine human effort and machine computation by crowdsourcing the evaluation of an automatically generated ontology with the aim of dynamically validating the extracted relations.

3 The Computer Science Ontology

The Computer Science Ontology is a large-scale ontology of research areas that was automatically generated using the Klink-2 algorithm [13] on a dataset of about 16 million publications, mainly in the field of Computer Science. In the rest of the paper, we will refer to this corpus as the *Rexplore dataset* [7]. Some parts of the ontology were later refined manually by domain experts during the preparation of two ontology-assisted surveys in the fields of Semantic Web [36] and Software Architecture [16].

The current version of CSO includes 14,164 topics and 162,121 semantic relationships. The main root is Computer Science; however, the ontology includes also a few secondary roots, such as Linguistics, Geometry, Semantics, and so on.

The CSO data model[15] is an extension of SKOS[16] and it includes eight semantic relations:

- *relatedEquivalent*, which is a subproperty of *skos:related*, indicates that two topics can be treated as equivalent for the purpose of exploring research data (e.g., Ontology Matching and Ontology Mapping). For the sake of avoiding technical jargon, in the CSO Portal this predicate is referred to as *alternative label of*.
- *superTopicOf*, which is a subproperty of *skos:narrower*, indicates that a topic is a super-area of another one (e.g., Semantic Web is a super-area of Linked Data). This predicate is referred to as *parent of* in the portal. The inverse of this relationship is *subTopicOf*.
- *contributesTo*, which indicates that the research output of one topic contributes to another. For instance, research in Ontology Engineering contributes to Semantic Web, but arguably Ontology Engineering is not a sub-area of Semantic Web – that is, there is plenty of research in Ontology Engineering outside the Semantic Web area.
- *owl:sameAs*, which is used for mapping CSO topics to equivaled entities in other knowledge graphs (DBpedia, Freebase, Wikidata, YAGO, and Cyc).
- *schema:relatedLink*, which links CSO concepts to relevant web pages that either describe the research topics (Wikipedia articles) or provide additional information about the research domains (Microsoft Academic).

---

[15] CSO Schema - http://cso.kmi.open.ac.uk/schema/cso.
[16] SKOS Simple Knowledge Organization System - http://www.w3.org/2004/02/skos.

- *preferentialEquivalent*, which is used to state the main label for topics belonging to a cluster of *relatedEquivalent*. For instance, the topics Ontology Matching and Ontology Alignment both have their *preferentialEquivalent* set to Ontology Matching. Similarly to *relatedEquivalent*, in our data model we defined *preferentialEquivalent* as a subproperty of *skos:related*.
- *rdf:type*, this relation is used to state that a resource is an instance of a class. For example, a resource in our ontology is an instance of *Topic*, which is a subclass of *skos:Concept*.
- *rdfs:label*, this relation is used to provide a human-readable version of a resource's name.

The Computer Science Ontology is available for download in various formats (N-Triples, OWL, and CSV) from https://cso.kmi.open.ac.uk/downloads. This ontology is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)[17] meaning that everyone is allowed to:

- copy and redistribute the material in any medium or format;
- remix, transform, and build upon the material for any purpose, even commercially.

In the following subsection, we will discuss the automatic generation of CSO by means of the Klink-2 algorithm (Section 3.1), the alignment with external resources (Section 3.2), and the strategy for producing new versions of the ontology (Section 3.3).

3.1 Generating CSO

CSO was automatically generated by Klink-2 [13], an algorithm that produces an ontology of research topics by processing information from scholarly metadata (titles, abstracts,

[17] CC BY 4.0 International License - https://creativecommons.org/licenses/by/4.0.

keywords, authors, venues) and external sources (e.g., DBpedia, calls for papers, web pages). Klink-2 is able to produce a complete ontology including all the topics represented in the input dataset or can, alternatively, focus on specific branches under seed keywords (e.g., "Semantic Web"). In the following, we briefly summarize this approach. A more comprehensive description is available in Osborne and Motta [13].

In Algorithm 1, we report the pseudocode of Klink-2. The algorithm takes as input a set of keywords and investigates their relationships with the set of their most co-occurring keywords. In particular, Klink-2 infers the semantic relationship between keyword *x* and *y* according to a relationship R with a set of entities (e.g., research papers, authors, tools) by means of three metrics: i) $H_R(x, y)$, which uses a semantic variation of the subsumption method for measuring the intensity of a hierarchical relationship; ii) $T_R(x, y)$, which uses temporal information to do the same; and iii) $S_R(x, y)$, which estimates the similarity between two topics. The first two are used to detect *superTopicOf* and *contributesTo* relationships, while the latter is used to infer *relatedEquivalent* relationships.

$H_R(x, y)$ quantifies the hierarchical relationship between *x* and *y* according to the following formula:

$$H_R(x, y) = \left(\frac{I_R(x,y)}{I_R(x,x)} - \frac{I_R(y,x)}{I_R(y,y)}\right) \cdot c_R(x, y) \cdot n(x, y) \qquad (1)$$

where $I_R(x, y)$ is the number of elements associated with both *x* and *y* according to relation *R* (e.g., number of co-occurrences in research papers), $\frac{I_R(x,y)}{I_R(x,x)}$ is the conditional probability that an element associated with keyword *x* will be associated also with keyword *y*, $n_R(x, y)$ is the Levenshtein distance between the two keywords normalized by the length of the longest one, and $c_R(x, y)$ is the cosine similarity between the two vectors in which each index represents the keyword *k,* which has in common with *x* and/or *y* a set of

instantiations of a relation *R* with the same scholarly entities, with the values equal to $I_R(k,x)$ for *x* and $I_R(k,y)$ for *y*.

$T_R(x,y)$ is a temporal version of $H_R(x,y)$, which weighs more the information associated with the first years of *x*. It is useful to detect the cases in which the relationship between two terms fade because their association has become implicit (e.g., Artificial Intelligence and Machine Learning). $T_R(x,y)$ is calculated using a variation of formula (1) in which $I_R(x,y)$ is computed by weighting the intensity of the relationships in each year according to the distance from the debut of *x*. The weight is computed as $w(year, x) = (year - debut(x) + 1)^{-\gamma}$, with $\gamma > 0$ ($\gamma = 2$ in the prototype).

Finally, $S_R(x,y)$ is used to assess the similarity of two terms and is computed according to the following formula:

$$S_R(x,y) = \frac{c_R(x,y)}{max\left(c_R{}^{super}(x,y)\,,\, c_R{}^{sib}(x,y)\right)+1} \qquad (2)$$

where $c_R{}^{super}(x,y)$ is the cosine similarity of the super topics of the two terms in the taxonomy produced by previous iteration, and $c_R{}^{sib}(x,y)$ is the cosine similarity of their siblings.

A hierarchical relationship between two topics is inferred when a sufficient number of hierarchical indicators are above a threshold. An analysis of the precision/recall trade-off associated with different thresholds is available in [13]. The nature of the inferred relationship is assessed by Klink-2 using a rule-based approach. In brief, if *x* is older, associated with more entities, and the $T_R(x,y)$ indicators score higher, Klink-2 will infer a *superTopicOf* relationship, otherwise a *contributesTo* one. Then, Klink-2 removes loops in the topic network (instruction #9 of Algorithm 1), merges keywords linked by a *relatedEquivalent* relationship, and splits ambiguous keywords associated to multiple

meanings (e.g., “Java”). The keywords produced in this step are added to the initial set of keywords to be further analysed in the next iteration and the while-loop is re-executed until there are no more keywords to be processed. Finally, Klink-2 filters the keywords considered “too generic” or “not academic” according to a set of heuristics (instruction #13) and generates the triples describing the ontology.

```
Input  : List of keywords keywords, User feedbacks feedbacks
Output: Ontology CSO
relationships={};                          // Initialise an empty set
while some keywords yet to process do
    foreach k1 in keywords do
        candidates = GetCandidates(k1, feedbacks);
        foreach k2 in candidates do
            relationship = InferRelationship(k1, k2, feedback,
              relationships);
        end foreach
    end foreach
    relationships = RemoveLoops(relationships);
    new.keywords = MergeAndSplitKeywords(keywords, relationships);
    keywords = AddNewKeywords(new.keywords);
end while
keywords = FilterTopics(keywords, feedbacks, relationships);
CSO = GenerateSemanticRelationships(relationships);
return(CSO);
```

Algorithm 1. The Klink-2 algorithm used to generate CSO.

A formal evaluation of Klink-2, centred on the Semantic Web area, is described in [13]. In particular, with the help of three senior researchers, we generated a gold standard ontology[18], which includes 88 research topics in the field of the Semantic Web. When comparing automatically generated topic taxonomies in the Semantic Web area with the manually-created gold standard, we found that Klink-2 significantly outperformed the alternative algorithms (p=0.0005), yielding a precision of 86% and a recall of 85.5%. More details about Klink-2 and its evaluation can be found in [13].

The other five semantic relations are also automatically generated. The *rdf:type* and *rdfs:label* relations are created to identify all topics and their label. The *preferentialEquivalent* relation, which identifies the main label to be used for a cluster of

[18] Gold Standard - http://technologies.kmi.open.ac.uk/rexplore/data.

topics linked by a *relatedEquivalent*, is produced by choosing the label associated with most articles in the source corpus. In the next section, we will describe the generation of the *owl:sameAs* and *schema:relatedLink* relations.

