Software Impacts 14 (2022) 100444

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

R-classify: Extracting research papers' relevant concepts from a controlled vocabulary

Tanay Aggarwal, Angelo Salatino *, Francesco Osborne, Enrico Motta

KMi, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK

ARTICLE INFO

Keywords:
Topic detection
Topic extraction
Scholarly data
Science of science
Text mining
Scholarly ontologies

ABSTRACT

In the past few decades, we saw a proliferation of scientific articles available online. This data-rich environment offers several opportunities but also challenges, since it is problematic to explore these resources and identify all the relevant content. Hence, it is crucial that they are appropriately annotated with their relevant concepts so to increase their chance of being properly indexed and retrieved. In this paper, we present R-Classify, a web tool that assists users in identifying the most relevant concepts according to a large-scale ontology of research areas in the field of Computer Science.

Code metadata

Current code version	*v1.0*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-247
Permanent link to Reproducible Capsule	
Legal Code License	Apache License 2.0
Code versioning system used	*git*
Software code languages, tools, and services used	*Languages: Python, JavaScript, PHP, HTML, CSS, Java. Framework: Django. DataBase: MongoDB. Additional tool: Grobid.*
Compilation requirements, operating environments & dependencies	All necessary requirements are listed in the file *requirements.txt*. See documentation to automatically install them.
If available Link to developer documentation/manual	https://github.com/angelosalatino/r-classify/blob/master/README.md
Support email for questions	angelo.salatino@open.ac.uk

1. Introduction

In recent years, the scientific community has produced and circulated knowledge at an unprecedented rate. It is currently estimated that more than two million research papers are published each year [1]. Consequently, it has become challenging to navigate and search such a deluge of documents. On several occasions, web search engines struggle to find relevant information, which eventually leads to frustrating and dissatisfying experiences [2]. This issue is further intensified by the lack of specific guidelines for defining the set of subject areas or keywords for annotating research documents. These are typically manually chosen by researchers or librarians, who however may use very different styles and levels of granularity, resulting in a very sparse and noisy representation. For instance, they typically use a large variety of syntactical forms for the same concept, e.g., *peer to peer*, *peer2peer*, *p2p*, *peer to peer networks*, *peer to peer systems*.

In order to improve the retrievability of research content it is necessary to annotated the documents with a set of concepts that are (i) *standardised*, (ii) *high-quality*, and (iii) *comprehensive*. Such a representation can supports more effectively digital libraries, search engines, and recommendation systems [3,4]. It can also facilitate scientometrics analyses and systems for monitoring and predicting research trends [5].

In this paper, we present R-Classify, a novel web application for assisting users in selecting the best set of research topics to describe a scientific article. The primarily aim is to help researchers in improving the quality of the keywords they use to annotate papers. However, it can also be used by librarians, publishing editors, and many other

* Corresponding author.
E-mail addresses: tanay.aggarwal@open.ac.uk (T. Aggarwal), angelo.salatino@open.ac.uk (A. Salatino), francesco.osborne@open.ac.uk (F. Osborne), enrico.motta@open.ac.uk (E. Motta).

https://doi.org/10.1016/j.simpa.2022.100444
Received 28 October 2022; Accepted 7 November 2022

stakeholders that need a good description of the topics within a technical text. R-Classify builds on the CSO Classifier, which is a tool for automatically classifying research papers in terms of relevant topics drawn from the Computer Science Ontology [6]. The Computer Science Ontology (CSO) describes more than 14K research concepts in the field of Computer Science, organising them in a hierarchical structure with parent–child relationships [7].

The CSO Classifier is available as a Python library, and in the last few years it has been adopted by several organisations for classifying research articles, patents, technical documents, research reports, and course material. [8–10]. For instance, it is currently being employed by Springer Nature to improve the metadata quality of their conference proceedings [3]. A recent evaluation against 22 approaches on a gold standard of research papers in Computer Science shows that the classifier provides a high-quality and accurate representation of the salient topics within the research papers [11]. R-Classify aims at making the CSO Classifier easily available to all users, regardless of their technical skills.

