

Open Research Online

The Open University's repository of research publications and other research outputs

Classifying Research Papers with the Computer Science Ontology

Conference or Workshop Item

How to cite:

Salatino, Angelo; Thanapalasingam, Thiviyan; Mannocci, Andrea; Osborne, Francesco and Motta, Enrico (2018). Classifying Research Papers with the Computer Science Ontology. In: ISWC 2018 Posters & Demonstrations and Industry Tracks (van Erp, Marieke ed.).

For guidance on citations see [FAQs](#).

© [not recorded]

Version: Not Set

Link(s) to article on publisher's website:

<http://iswc2018.semanticweb.org/sessions/classifying-research-papers-with-the-computer-science-ontology/>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

Classifying Research Papers with the Computer Science Ontology

Angelo A. Salatino, Thiviyan Thanapalasingam, Andrea Mannocci,
Francesco Osborne, Enrico Motta

Knowledge Media Institute, The Open University, MK7 6AA, Milton Keynes, UK
`{firstname.lastname}@open.ac.uk`

Abstract. Ontologies of research areas are important tools for characterising, exploring and analysing the research landscape. We recently released the Computer Science Ontology (CSO), a large-scale, automatically generated ontology of research areas, which includes about 26K topics and 226K semantic relationships. CSO currently powers several tools adopted by the Springer Nature editorial team and has been used to enable a variety of solutions, such as classifying research publications, detecting research communities, and predicting research trends. As an effort to encourage the usage of CSO, we have developed the CSO Portal, a web application that enables users to download, explore, and provide granular feedbacks at different levels of the ontology. In this paper, we present the *CSO Classifier*, an application for automatically classifying academic papers according to the rich taxonomy of topics from CSO. The aim is to facilitate the adoption of CSO across the various communities engaged with scholarly data and to foster the development of new applications based on this knowledge base.

Keywords: Scholarly Data, Ontology Learning, Scholarly Ontologies, Text Mining, Topic Detection, Taxonomy, Classifier, Web Application.

1 Introduction

Ontologies are powerful tools for representing domain knowledge, integrating data from different sources, and supporting a variety of semantic applications. In the scholarly domain, ontologies support the integration of large datasets of research data, information extraction from scientific articles, the exploration of the academic landscape, and so on. In particular, ontologies describing research areas and their relationships are invaluable tools for: i) making sense of the research dynamics, ii) classifying publications, iii) identifying research communities, and iv) forecasting research trends.

In a recent paper [1], we presented the *Computer Science Ontology (CSO)*, a large-scale, granular, and automatically generated ontology of research areas which includes about 26K semantic topics and 226K relationships. CSO currently supports a range of applications including Smart Topic Miner [2], a tool designed to assist the Springer Nature editorial team in classifying proceedings, and Smart Book Recommender [3], an ontology-based recommender system for selecting books to market at academic venues. A comprehensive list of applications and approaches building on CSO is available in [1].

We released CSO through the *CSO Portal*¹, a web application that enables users to download, explore, and provide feedback on CSO. The aim was to make available to all the relevant communities an open knowledge base for supporting the development of further applications. However, many users interested in adopting CSO for characterizing their data have limited understanding of semantic technologies and how to use an ontology for annotating documents. They thus need a simple solution for classifying research papers.

In this demo paper, we briefly introduce 1) the CSO Classifier, an application for classifying academic documents according to CSO, and 2) a web application² that exploits the CSO Classifier for annotating research papers. This tool is meant to allow researchers and developers to easily adopt CSO for their own applications.

2 The Computer Science Ontology

The Computer Science Ontology (CSO) is a large-scale ontology of research areas that was automatically generated using the Klink-2 algorithm [4] on a dataset of 16 million publications, mainly in the field of Computer Science [5]. Differently from other solutions available in the state of the art, CSO includes a much larger number of fine-grained research topics, enabling a granular characterisation of the content of research papers, and it can be easily updated by running Klink-2 on recent corpora of publications.

The current version of CSO³ includes 26K semantic topics and 226K relationships. The main root is Computer Science; however, the ontology includes also a few secondary roots, such as Linguistics, Geometry, Semantics, and so on.

More information about CSO, its data model, its semantic relations, and how it was generated are reported in [1].

3 The CSO Classifier

The CSO Classifier is an application that classifies the content of scientific papers (i.e., full-text, abstract, and title) according to CSO. Specifically, given a research paper, the classifier takes as input text from its abstract or full-text and outputs a list of relevant concepts from CSO. It does so by mapping the n-grams in the text to concepts in the CSO and then inferring their super concepts. It accepts four optional parameters:

- *min_similarity*, which controls the minimum similarity value for mapping n-grams to concepts.
- *infer_super_topics*, which controls whether the classifier will try to infer, given a topic (e.g., Linked Data), only the direct super-topics (e.g., Semantic Web) or all its super-topics (e.g., Semantic Web, WWW, Computer Science).
- *num_children*, which controls the number of concepts necessary for inferring a super concept. For example, when this factor is set to three, the topic

¹ CSO Portal: <http://w3id.org/cso> or <https://cso.kmi.open.ac.uk>

² CSO Classifier web application: <https://cso.kmi.open.ac.uk/classify>

³ Computer Science Ontology available for download at <https://w3id.org/cso/downloads>

Semantic Web will be inferred if at least three of its sub-topics (e.g., OWL, RDF, Linked Data) are present.

- *verbose*, is a flag controlling the verbosity level of the result.