3.2 Aligning CSO with other Knowledge Bases

Aligning CSO with other Knowledge Bases (KBs) can provide access to a wealth of information which can be beneficial for a number of tasks, such as creating multi-lingual translations and exploiting other domain-specific datasets. For this reason, we aligned CSO with five well-known KBs (DBpedia [19], Wikidata [20], YAGO [21], Freebase [22], Cyc [23]). We also linked CSO with two web sites that contain additional information about research topics: Wikipedia and Microsoft Academic. The latter is a free public web search engine for academic publications, which offers a multi-disciplinary taxonomy of research areas, known as Fields of Study (FOS).

We used the *owl:sameAs* relation to link CSO concepts to equivalent entities in the other KBs and the *schema:relatedLink* relation to refer to external webpages that contain further information about the research topic. In total, we produced 27,803 relationships.

The mapping of CSO with others KBs was performed in two steps. First, we identified the DBpedia entities corresponding to CSO topics by exploiting the DBpedia Spotlight API [37]. Then we extracted the links from DBpedia to other KBs in the Linked Open Data (LOD) cloud [38] by using the DBpedia SPARQL endpoint[19]. Each topic was associated to the corresponding DBpedia entity by feeding an artificial sentence, listing the labels of the topic and of its direct sub- and super-topics, into the DBpedia Spotlight API. The resulting JSON response contains a list of candidate DBpedia entities, each one parametrized by means of a *SimilarityScore* from 0 to 1. While considering the entities with *SimilarityScore*

[19] DBpedia SPARQL endpoint - http://dbpedia.org/sparql

= 1 is the most conservative solution, only 812 out of the 8,517 retrieved DBpedia entities (9.53%) met these criteria. Therefore, we employed a decision tree classifier to mark the suggested DBpedia entries as correct or incorrect. In order to accomplish this, we produced a gold standard in which three human annotators manually tagged 550 DBpedia entities. Out of the 550 labelled data, 495 were used for training the classifier and 55 for validating it. The classifier used the following DBpedia Spotlight outputs as the training features: *SimilarityScore*, *Offset*, and *PercentageOfSecondRank*. The trained classifier scored 76.4% accuracy on the test set. For the sake of completeness, we tested the classifier with smaller training sets obtaining accuracy values that range from 70.9% (using 100 training samples) up to 76.4% (using 495 training samples). When using a validation set of 110 samples (20% of the gold standard), we obtain similar accuracy values ranging from 72.7% (with 100 training samples) up to 75.4% for 440 training samples. Using the resulting decision tree, we mapped 5,234 CSO topics (36.9% of the total) to the corresponding DBpedia entities. The URL of the relevant Wikipedia articles were extracted from the DBpedia resource pages. The URL of Microsoft Academic pages were instead generated by matching the labels of the CSO topics with those of the FOS concepts. The resulting web pages provide a description of the topics, and a list of top authors, affiliations and publications associated with them.

Table 1 shows the number of relationships to external resources in CSO. We also plan to list our ontology as an open dataset in the Linked Open Data Cloud and increase the number of resources connected with the CSO.

Table 1. Number of resources linked to concepts in the CSO.

Type of resource	Resource	Number of matched entities
Knowledge Graphs	DBpedia	5,234
	Wikidata	5,202
	Freebase	5,133
	YAGO	3,324
	Cyc	167
Web Pages	Wikipedia	5,204
	Microsoft Academic	3,539

3.3 Generation of New Versions

We plan to periodically release new versions of the CSO ontology by running Klink-2 on the latest publications in Computer Science and by incorporating the feedback from the community. This process will be supervised by a steering committee composed of a small number of individuals drawn from The Open University and our collaborators. The current composition of the steering committee is available at https://cso.kmi.open.ac.uk/about. Depending on the success and impact of the initiative, we expect that the committee will grow significantly in the future and expand to include representatives of a variety of organisations. Both minor and major revisions will be released on a regular basis.

M*inor revisions* will be produced by directly implementing in the ontology the changes suggested by users and confirmed by the steering committee. The changes may include: i) removal of a topic, ii) removal of a relationship, iii) inclusion of a relationship, and iv) inclusion of a topic. In this phase, we will focus on correcting specific errors rather than expanding the ontology.

Major revisions will be produced by feeding to the Klink-2 algorithm an up-to-date corpus of publications and the set of "correct relationships" suggested by users and confirmed by the steering committee. Indeed, the current version of Klink-2 is able to take as input user defined relationships and incorporate them in the automatically generated ontology. The goal is to make sure that major revisions of CSO include all significant research areas that have emerged in the interval since the previous major release.

We aim to produce at least one major revision every year. The timing on the other revisions will depend on the number and quality of feedback entries. For instance, a significant number of negative feedback entries on a certain branch will trigger a comprehensive revision of it. In such a case, we will contact domain experts and invite them to review the associated branch on the CSO Portal. For instance, in a recent study

[16], we assessed the CSO branch regarding Software Architecture by generating a spreadsheet representation of it and having it reviewed by three senior researchers. The CSO Portal should make this process simpler and easier to track.

The current version of CSO is 3.1. In this release we improved the accuracy of the relationships by fixing the issues flagged by the initial users of the CSO Portal and discarding several branches outside the field of Computer Science. Since the CSO has been generated using a corpus of papers in Computer Science, it tends to be most accurate with regard to the topics which appear prevalently in this domain. Conversely, the relationships involving topics from other fields were generated according to a possibly skewed sample of publications, leading to a higher error rate. It was thus decided to produce a revision on the ontology that would include only the sub-topics of Computer Science and a few other verified roots, including Communication, Economics, Education, Engineering, Geometry, Linguistics, Mathematics, Semantics, Sociology, and Topology. The resulting release naturally contains fewer topics and relationships, but it is overall much cleaner than the previous version [18]. Another important novelty of version 3.1 is the new data model described in Section 3.

4 CSO Classifier

A key role of CSO is to support the classification of scholarly documents in the Computer Science field. To this purpose, we also released the CSO Classifier, an unsupervised approach for automatically classifying research papers according to CSO. This application takes in input the metadata associated with a research paper (title, abstract, and keywords) and returns a selection of research concepts drawn from CSO.

The first version was developed in 2013 in the context of developing the Rexplore platform [7] and it was subsequently used in support of several research approaches and

applications, more thoroughly discussed in the next section. In particular, this initial version of the CSO Classifier was used for supporting Springer Nature editors in annotating Computer Science proceedings [3]. However, this initial version could only identify concepts that were explicitly referred to in the input papers [3, 39]. For this reason, we have recently developed a new version of the CSO Classifier [24], which uses a combination of linguistics and semantics to generate a more comprehensive set of topics, including topics that may not be explicitly mentioned in the metadata.

This new version, which is described in detail in [24], operates in three steps, as shown in Figure 1.

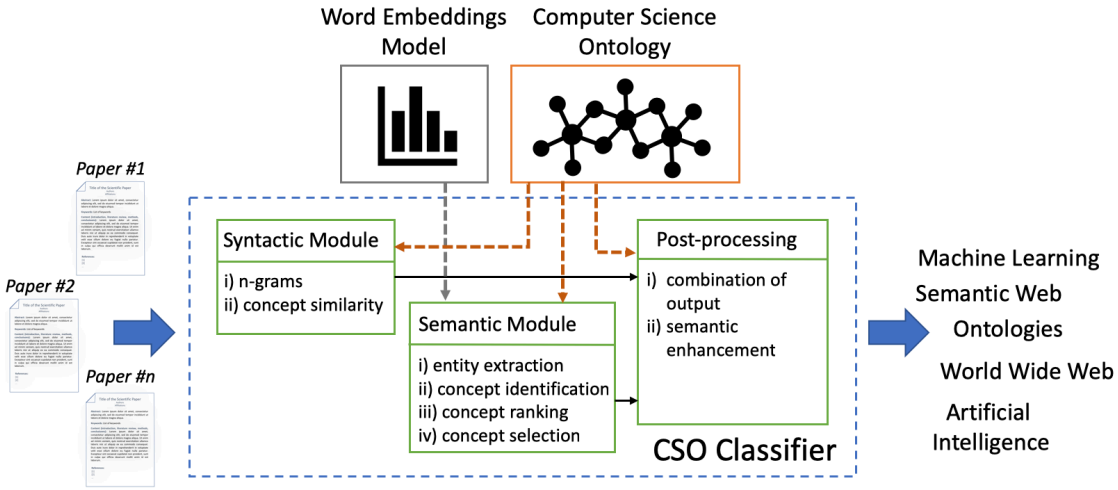

Figure 1. Workflow of the CSO Classifier.

Initially, it finds all topics in the ontology that are explicitly mentioned in the paper. Then, it employs part-of-speech tagging and word embeddings to infer additional semantically-related topics. Finally, it enriches the resulting topics by including their relevant super-areas, exploiting the *superTopicOf* relationships. For instance, given the topic "machine learning", it will also infer "artificial intelligence". This new version has been evaluated on a gold standard of manually annotated research papers, demonstrating a significant improvement over alternative approaches, including the earlier version of the CSO classifier.

The Python implementation of the latest version of the CSO Classifier is available at https://github.com/angelosalatino/cso-classifier. An online demo of this application is available within the CSO Portal[20].

5 CSO Adoption

The Computer Science Ontology has been used in a variety of applications and research efforts. In particular, it informs several tools supporting editorial activities at Springer Nature. In this section, we discuss these systems with the aim of showing the practical value of CSO and inspiring further applications.

5.1 CSO and Springer Nature

The Open University and Springer Nature have been collaborating since 2013 in the development of new solutions to assist the work of the Computer Science editorial team at Springer Nature. The main result of this collaboration are two applications which demonstrate the value of using CSO in the context of developing intelligent functionalities that take as input scholarly entities: Smart Topics Miner and Smart Book Recommender.