R-Classify is available via browser at https://w3id.org/cso/classify. The interface allows users to either input an arbitrary text (typically title and abstract of a scientific paper) or upload a PDF file. In the latter case, the tool leverages GROBID [12] to extract the title and abstract from the PDF file. Once the classification is complete, the interface displays a list of concepts from the CSO ontology, as well as the annotated text (see Fig. 3). At this stage, the users can refine and export the returned concepts that will typically be used as keywords for the input research paper.

As a result, the resulting set of concepts is *standardised* through the Computer Science Ontology; is of *high-quality* thanks to capabilities of the CSO Classifier; and it *comprehensively* describes the content of the documents as consequence of the large coverage of CSO and the refinement of the user.

The code is open-source with an Apache License 2.0 and it is available at https://github.com/angelosalatino/r-classify.

The paper is structured as follows. In Section 2 we present the main architecture of R-Classify, explaining all its components in detail. In Section 3 we describe the impact of this tool. Finally, in Section 4, we discuss future line of research and development.

2. Architecture

R-Classify is a web application assisting researchers in annotating their papers with a comprehensive selection of research concepts. It takes as input title and abstract of a paper and returns (i) a set of relevant research topics and (ii) an annotated version of the input text, identifying the portions of text that triggered each topics.

Fig. 1 depicts the architecture of R-Classify. It consists of five main components. The user interface allows users to interact with the tool by loading the data, analysing and modifying the resulting topics, and exporting the result. The back-end engine deals with the user interactions and orchestrates the classification process. GROBID extracts the text of the papers from PDF files. The CSO Classifier parses the text and extracts the relevant topics. Finally, the database keeps a record of the classification results and the chosen concepts, with the aim of improving the performance of the classifier. In the following subsections, we will describe all the components in detail.

2.1. Back-end

The back-end is the core engine that manages the various modules. It is developed in Python using Django, an open framework that allows HTML pages to embed Python code.

The back-end is responsible for dealing with the user requests from the web interface by using the relevant modules. Whenever the user loads a PDF file, the back-end sends it to the GROBID module to extract the raw text and then deletes the file. After the text is ready to be processed, the back-end sends it to the CSO Classifier and return to the user the annotated document. Finally, the back-end takes care of the communication with the MongoDB database and saving the text and the topics produced for each document.

The back-end has been designed and implemented in a modular way to enable future extensions. In particular, we plan to incorporate additional classifiers covering other fields of science.

2.2. The CSO classifier

The CSO Classifier is an open-source Python tool[1] that we developed for classifying documents according to the research topics within the Computer Science Ontology (CSO) [7]. In this context, the CSO serves as a controlled vocabulary of 14K research topics (e.g., *blockchain*, *deep learning*) within the domain of Computer Science. To make this paper self-contained, we will give a brief overview of the classifier. The interested reader can refer to [11] for additional details.

The CSO Classifier consists of three sequential modules: (i) *syntactic*, (ii) *semantic*, and (iii) *post-processing*. The syntactic module finds all topics in the ontology that are explicitly mentioned in the paper. First, it extracts unigrams, bigrams, and trigrams. Then, it uses the Levenshtein similarity to compare each n-gram against all topics in CSO. As output, this module returns the topics that have a similarity greater than or equal to a predefined threshold.

The semantic module identifies further topics by computing the semantic similarity between the terms in the document and the CSO topics. To do this, it leverages a part-of-speech tagger and a pre-trained word embeddings model, which is able to capture the semantic properties of words. In particular, with a part-of-speech tagger, the semantic module identifies candidate terms that are a combination of nouns and adjectives, and then it decomposes them into unigrams, bigrams, and trigrams. Next, for all the extracted n-grams it retrieves the most similar topics from the word embeddings model. Then, it ranks and selects the most relevant ones by computing their relevance score. Specifically, it computes the product between the number of times a topic was identified in the input text and the number of unique terms that led to it. Finally, it selects the most relevant topics with the elbow method.