The CSO Classifier removes English stop words and it gathers together unigrams, bigrams and trigrams. Then, for each n-gram, it computes the Levenshtein similarity with the labels of the topics in CSO. Research topics having similarity equal or higher than the minimum similarity threshold with an n-gram, are added to the final set of topics. In order to further enrich the set of inferred topics, the CSO Classifier infers also their super topics by exploiting the *skos:broaderGeneric* relationships within the CSO [1]. The output of this process can contain equivalent topics linked by *relatedEquivalent* relationships in CSO, e.g., Ontology Matching and Ontology Mapping. Therefore, the CSO Classifier also clean up these redundant concepts by preserving only one of them.

The algorithm produces two kinds of result, depending on the *verbose* parameter. When it is set to true, the algorithm returns a detailed list of topics, with the matched n-grams and the evaluated similarity scores. Conversely, if *verbose* is set to false, the algorithm returns a more synthetic list of topics.

The CSO Classifier was developed in Python and the open-source codebase is available on a GitHub repository: <https://github.com/angelosalatino/cso-classifier>.

4 CSO Classifier as Web Application

The CSO Classifier has been deployed as a web application within the CSO Portal. Its user interface consists of a web form in which users can provide either 1) a paper metadata (i.e., title, abstract, and keywords), 2) the DOI of a research paper, or 3) generic text (e.g., the abstract of a paper). Figure 1 displays the form for providing title, abstract and keywords, pre-filled with the metadata of this demo paper.

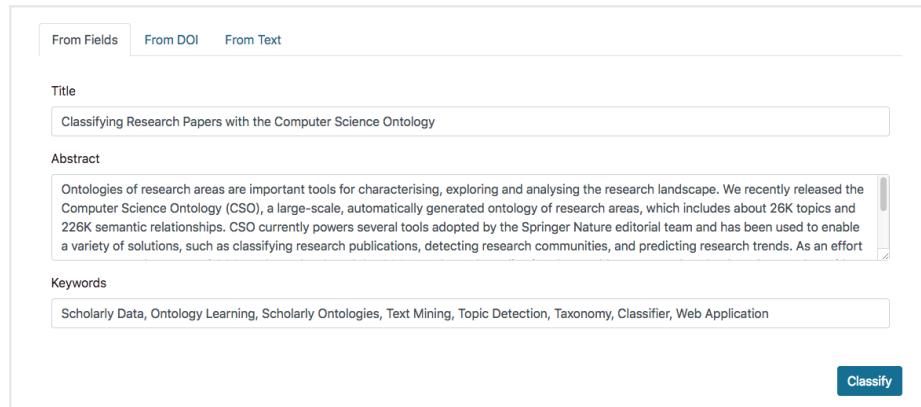


Figure 1. Screenshot of the form to classify research papers within the CSO Portal.

Once the classification process is complete, the web application returns the topics inferred from the document alongside the values of similarity between topics and n-grams. In addition, it displays the classified document with an annotation layer containing the topics linked to the CSO Portal (see Figure 2).

Annotated document:

Classifying Research Papers with the **Computer Science Ontology**

Abstract: **Ontologies** of research areas are important tools for characterising, exploring and analysing the research landscape. We recently released the **Computer Science Ontology** (CSO), a large-scale, automatically generated **ontology** of research areas, which includes about 26K topics and 226K **semantic relationships**. CSO currently powers several tools adopted by the Springer Nature editorial team and has been used to enable a variety of solutions, such as classifying research publications, detecting research communities, and **predicting** research trends. As an effort to encourage the usage of CSO, we have developed the CSO Portal, a **web application** that enables users to download, explore, and provide granular **feedbacks** at different levels of the **ontology**. In this paper, we present the CSO **Classifier**, an application for automatically classifying academic papers according to the rich **taxonomy** of topics from CSO. The aim is to facilitate the adoption of CSO across the various communities engaged with scholarly data and to foster the **development** of new applications based on this **knowledge base**.

Keywords: Scholarly Data, **Ontology Learning**, Scholarly **Ontologies**, **Text Mining**, Topic Detection, **Taxonomy**, **Classifier**, **Web Application**

Figure 2. A paper metadata annotated by the CSO classifier.

5 Conclusions

In this paper, we presented the CSO Classifier, a tool that automatically classifies text according to the Computer Science Ontology. In [1], we showed how CSO supports several useful tasks, such as classifying research papers, exploring scholarly data, forecasting new research topics, detecting research communities, and so on. The CSO Classifier presents an initial step towards the establishment of new applications based on this knowledge base. We intend also to take advantage of the CSO Classifier, alongside the CSO Portal, to involve a wider research community in the ontology evolution process, with the aim of periodically releasing up-to-date revisions of CSO and allowing members of the community to provide feedback.

As future work, we plan to enhance the method for mapping concepts to n-grams by adopting Name Entity Recognition techniques for extracting acronyms and considering also the context surrounding a word.

References

1. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas. In: The Semantic Web -- ISWC 2018. Pre-Print: <http://oro.open.ac.uk/55484/> (2018).
2. Osborne, F., Salatino, A., Birukou, A., Motta, E.: Automatic Classification of Springer Nature Proceedings with Smart Topic Miner. *Semant. Web -- ISWC 2016*. 9982 LNCS, 383–399 (2016).
3. Thanapalasingam, T., Osborne, F., Birukou, A., Motta, E.: Ontology-Based Recommendation of Editorial Products. In: *International Semantic Web Conference 2018*. , Monterey, CA (USA) (2018).
4. Osborne, F., Motta, E.: Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks. In: *The Semantic Web - ISWC 2015*. pp. 408–424 (2015).
5. Osborne, F., Motta, E., Mulholland, P.: Exploring scholarly data with rexplore. *Semant. Web -- ISWC 2013*. 8218 LNCS, 460–477 (2013).