Smart Topic Miner [3, 40] (STM)[21] is a tool developed for supporting the Springer Nature editorial team in classifying editorial products according to a taxonomy of research topics drawn both from CSO and the *Product Market Codes* (PMC), Springer Nature's own editorial classification system. This information is then used for: i) classifying proceedings in digital and physical libraries; ii) enhancing semantically the metadata associated with publications and consequently improving the discoverability of the proceedings; and iii)

[20] Demo of the CSO classifier within the CSO Portal, available at https://cso.kmi.open.ac.uk/classify/.
[21] Demo of Smart Topic Miner - http://stm-demo.kmi.open.ac.uk.

detecting promising emerging research areas that may deserve more attention from the publisher.

STM takes as input the metadata associated with the proceedings of a conference and returns the set of relevant CSO topics and PMCs as output. The input metadata contains titles, abstracts and author-provided keywords for each paper in the proceedings. STM performs three steps on this data. First, it uses the CSO Classifier to annotate each paper with the topics from CSO. Then it groups and ranks the topics according to the number of papers addressing them. Finally, it infers the relevant PMCs, using the mapping between the CSO ontology and PMC. The editors then review the CSO topics and the PMC categories using the interface depicted in Figure 2 and submit these annotations to the Springer Nature production system. Such keywords are then displayed in the Springer Nature's digital library, SpringerLink[22] and included in the ONIX metadata feeds, delivered to various libraries and bookshops.

STM was first introduced in 2016 [3] and has since been used routinely by the editorial team to annotate all book series covering conference proceedings in Computer Science, including LNCS, LNBIP, CCIS, IFIP-AICT and LNICST, for a total of about 800 volumes each year. During this period, the adoption of STM has halved the time needed for classifying proceedings from 20-30 to 10-15 minutes. In addition, STM also provided the additional important benefit of reducing the complexity of this task, which traditionally has been performed by Senior Editors. Indeed, thanks to the introduction of STM in the editorial workflow, it has now become possible for this task to be carried out by junior editors, ultimately achieving an overall 75% cost reduction. The adoption of CSO topics produced a significant increment of the discoverability of relevant publications on

[22] SpringerLink - https://link.springer.com.

SpringerLink[23], Springer Nature digital library, resulting in about 9 million additional downloads over the last three years.

The current version of STM (STM 2.0 [40]) implements several new features based on the feedback received by the editors over the last few years. The main novelties include 1) a more user-friendly interface (see Figure 2), 2) a new back-end which utilize the CSO Classifier [24], 3) the ability to take into account the annotations of previous editions of a conference, and 4) the integration with the production system at Springer Nature and the CSO Portal. We refer the reader to Salatino et al. (2019 [40]) for a comprehensive description of the system and its evolution over the years.

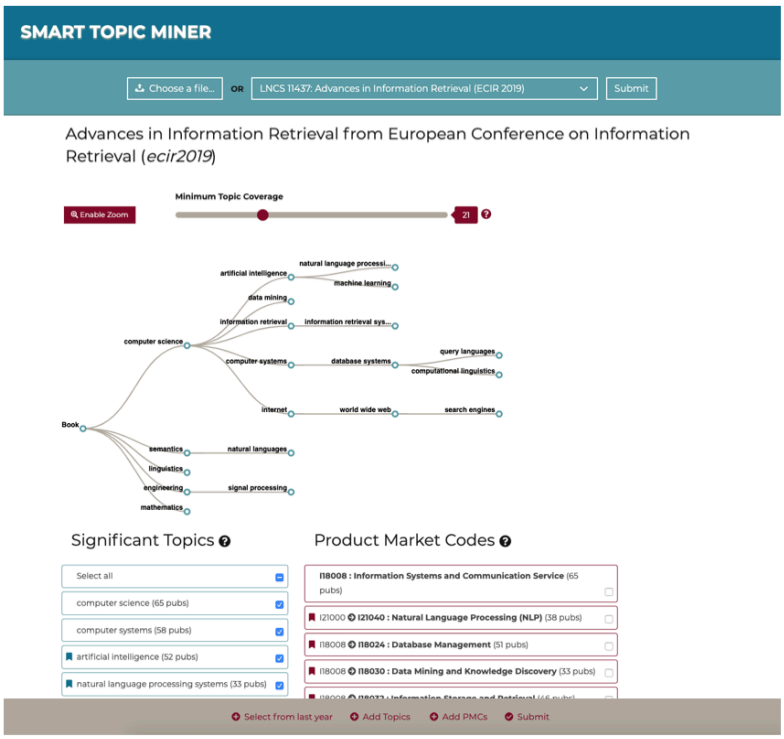

Figure 2. The STM interface.

Smart Book Recommender [17] (SBR)[24] is an ontology-based recommender system that takes as input the proceedings of a conference and suggests books, journals, and other

[23] https://link.springer.com/
[24] Demo of Smart Book Recommender - http://rexplore.kmi.open.ac.uk/SBR_demo.

conference proceedings which are likely to be relevant to the attendees of the conference in question. It uses the CSO Classifier to represent 27K books and 320 journals according to their research topics. This semantic representation is then used to compute the similarity between the input conferences and the editorial products. SBR also exploits the CSO topic taxonomy to graphically represent and compare conferences and books, allowing users to understand the rationale behind a recommendation. For instance, Figure 3 shows the interactive graph view that compares the topic taxonomy associated with the International Semantic Web Conference with the one associated with the suggested book, "Handbook of Semantic Web Technology".

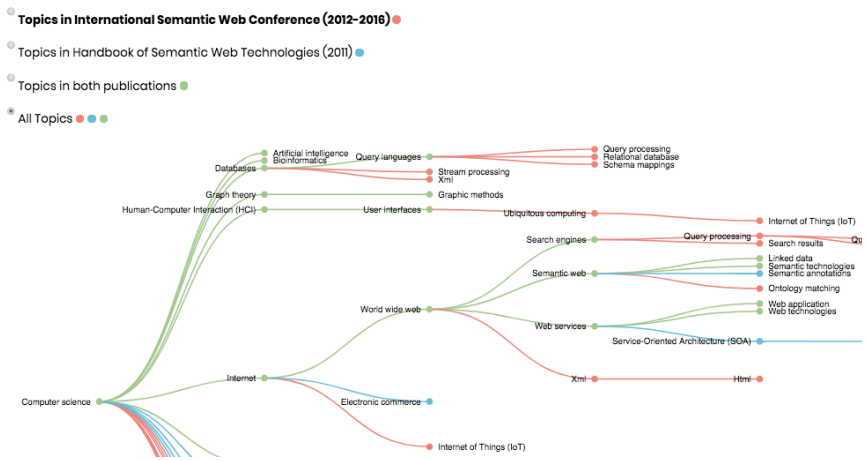

Figure 3. Portion of the graph view showing the taxonomies of the topics associated with the input conference and one of the recommended editorial products.

SBR was evaluated with a user study involving seven Springer Nature editors from the Heidelberg, London, New York, and Beijing, who assessed the quality of the book lists produced for the main conferences in their areas of expertise. 72.9% of the SBR recommendations were marked as relevant[25]. In addition, editors assessed SBR as very user friendly, yielding an average SUS scores of 77.1±15.2[26].

[25] The full results of the survey are available on Figshare at:

https://figshare.com/articles/Smart_Book_Recommender_Evaluation_Data/6087032

[26] Specifically, the editors scored 57.5, 77.5, 95, 60, 92.5, 70, and 87.5.

While the results are promising, there is nonetheless much scope for improvement and, in particular, we are currently working on a new version of the system, in which the basic similarity matching mechanism used by SBR will be augmented with the heuristic selection criteria used by the Springer Nature marketing team, to provide a more robust solution. The goal is for this new version of the system to be used for selecting the books to be marketed at a couple of hundred computer science conferences per year.

Broader implications. The use of CSO in these two applications demonstrates several benefits. First of all, the adoption of an automatic mechanism for annotating scholarly documents with CSO topics introduces a more consistent process compared to manual annotation. This robustness is also enhanced by the use of a semantic characterization. For example, "ontology matching" and "ontology alignment" are modelled as equivalent in CSO, which means that the noise associated with the use of different labels for the same research area in scholarly documents does not affect the accuracy of either SBR or STM.

Nonetheless, several improvements could be made by considering not just the production process associated with published papers but the entire lifecycle of academic publications. In particular, it would be desirable to use CSO earlier in the research lifecycle, starting with submission systems, such as EasyChair, Microsoft Conference Management Toolkit, etc, to assist both authors in generating keywords for their papers and program chairs to assign papers to reviewers. This latter task could, for instance, be supported by using both the CSO Classifier and the Toronto Matching System [41] in the same conference submission system, combining the strengths and weaknesses of the two approaches.

5.2 Evaluation of CSO on different tasks

Since its introduction in 2012, the Computer Science Ontology has been used in several studies and proved to effectively support a wide range of research tasks such as:

- forecasting research topics [11];
- exploration of scholarly data [7];
- detection of research communities [10];
- ontology forecasting [42];
- ontology evolution [15];
- forecasting technology adoption [12];
- systematic literature reviews [16, 36, 43] .

Here we describe a selection of these research efforts and report several evaluations demonstrating the practical advantage of adopting CSO. For reason of space, we will be necessarily brief; we refer the interested reader to the original papers for additional details.