The post-processing module takes the union of the topics returned by the two previous modules, as this solution maximises the f-score, based on our previous experiments [11]. Next, it identifies outlier topics that appear to be irrelevant in the context of the analysed document. Specifically, the classifier computes the pairwise similarity between topics, considering both the length of their shortest paths within CSO and their similarity within the word embeddings model. It then discards the topics that are the least similar with the others. Finally, it leverages the hierarchical relationships in CSO, to further enhance the resulting set of topics by including their broader topics. For instance, a paper tagged with *deep learning* is also tagged with *artificial intelligence*. Indeed, this solution yields a comprehensive characterisation of high-level topics that are not explicitly mentioned in the documents.

2.3. GROBID

R-Classify employs GROBID to extract title and abstract from PDF files. GROBID stands for *GeneRation Of BIbliographic Data*[2] and is a machine learning tool for parsing technical and scientific publications in PDF format. It automatically extracts information such as metadata, structured text, and bibliographic Refs. [12]. It is developed in Java and offers a REST API that can be used to POST PDF documents and retrieve a representation of the document according to the TEI (Text Encoding

[1] Download CSO Classifier from Pypi - https://pypi.org/project/cso-classifier/

[2] GROBID - https://github.com/kermitt2/grobid

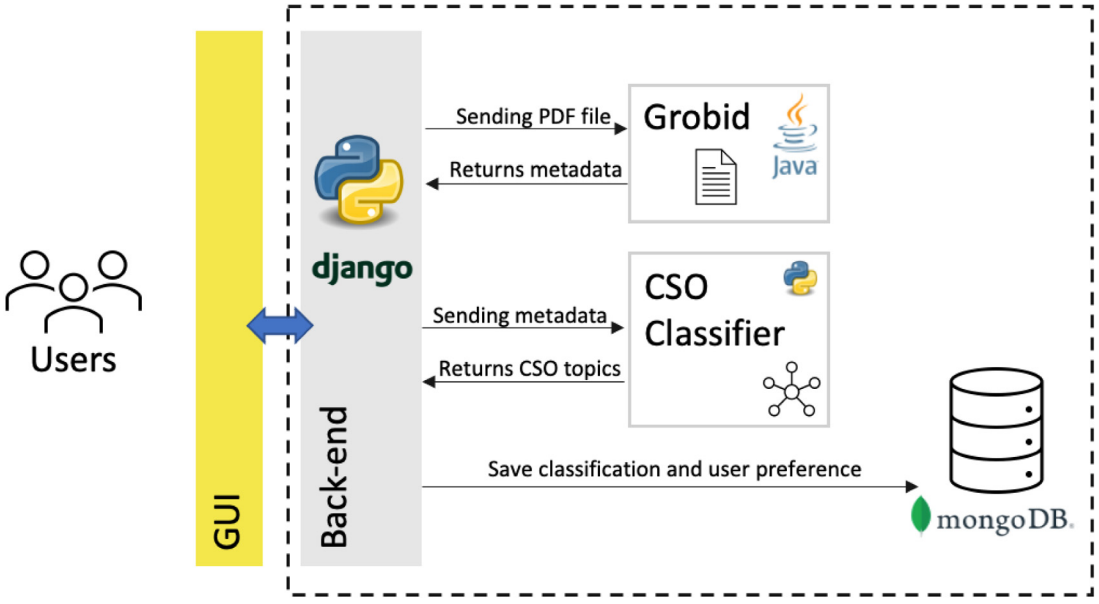

Fig. 1. The R-Classify architecture.