Forecasting research topics. *Augur* [11] is an approach that aims to detect the emergence of new research areas by analysing topic networks and identifying clusters associated with a significant increase in the pace of collaboration. It exploits CSO for creating semantically enhanced topic networks describing the collaboration between research topics over time. Over these networks, Augur applies a novel clustering algorithm called the Advanced Clique Percolation Method (ACPM). The resulting clusters of topics indicate the areas of the network that are nurturing new research areas.

The evaluation of Augur [11] showed that semantically enriching topics networks with CSO yields a significant performance improvement on the task of predicting the emergence of new research areas. Table 2 reports precision and recall obtained in the period 1999-2009 by a version of Augur using CSO and by an alternative version exploiting keywords to represent research topics[27].

[27] The evaluation material of Augur is available at http://rexplore.kmi.open.ac.uk/JCDL2018.

Table 2. Performance of Augur [11] when characterising topics with keywords or CSO. In bold the best results.

	Keywords		CSO	
	Precision	Recall	Precision	Recall
1999	0.68	0.49	**0.86**	**0.76**
2000	0.62	0.39	**0.78**	**0.70**
2001	0.69	0.49	**0.77**	**0.72**
2002	0.65	0.50	**0.82**	**0.80**
2003	0.72	0.54	**0.83**	**0.79**
2004	0.70	0.47	**0.84**	**0.68**
2005	0.62	0.49	**0.71**	**0.66**
2006	0.32	0.32	**0.43**	**0.51**
2007	0.06	0.21	**0.28**	**0.44**
2008	0.06	0.08	**0.15**	**0.33**
2009	0.05	0.59	**0.09**	**0.76**

Exploration of scholarly data. *Rexplore* [7] is a system that leverages novel solutions in large-scale data mining, semantic technologies and visual analytics, to provide an innovative environment for exploring and making sense of scholarly data. It uses CSO for characterising research papers, authors, and organisations according to their research topics and for producing relevant views. For instance, Rexplore is able to plot the collaboration graph of the top researchers in a field and to visualise researchers in terms of the shifting of their research interests over the years. Rexplore also describes each topic in CSO with a variety of analytics, and allows users to visualise the trends of its sub-topics.

The Rexplore system was shown to be able to support users in performing specific tasks more effectively than Microsoft Academic Search (MAS), thanks to its organic representation of research topics [7]. In a user study, 26 researchers were asked to perform the following three tasks:

- Task 1. Find the top 3 ‘rising stars’ in the United Kingdom with expertise in both *Semantic Web* and *Social Networks*, in the career range 5-15 years from first publication, ranked in terms of number of citations in these 2 areas.
- Task 2. Find the top 5 authors with the highest number of publications in the *Semantic Web* and rank them in terms of number of publications in *Artificial*

Intelligence. For each of them find their most cited paper in *Artificial Intelligence*.

- Task 3. Find which are the 2 sub-topics in *Semantic Web* that have grown the most in 2005-2010 (as measured by the difference between the number of papers in 2010 and in 2005) and who are the top 2 authors (ranked by number of publications in topic) in these 2 topics.

The results indicate that in adopting Rexplore, the participants were able to complete such tasks more quickly and with higher success rate. More details about the evaluation are available in Table 3.

Table 3. Experimental results (in min:secs) using Rexplore and MAS to perform three different tasks. In bold the best result.

Rexplore (CSO) (17 participants)			
	Average Time	Standard Deviation	Success Rate
Task 1	**03:06**	**00:45**	**100%**
Task 2	**08:01**	**02:50**	**94%**
Task 3	**07:51**	**02:32**	**100%**
MAS (no CSO) (9 participants)			
	Average Time	Standard Deviation	Success Rate
Task 1	14:46	00:24	33%
Task 2	13:52	01:35	50%
Task 3	15:00	00:00	0%

Detection of research communities. The *Temporal Semantic Topic-Based Clustering* (TST) [10] is an approach for detecting research communities by clustering researchers according to their research trajectories, defined as distributions of topics over time.

Figure 4 shows the performance of four alternative approaches and 13 human experts in detecting communities in the field of Semantic Web. The best version of TST (labelled TST), that took into consideration both CSO topics and their semantic relationships, was found to be not significant different from the human experts ($p > 0.14$). Conversely, the two approaches (labelled FC and FT) that used CSO simply as a vocabulary of terms, ignoring their semantic relationships, obtained a significantly lower performance

($p<0.0001$). Finally, the baseline that simply used author-defined keywords (labelled F) performed the worst.

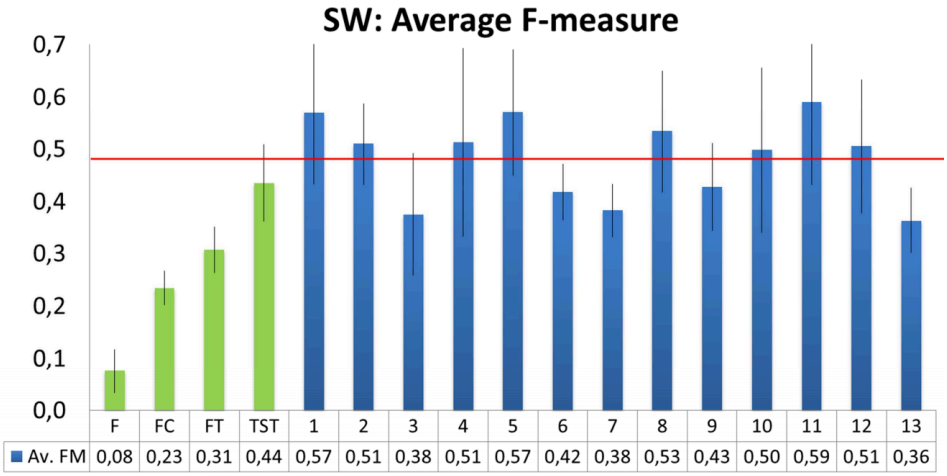

Figure 4. Average F-measure between each expert/algorithm and all the other experts for the SW topic. The red line represents the average F-measure of the experts

Ontology Forecasting. The *Semantic Innovation Forecast* model (SIF) [42] is an approach to predict new concepts of an ontology at time *t+1*, using only data available at time *t*. Specifically, the proposed model favours the generation of innovative topics by considering distributions that enclose innovative and adopted lexicons based on word priors computed from historical data.

The full version of SIF, learning from concepts in CSO, was able to significantly outperform[28] several variations of LDA [44], as reported in Table 4.

Table 4. Mean average precision @10 for SIF [42] and other four alternative algorithms based on LDA [44]. In bold the best results.

YEAR-FORECAST	YEAR-TRAINED	YEAR-PRIOR	SIF (CSO)	LDA	LDA-A	LDA-I	LDA-IA
2000	1999	1997-1999	**0.7031**	0.125	0.4761	0	0.408
2002	2001	1999-2001	**0.875**	0	0.8227	0.6428	0.7486
2004	2003	2001-2003	**0.906**	0	0.5822	0.5726	0.6347
2006	2005	2003-2005	**0.8755**	0.3069	0.7853	0.8385	0.6893
2008	2007	2005-2006	**0.988**	0.398	0.681	0.5661	0.7035
AVG			**0.8695**	0.1659	0.6694	0.524	0.6368

[28] The evaluation material of SIF can be found at http://technologies.kmi.open.ac.uk/rexplore/ekaw2016/OF.

Ontology Evolution. The *Pragmatic Ontology Evolution* (POE) [15] is an approach for selecting the best set of new concepts to include in the evolved version of an ontology so that i) is consistent with user requirements, ii) is parametrised with respect to a number of dimensions (e.g., topological considerations), and iii) effectively supports relevant computational tasks. POE tests different combinations of several parameters to weigh the candidate concepts by measuring to the performance of the resulting ontologies on a set of tasks, such as instance tagging and generation of recommendations. Then it applies variation of the Recursive Feature Elimination to produce the set of concepts that optimise the ontology ability to support the task.

POE was evaluated by measuring the performance of the produced ontologies in supporting four typical computational tasks. The versions of POE using the CSO ontology for representing research topics significant outperformed (p=0.0004 with Wilcoxon's rank test) the alternative approaches based on term frequency (labelled FS, FR) and TF-IDF (TS, TR)[29]. Table 5 summarise the results.

Table 5. Performance of alternative approaches for ontology evolutions. In bold the best results.

Tasks	FS	FR	TS	TR	POE1 (CSO)	POE2 (CSO)	POE3 (CSO)	POE4 (CSO)	POES (CSO)
Instance Tagging	0.922	0.903	0.908	0.895	**0.969**	0.966	0.968	0.945	0.967
Similarity Computation	0.861	0.858	0.358	0.859	0.903	**0.916**	0.915	0.904	0.914
Generation of Recommendations	0.957	0.957	0.956	0.957	0.976	0.981	**0.982**	0.978	**0.982**
Clustering	0.926	0.906	0.911	0.931	0.948	0.974	0.974	**0.975**	0.973

A prototype of POE was used for producing recommendations to evolve PMC at Springer Nature. These resulting suggestions were reviewed by Springer Nature editors and eventually adopted in the current version of PMC.

Forecasting Technology Adoption. The *Technology-Topic Framework* (TTF) [12] is an approach that characterises technologies according to their propagation through research

[29] The evaluation material of POE is available at http://rexplore.kmi.open.ac.uk/POE.

topics drawn from CSO, and uses this representation to forecast the propagation of novel technologies across research fields. The aim is to suggest promising technologies to scholars and accelerate the flow of knowledge from one community to another and the pace of technology propagation.

The system was evaluated[30] on a set of 1,118 technologies and proved to be able to forecast the adoption of these technologies in research areas such as Information Retrieval, Databases Systems, and World Wide Web, as showed in Table 6.

Table 6. TTF performance in selected topics.

Topics	Prec	Rec.	F1	Topics	Prec.	Rec.	F1
information retrieval	92.6%	66.8%	77.6%	wireless networks	64.7%	47.3%	55.0%
database systems	82.6%	65.9%	73.3%	sensor networks	71.9%	13.6%	54.3%
world wide web	88.6%	56.1%	68.7%	software engineering	70.6%	44.0%	54.2%
artificial intelligence	63.6%	55.2%	66.5%	distributed com.sys.	67.5%	45.0%	54.0%
computer architecture	62.3%	63.3%	65.7%	quality of service	19.6%	48.6%	53.5%
computer networks	82.1%	54.0%	65.2%	imaging systems	100.0%	35.8%	52.8%
image coding	96.8%	46.9%	63.2%	data mining	60.8%	45.3%	52.0%
P2P networks	78.9%	50.8%	61.9%	computer vision	92.3%	36.0%	51.8%
telecom. traffic	70.8%	48.1%	57.3%	Program, languages	65.3%	42.0%	51.2%
wireless telecom.sys.	74.4%	46.4%	57.1%	problem solving	69.0%	39.7%	50.4%
sensors	78.8%	43.7%	56.2%	semantic web	77.8%	37.1%	50.2%
web services	13.3%	42.2%	56.0%	image quality	74.2%	37.7%	50.0%

Systematic Literature Reviews. *EDAM* [16] is an expert-driven automatic methodology for creating systematic reviews that limits the amount of tedious tasks that have to be performed by human experts. Typically, systematic reviews require domain experts to annotate hundreds of papers manually. EDAM is able to skip this step by i) characterising the area of interest using an ontology of topics, ii) asking domain experts to refine this ontology, and iii) exploiting this knowledge base for classifying relevant papers and producing useful analytics. The implemented approach adopted CSO and used a previous version of the CSO Classifier for categorising under a topic all papers that contain in the title, abstract, or keyword field the label of the topic, its *relatedEquivalent*, or its *superTopicOf*. It was evaluated on the task of classifying papers in field of Software

[30] The evaluation material of TTF is available at http://rexplore.kmi.open.ac.uk/TTF.

Architecture and its performance was not statistically significantly different from that of six senior researchers in the field (p=0.77). Table 7 shows the degree of agreement between the researchers, computed as the ratio of papers which were tagged with the same category by both annotators. The approach adopting CSO yielded the highest average agreement and also obtained the highest agreement with three out of six domain experts.

Table 7. Agreement between annotators (including EDAM) and average agreement of each annotator. In bold the best results.

	EDAM (CSO)	User1	User2	User3	User4	User5	User6
EDAM (CSO)	-	56%	68%	64%	64%	76%	64%
User1	**56%**	-	40%	**56%**	36%	48%	44%
User2	68%	40%	-	64%	52%	**76%**	64%
User3	64%	56%	64%	-	52%	64%	**68%**
User4	**64%**	36%	52%	52%	-	**64%**	52%
User5	**76%**	48%	**76%**	64%	64%	-	72%
User6	64%	44%	64%	**68%**	52%	72%	-
AVG	**66%**	45%	58%	59%	51%	63%	60%

EDAM and CSO have been used in two recent reviews in the field of Human-Computer Interaction (HCI) [43] and Semantic Web [36]. The first study [43] focuses the evolution of HCI in the last 50 years and analyses the papers published by the International Journal of Human-Computer Studies (IJHCS, active since 1969) and the Conference on Human Factors in Computing Systems (CHI, active since 1982). The authors annotated these articles with the CSO Classifier and produced several analytics about the evolution of the topics over the years, their geographical distribution, and the emerging trends. The second study [36] performed a similar analysis on two datasets covering respectively the main Semantic Web venues (ISWC, ESWC, SEMANTiCS, SWJ, and JWS) and 32,431 publications associated with the Semantic Web from a dump of Scopus[31]. It also compares three methods of associating topics with research papers: Rexplore [7] (using CSO), PoolParty[32], and Saffron [45]. We hope that releasing CSO and the CSO Classifier will encourage other researchers to produce similar analyses on other domains.

[31] Elsevier's Scopus: https://www.scopus.com
[32] https://www.poolparty.biz/system-architecture/

6 The CSO Portal

The CSO Portal is a web application that enables users to download, explore, and provide granular feedback on CSO. It is available at http://cso.kmi.open.ac.uk.

Figure 5 shows an overview of the CSO Portal. We consider three kinds of users: unregistered users, registered users, and members of the steering committee. Unregistered users can download the ontology and browse it by using three alternative interfaces. Registered users are also allowed to post feedback regarding the full ontology or specific topics or relationships. The members of the steering committee have the task of reviewing the user feedback and select the changes to be incorporated in new releases of CSO.

In the following sections, we will discuss the different functionalities offered by the CSO Portal, such as: i) exploring CSO, ii) leaving feedback at different levels of granularity, iii) curating and modifying CSO according to the decisions of the steering committee, and iv) finding the shortest path between two topics. Finally, we will report some preliminary data about the usage of CSO and the CSO Portal.

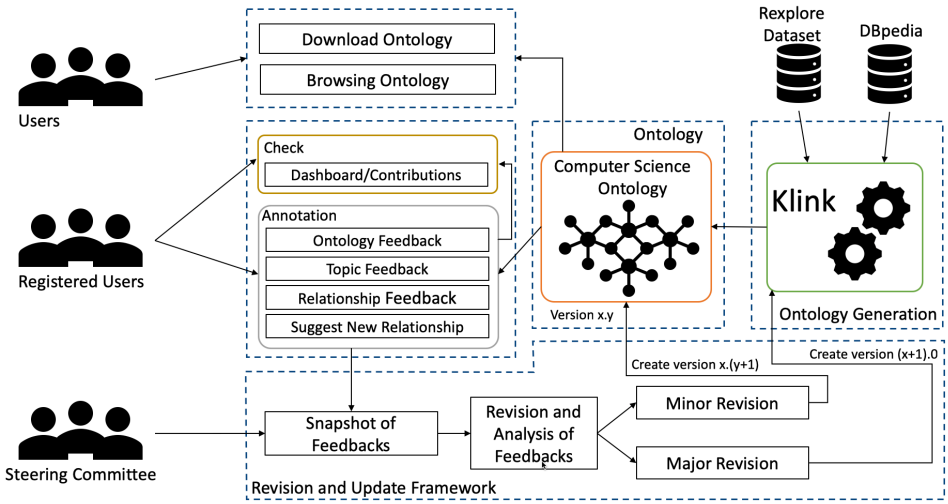

Figure 5. Overview of the Computer Science Ontology Portal.

6.1 Exploring CSO

An important functionality of the CSO Portal is the ability to search and navigate the 14K research topics in CSO. The homepage of the portal (Figure 6) provides a simple search bar as a starting point. The user can type the label of any topic (e.g., “Semantic Web”) and submit it to be redirected to that topic page.

For a given topic, this page shows its *superTopicOf* and *relatedEquivalent* relationships with the relevant topics. For the sake of clarity, these relationships are presented to the users as **parent of/child of** and **alternative label of.** For instance, the relationships:

- *semantic web* **superTopicOf** *RDF*
- *ontology mapping* **relatedEquivalent** *ontology alignment*

are presented as:

- *semantic web* **parent of** *RDF* or *RDF* **child of** *semantic web*
- *ontology mapping* **alternative label of** *ontology alignment*

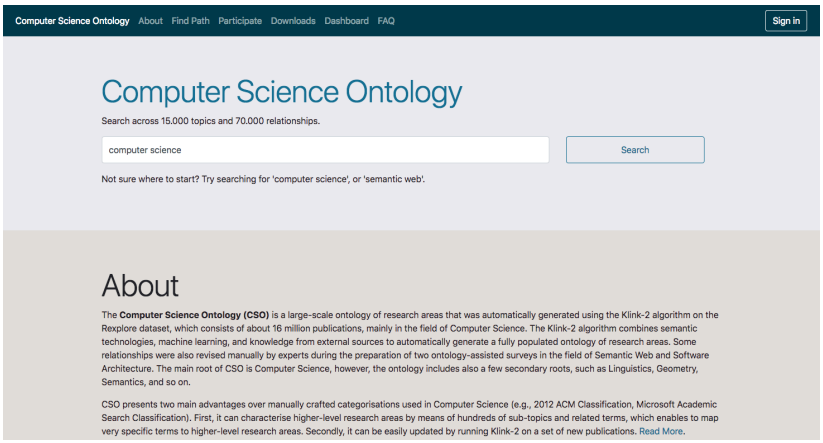

Figure 6. Homepage of the Computer Science Ontology Portal.

The CSO Portal offers three different interfaces to visualise and explore the topic relationships: the *graph view*, the *detailed view*, and the *compact view*. Figure 7, Figure 8, and Figure 9 show how these three views represent the topic "*semantic web*"[33].

The **graph view** is an interactive interface that allows users to seamlessly navigate the network of topics within CSO. In this view, each topic is represented as a node and the *superTopicOf* relationships are represented as links. Initially, the view focuses on the topic searched by the user and its direct relationships. The user can explore the ontology by expanding nodes, hiding unwanted branches, and zooming in and out. The nodes can be expanded or collapsed by left clicking on them. The user can also utilise a checkbox for highlighting the 15 key topics in the branch. This feature allows a quick identification of the most significant topics, making use of an approximate count of the relevant papers within the Rexplore dataset [7]. Right-clicking on a specific node prompts a menu containing the following two options: i) *Inspect* – This opens a sidebar window, as shown in Figure 7, providing more information about the topic (description and equivalent topics), and ii) *Explore in new page* – This redirects the user to another page where the selected topic is the central node in the graph. The user can also right-click on links, which also opens a sidebar window, to find more details about that particular relationship. The graph view is generated dynamically using the D3 library[34].