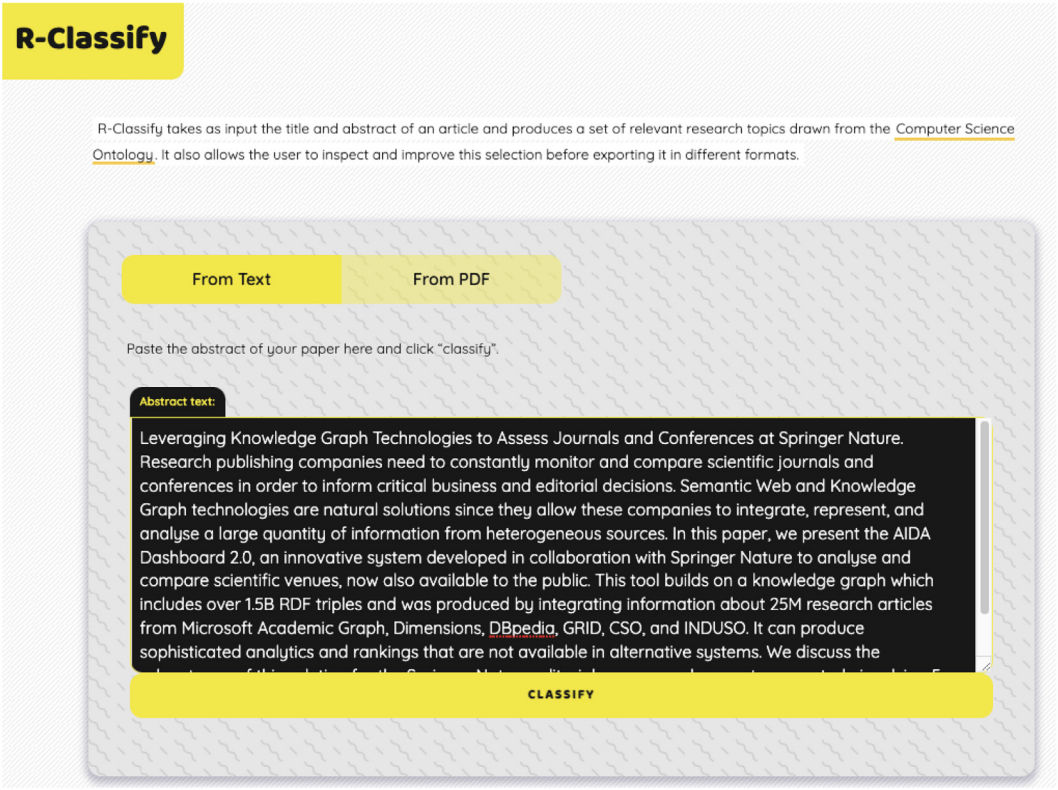

Fig. 2. Landing page of R-Classify.

Initiative) standard. In particular, GROBID is able to extract up to 55 different features, including publication metadata (such as title, authors, abstract, keywords, affiliations, affiliation addresses, journal name, and others) as well as full-text structures (section titles, reference markers, paragraphs, footnotes, captions and so on).

Previous experiments [13] show that GROBID obtains high values of precision (85.11%) and recall (83.8%) and is able to outperforms several alternative tools. For this reason, it is currently adopted by several online tools, including ResearchGate, Mendeley, scite.ai, and Academia.edu.

2.4. Database

R-Classify uses an instance of MongoDB, a popular document-oriented NoSQL database, to store relevant information regarding the various classifications performed on the server. The aim is to use the

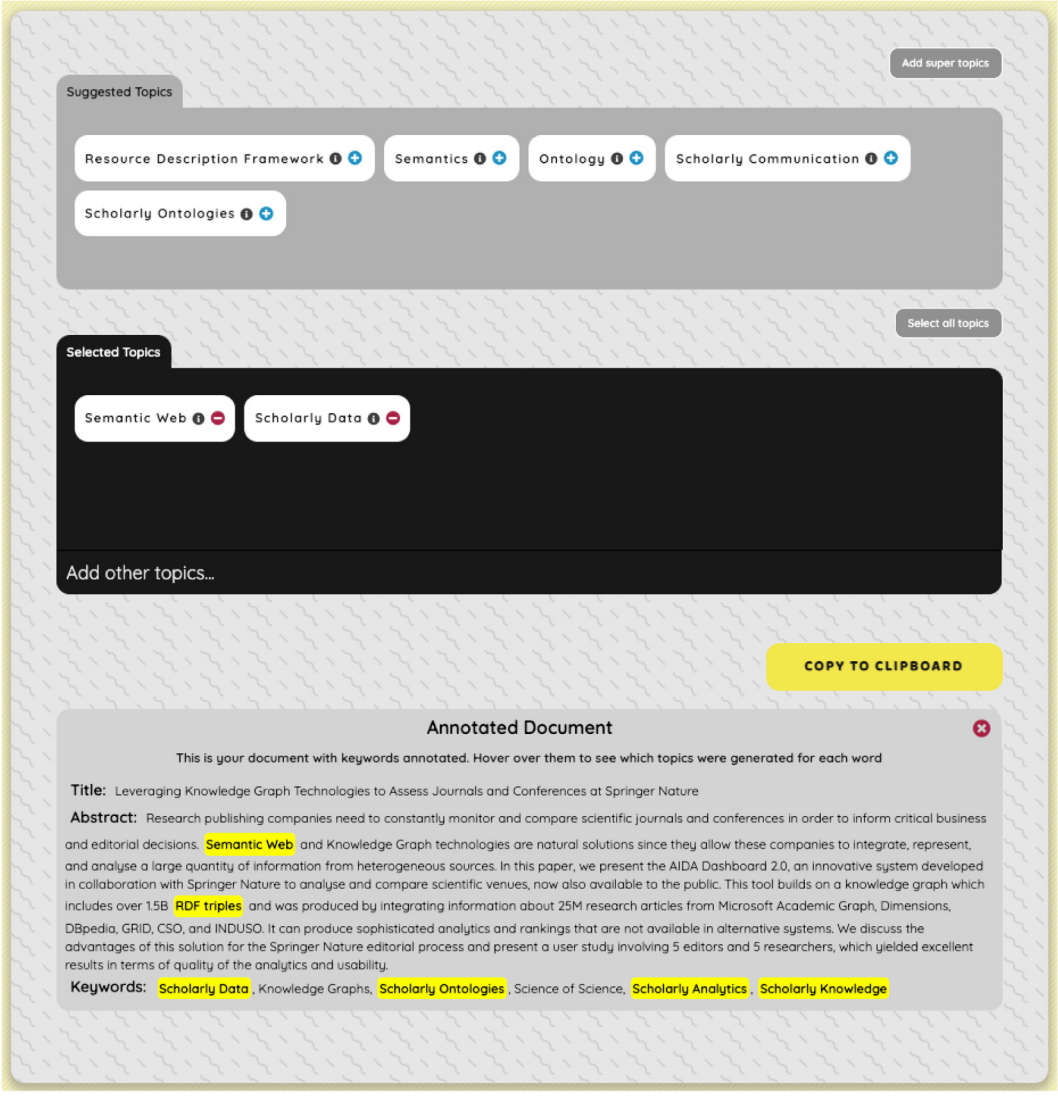

Fig. 3. Results of the classification.

user feedback on the topics to assess and improve the classification process.

In particular, for each classified document, R-Classify saves a JSON record with the following information:

- **User Content**: the title and abstract of the input document;
- **Topics Generated**: the topics returned by the CSO Classifier;
- **Topics Chosen**: the topics chosen by the user;
- **Topics Added**: additional topics chosen by the user that are not returned by the CSO Classifier.

We use the portion of topics selected and exported by the user to build and improve our gold standard. This will be used to test and enhance future versions of the CSO Classifier as well as for developing new classifiers. In addition, we save the IP address and the timestamp for a limited time in order to assess the usage of the tool over time and across different countries.

As we process also research papers that are not published yet (e.g., during the submission or camera-ready phase), some concerns may arise regarding our use of the parsed research documents. It is thus important to clarify that R-Classify neither extracts nor retains additional information available in PDF documents, such as authors, affiliation, full text. Furthermore, PDF files are deleted soon after being processed by GROBID to protect the intellectual property of our users. All the collected data and the MongoDB instance reside within a server behind a firewall and will not be shared to other teams/researchers outside The Open University.