The **detailed view** presents each relevant triple in a separate row. The user can click on the name of a topic to jump to that topic page and navigate the ontology. Finally, the **compact view** shows the same information in a more condensed format, by grouping topics according to their relationship with the main one.

---

[33] The "semantic web" topic - http://cso.kmi.open.ac.uk/topics/semantic%20web.
[34] D3.js - https://d3js.org.

Whenever a concept is linked with a DBpedia concept, its page will display a short description taken from DBpedia and a hyperlink to the corresponding Wikipedia article.

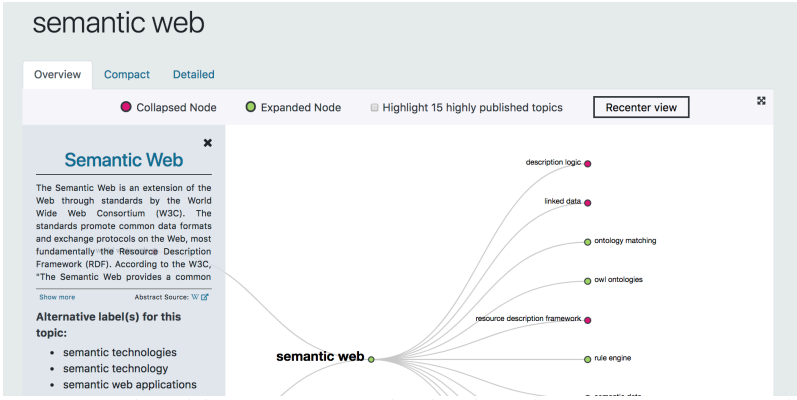

Figure 7. Screenshot of the resource page related to the topic *"semantic web"* (Overview).

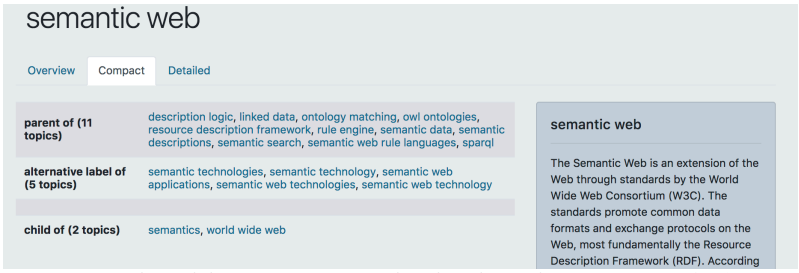

Figure 8. Screenshot of the resource page related to the topic "*semantic web"* (Compact).

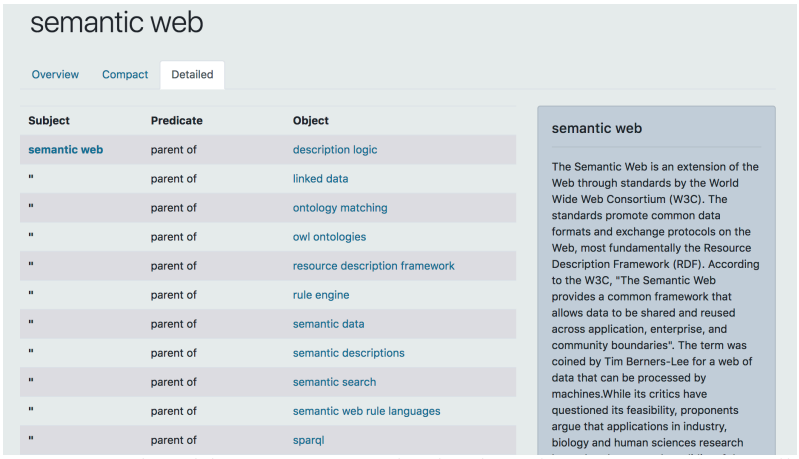

Figure 9. Screenshot of the resource page related to the topic *"semantic web"* (Detailed).

The portal supports content negotiation and yields different representations of the resources according to the content-type specified in the request. It currently supports

'*text/html*', '*application/rdf+xml*', '*text/turtle*', '*application/n-triples*', and '*application/ld+json*'.

6.2 User feedback

Registered users can provide feedback about the ontology and its relationships in all the views. In particular, users can offer feedback at i) ontology level, ii) topic level, and iii) relationship level.

The ontology level feedback is a general assessment expressing thoughts and criticisms about CSO. The user can provide it by clicking the feedback tab in the top menu and filling a text form.

Users can give feedback on specific topics by means of a form that can be triggered by clicking an icon near the topic name. Figure 10 shows as example the feedback form for the topic "ontology mapping". Users can rate the topic as "correct", "incorrect" or "is complicated" and comment their rating in a text field. In the same form, users can also suggest one or more relationships that are currently missing from the ontology or a new topic that should be linked by this relationship. Figure 11 shows the form for suggesting new relationships for the topic "ontology mapping". The users can choose the predicate from "parent of", "alternative label of", "and child of". The object could be either a topic that already exists in CSO or a new one.

Finally, users can offer feedback on specific relationships by means of an alternative form. As in the previous case, they can rate the relationship and add a short comment.

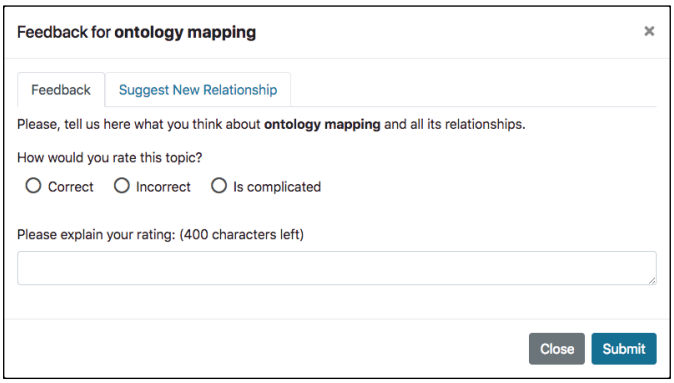

Figure 10. Form for providing feedback about the topic *"ontology mapping"*.

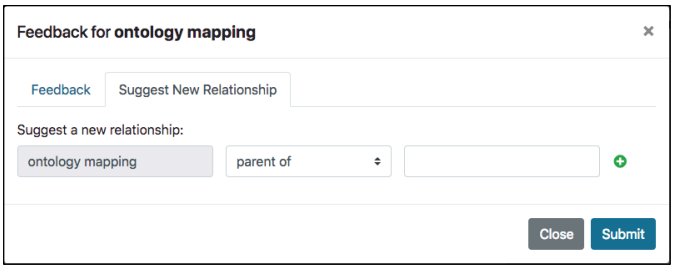

Figure 11. Form for suggesting new relationships about the topic *"ontology mapping"*.

The CSO Portal allows users to review their own feedback entries. In the "*My Contributions*" page (Figure 12) users can inspect, edit, and delete any previously given suggestion. The feedback entries are organised by typology (ontology level, topic level, relationship level, and recommendation of new relationships), and they can be either retracted or modified.

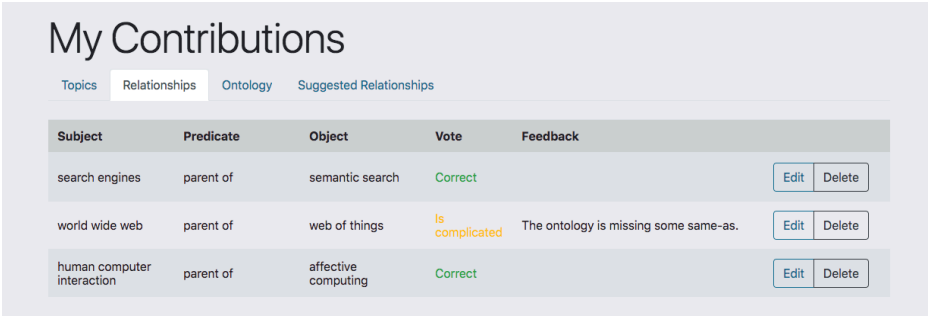

Figure 12. *My Contribution* page where users can review their own feedback.

6.3 Editorial Panel

The CSO Portal also offers a dedicated panel for the steering committee, which allow them to curate, update, and release new versions of CSO. Specifically, this panel provides a set of functionalities that enable editors to: i) add or remove research concepts, ii) add or remove relationships, iii) change the *preferentialEquivalent* topic of a cluster of *relatedEquivalent* topics, iv) check the consistency of the ontology, v) read user feedback, vi) deploy a new version of CSO, and others. Figure 13 shows a snapshot of the panel.

The Editorial Panel is an ongoing project and thus we frequently introduce new functionalities. We plan to create a collaborative environment allowing editors to discuss user feedback and evolve the ontology accordingly.

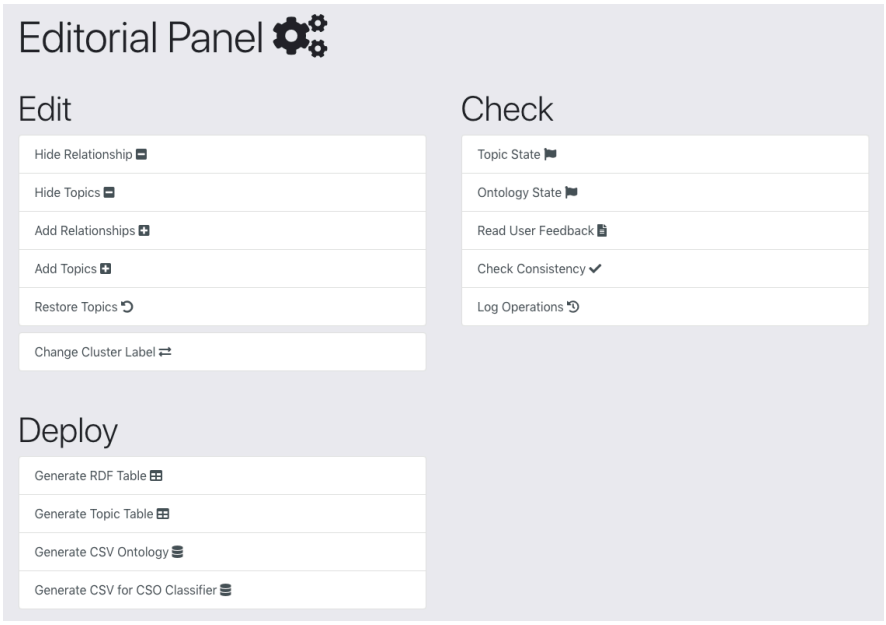

Figure 13. Screenshot of the Editorial Panel available in the CSO Portal.

6.4 Finding the Shortest Path Between Topics

The CSO Portal offers also a tool for finding the shortest paths between two research topics[35]. This tool models the ontology as a network, having topics as nodes and the *superTopicOf* relationships as links. Then, it uses the Dijkstra algorithm to identify the shortest path between the input topics. For instance, Figure 14 shows one of the paths connecting the topic "deep learning" to "blockchain".

The paths involving the topic Computer Science are generally less informative, since it is the most general concept in the ontology. Therefore, whenever all the shortest paths contain Computer Science, the application keeps searching for a shortest path that does not involve this topic. For the same reason, the paths involving Computer Science are also shown at the end of the result list.

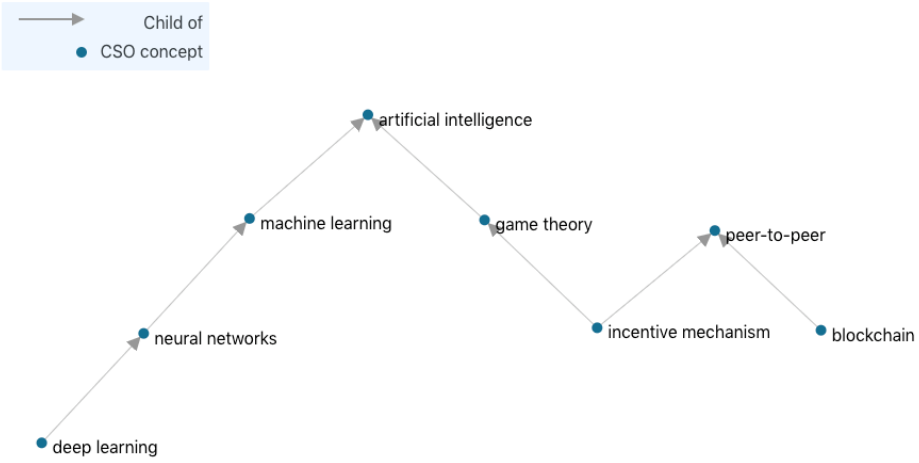

Figure 14. One of the several paths connecting Deep Learning with the field of Blockchain.

[35] Find path between topics - https://cso.kmi.open.ac.uk/findpath.

6.5 CSO Usage

While the CSO Portal was officially launched on 10 January 2019, a preliminary version has been available since April 2018, with the purpose of collecting feedback and presenting it at the International Semantic Web Conference 2018 [18]. Here, we summarise the insights and statistics that we have been gathering during this period. In particular, we recorded the downloads from October 2018 to July 2019 and the topics requested via content negotiation from July 2018 to July 2019.

CSO was downloaded 408 times in this period. Figure 15 reports the distribution of downloads on a world map, obtained by performing reverse DNS lookups on the collected IP addresses. The countries with the most downloads are USA, United Kingdom, Italy, India, Germany, France, China, Canada, Spain and Brazil.

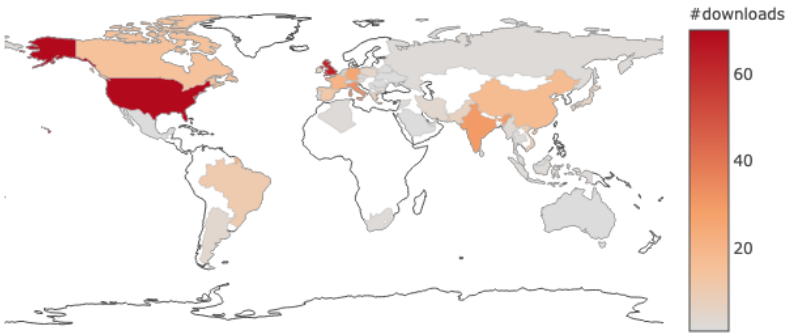

Figure 15 - Download distribution.

In the observed period, we registered a volume of 28,169 topic requests. In the majority of requests (94%), users interact with CSO in HTML via the browser, while the remaining 6% mostly consists of RDF requests (5%), with little amount for JSON-LD, turtle and triple formats (approximately 1% altogether). Figure 16 shows the distribution of users and requests on a world map. The country colour encodes the number of unique users located in a country (the brighter, the higher), while the radius of bubbles encodes the number of

requests received from a given geolocation (i.e. longitude, latitude). These results exclude search engines crawlers (e.g. Google, Yahoo, Yandex, etc.), bots and other automatically generated requests. Finally, in Figure 17, we report the total number of requests served for the top-25 topics in CSO.

An interactive version of these figures, and the code for the analysis is available at https://github.com/andremann/CSO-stats-analytics.

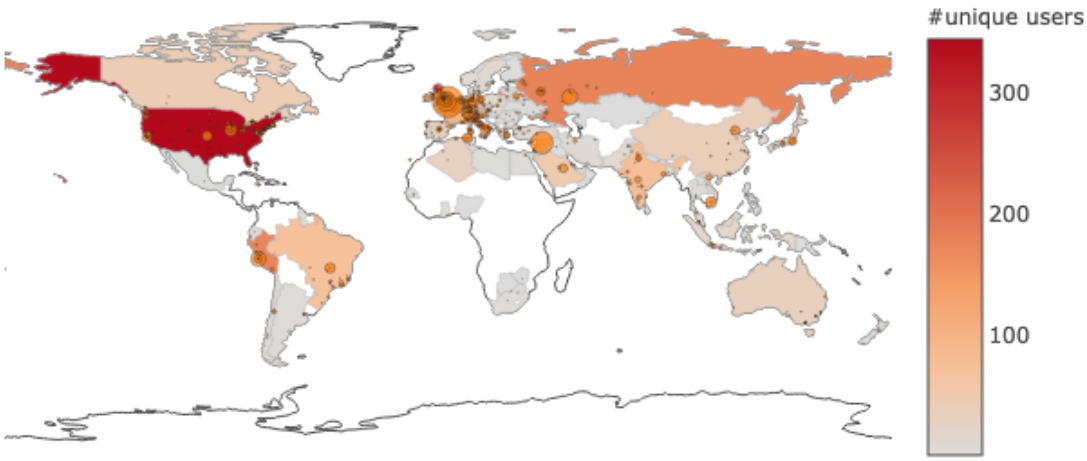

Figure 16 - Topic requests distribution.

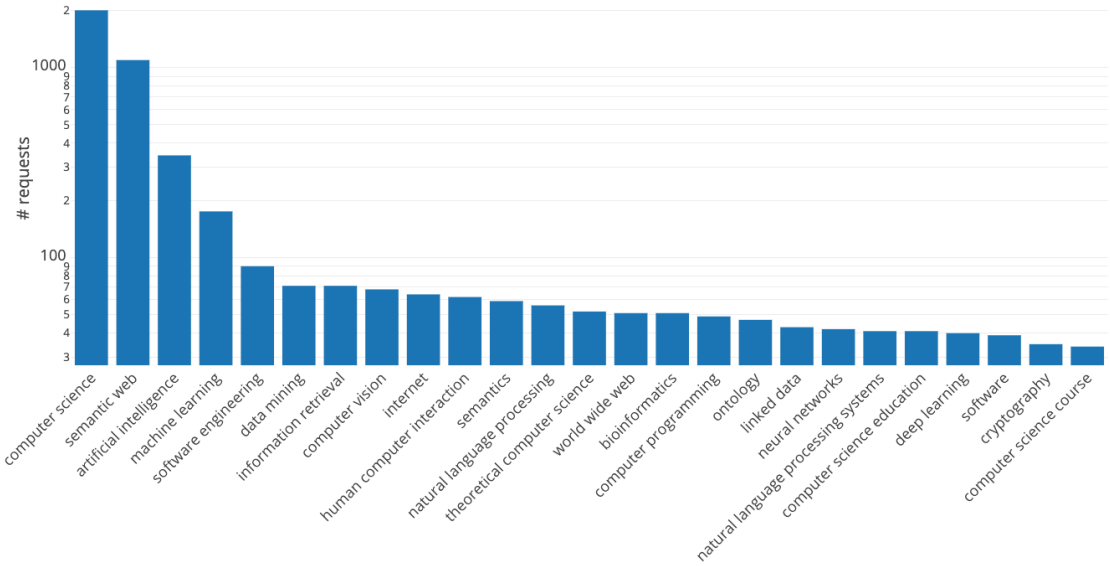

Figure 17 - Top-25 topics requested (logarithmic scale).

7 Conclusions

In this paper, we presented the Computer Science Ontology (CSO), a large-scale, automatically generated ontology of research areas, which provides a much more comprehensive and granular characterisation of research topics in Computer Science than what is currently available in other state-of-the-art taxonomies. CSO has been used to support a variety of tasks, such as classifying research papers, exploring scholarly data, forecasting new research topics, detecting research communities, and so on.

In the paper, we also introduced the CSO Classifier, a tool for automatically classifying research papers, and the CSO Portal, a web application that enables users to download, explore, and provide feedback on CSO. We intend to take advantage of the CSO Portal to involve the wider research community in the ontology evolution process, with the aim of allowing members of the community to provide feedback and incorporate such feedback in new versions of CSO. In this sense, the version of CSO presented in this paper can be considered as a starting point of this process, which integrates a fully automatic ontology generation approach with crowdsourced feedback from the community.

To this purpose, we are currently developing a new version of Klink-2 that will be able to take fully into account the crowdsourced feedback in the context of generating new versions of the CSO taxonomy. We also intend to apply our ontology learning techniques to other research fields, such as Biology and Engineering. The ultimate goal is to create a comprehensive set of large-scale topic taxonomies describing the different branches of science.

References

1. Saif, H., He, Y., Alani, H.: Semantic Sentiment Analysis of Twitter. In: The Semantic Web -- ISWC 2012. pp. 508–524. Springer, Berlin, Heidelberg (2012).
2. Ding, L., Kolari, P., Ding, Z., Avancha, S.: Using Ontologies in the Semantic Web: A Survey. In: Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems. pp. 79–113. Springer US, Boston, MA (2007).

3. Osborne, F., Salatino, A., Birukou, A., Motta, E.: Automatic Classification of Springer Nature Proceedings with Smart Topic Miner. In: International Semantic Web Conference 2016. pp. 383–399. Springer, Cham (2016).
4. Middleton, S.E., Roure, D. De, Shadbolt, N.R.: Ontology-Based Recommender Systems. In: Handbook on Ontologies. pp. 779–796. Springer Berlin Heidelberg, Berlin, Heidelberg (2009).
5. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering. In: Third IEEE International Conference on Data Mining. pp. 541–544. IEEE Comput. Soc.
6. Livingston, K.M., Bada, M., Baumgartner, W.A., Hunter, L.E.: KaBOB: ontology-based semantic integration of biomedical databases. BMC Bioinformatics. 16, 126 (2015).
7. Osborne, F., Motta, E., Mulholland, P.: Exploring Scholarly Data with Rexplore. In: International Semantic Web Conference 2013, Sydney, Australia. pp. 460–477. Springer, Berlin, Heidelberg (2013).
8. Fathalla, S., Vahdati, S., Auer, S., Lange, C.: Towards a Knowledge Graph Representing Research Findings by Semantifying Survey Articles. In: Research and Advanced Technology for Digital Libraries. pp. 315–327. Springer, Cham (2017).
9. Bettencourt, L.M.A., Kaiser, D.I., Kaur, J.: Scientific discovery and topological transitions in collaboration networks. Journal of Informetrics. 3, 210–221 (2009).
10. Osborne, F., Scavo, G., Motta, E.: Identifying diachronic topic-based research communities by clustering shared research trajectories. In: The Semantic Web: Trends and Challenges. pp. 114--129. Springer International Publishing (2014).
11. Salatino, A.A., Osborne, F., Motta, E.: AUGUR: Forecasting the Emergence of New Research Topics. In: Joint Conference on Digital Libraries 2018, Fort Worth, Texas. pp. 1–10 (2018).
12. Osborne, F., Mannocci, A., Motta, E.: Forecasting the Spreading of Technologies in Research Communities. In: Proceedings of the Knowledge Capture Conference (2017).
13. Osborne, F., Motta, E.: Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks. In: International Semantic Web Conference 2015. pp. 408–424. Springer, Cham, Bethlehem, USA. (2015).
14. Osborne, F., Motta, E.: Mining Semantic Relations between Research Areas. Presented at the (2012).
15. Osborne, F., Motta, E.: Pragmatic Ontology Evolution: Reconciling User Requirements and Application Performance. In: International Semantic Web Conference 2018. Springer, Monterey, CA (USA). (2018).
16. Osborne, F., Muccini, H., Lago, P., Motta, E.: Reducing the Effort for Systematic Reviews in Software Engineering. Data Science. (2019).
17. Thanapalasingam, T., Osborne, F., Motta, E.: Ontology-Based Recommendation of Editorial Products. In: International Semantic Web Conference 2018. , Monterey, CA (USA). (2018).
18. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas. In: International Semantic Web Conference 2018. pp. 187–205. Springer, Cham, Monterey, USA (2018).
19. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. Journal of Web Semantics. 7, 154–165 (2009).
20. Vrandečić, D., Krötzsch, M.: Wikidata: A Free Collaborative Knowledgebase. Communications of the ACM. 57, 78–85 (2014).
21. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia.
22. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08. p. 1247. ACM Press, New York, New York, USA (2008).
23. Lenat, D.B., Guha, R. V: Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project *. In: Artificial Intelligence. pp. 95–104 (1993).
24. Salatino, A.A., Osborne, F., Thanapalasingam, T., Motta, E.: The CSO Classifier: Ontology-Driven Detection of Research Topics in Scholarly Articles. In: TPDL 2019: 23rd International Conference on Theory and Practice of Digital Libraries (2019).
25. Osborne, F., Motta, E.: Exploring research trends with rexplore. D-Lib Magazine. 19, (2013).
26. Boyack, K.W., Newman, D., Duhon, R.J., Klavans, R., Patek, M., Biberstine, J.R., Schijvenaars, B., Skupin, A., Ma, N., Börner, K.: Clustering More than Two Million Biomedical Publications: Comparing the Accuracies of Nine Text-Based Similarity Approaches. PLoS ONE. 6, e18029 (2011).
27. Lipscomb, C.E.: Medical Subject Headings (MeSH). Bulletin of the Medical Library Association. 88, 265–6 (2000).
28. Cherrier, B.: Classifying Economics: A History of the JEL Codes. Journal of Economic Literature. 55, 545–579 (2017).
29. Clough, P., Sanderson, M., Gollins, T.: Examining the Limits of Crowdsourcing for Relevance Assessment. IEEE Internet Computing. 17, 32–38 (2013).
30. Cimiano, P., Völker, J.: Text2Onto. In: Natural Language Processing and Information Systems. pp. 227–238. Springer, Berlin, Heidelberg (2005).
31. Muller, A., Dorre, J., Gerstl, P., Seiffert, R.: The TaxGen framework: automating the generation of a taxonomy for a large document collection. In: Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers. p. 9. IEEE Comput. Soc (1999).
32. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99. pp. 206–213. ACM Press, New York, New York, USA (1999).
33. Shen, Z., Ma, H., Wang, K.: A Web-scale system for scientific knowledge exploration. In: Proceedings of ACL 2018, System Demonstrations. pp. 87–92. Association for Computational Linguistics, Melbourne, Australia

(2018).
34. Wohlgenannt, G., Weichselbraun, A., Scharl, A., Sabou, M.: Dynamic Integration of Multiple Evidence Sources for Ontology Learning. Journal of Information and Data Management. 3, 243–254 (2012).
35. Mortensen, J.M., Musen, M.A., Noy, N.F.: Crowdsourcing the verification of relationships in biomedical ontologies. AMIA Annual Symposium Proceedings. 2013, 1020–9 (2013).
36. Kirrane, S., Sabou, M., Fernández, J.D., Osborne, F., Robin, C., Buitelaar, P., Motta, E., Polleres, A.: A decade of Semantic Web research through the lenses of a mixed methods approach. Submitted to Semantic Web Journal. (2019).
37. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Proceedings of the 9th International Conference on Semantic Systems - I-SEMANTICS '13. p. 121. ACM Press, New York, New York, USA (2013).
38. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems. 5, 1–22 (2009).
39. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: Classifying research papers with the computer science ontology. In: In International Semantic Web Conference (P&D/Industry/BlueSky). CEUR Workshop Proceedings, vol. 2180. (2018).
40. Salatino, A.A., Osborne, F., Birukou, A., Motta, E.: Improving Editorial Workflow and Metadata Quality at Springer Nature. In: The Semantic Web – ISWC 2019. Springer Verlag (2019).
41. Charlin, L., Zemel, R.S.: The Toronto Paper Matching System: An automated paper-reviewer assignment system. (2013).
42. Cano-Basave, A.E., Osborne, F., Salatino, A.A.: Ontology forecasting in scientific literature: Semantic concepts prediction based on innovation-adoption priors. In: Knowledge Engineering and Knowledge Management. pp. 51–67 (2016).
43. Mannocci, A., Osborne, F., Motta, E.: The Evolution of IJHCS and CHI: A Quantitative Analysis. International Journal of Human-Computer Studies. (2019).
44. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research. 3, 993–1022 (2003).
45. Monaghan, F., Bordea, G., Samp, K., Buitelaar, P.: Exploring Your Research: Sprinkling some Saffron on Semantic Web Dog Food. In: Semantic Web Challenge at the International Semantic Web Conference (Vol. 117, pp. 420-435) (2010).