2.5. Graphical user interface

R-Classify holds an interactive and dynamic user interface. Once the web app is loaded within the browser, the user can either input the text or load a PDF file, as shown in Fig. 2. In the latter case, R-Classify will run GROBID and display the research paper metadata back to the user, asking for confirmation. The user can then proceed with the classification, by clicking on the R-Classify button.

After a few seconds, the app will display the output of the classification, as shown in Fig. 3. This consists of three main sections: (i) Suggested Topics, (ii) Selected Topics, and (iii) Annotated Document.

The Suggested Topics lists all the topics returned by the classifier. Within this set it is possible to further add all their super-topics according to CSO by clicking on the "Add super topics" button. For instance, if "Neural Networks" is one of the topics the system will add the high-level topics "Machine Learning" and "Artificial Intelligence".

The Selected Topics section is the basket containing all the topics chosen by the user. Topics can be moved from the suggested to the selected topics with a simple drag and drop, or clicking on blue plus-icon (+) button available on each individual topic. The user can also

accept the topics by using the "Select all topics" option. Each topic has an information icon (i) button, which leads to the CSO Portal[3] page describing the topic in detail. The user can also choose to manually add a topic by using the "Add other topics" button.

The Annotated Document section, at the bottom of Fig. 3, shows an annotated version of the original documents. This view highlights all the portions of text that were used to identify the topics. Moving the mouse over a highlighted portions will display the topics that were inferred from that chunk of text.

To export the chosen topics, the user can click on "Copy to Clipboard" which will copy them as comma separated values.

3. Impact

Since their release, both the Computer Science Ontology and the CSO Classifier have received a growing attention. In particular, they are being used by several applications and proved to effectively support a wide range of tasks, such as exploring and analysing scholarly data (e.g., Rexplore [14], ScholarLensViz [15], ConceptScope [10]), detecting research communities (e.g., Temporal Semantic Topic-Based Clustering [16], Research Communities Map Builder [17]), identifying domain experts (e.g., VeTo [18]), recommending articles [4] and video lessons [19], generating knowledge graphs (e.g., Temporal KG [20], AIDA KG [21], AI KG [22], CS KG [23]), and topic models (e.g., Co-CoNoW [9]).

The Computer Science Ontology, has also been adopted by a range of systems for forecasting: (i) academic impact (e.g., ArtSim [8], Augur [5]), (ii) research topics (e.g., Augur [5]), (iii) ontology concepts (e.g., SIM [24], POE [25]), and (iv) technologies (e.g., TTF [26], TechMiner [27]).

Moreover, CSO and the CSO Classifier both support several applications used by the Springer Nature editorial team, such as Smart Topic Miner [3], a tool for assisting the classification of proceedings books, and the Smart Book Recommender [4], a recommender systems for scientific volumes.

R-Classify was developed to reach an even wider set of users and in particular those that may not be technically savvy. Indeed, with the app being online and ready-to-go, it can be used by any user without specific technical and coding skills. It is aimed specifically to researchers that want to identify the appropriate set of concepts for their research documents before submitting it to the conference or journal. Workshop and conference organisers can also encourage authors to submit their manuscripts with well formed topics extracted by R-Classify in order to gain a better understanding of the research areas in the articles and improving the quality of the metadata. Indeed, we are now collaborating with several organisers of academic events (e.g., SciK, RefResh, Text2KG, DL4KG, SEMANTiCS) who plan to request authors to annotate their manuscript with R-Classify before submission. Editors and editorial assistants can also leverage R-Classify to organise content within conference proceedings or journal issues. Finally, it can be used by librarians who need to produce rich descriptions of the deposited manuscripts in order to improve their findability in digital libraries.

4. Conclusions and future work

In this paper, we introduced R-Classify, a web application that supports researchers, conference organisers, librarians, and editors in extracting the most salient concepts from research documents. It integrates the CSO Classifier, a tool for classifying research documents according to topics within the Computer Science Ontology, and GROBID, a machine learning tool for extracting metadata from research papers in PDF.

As future work, we plan to work on multiple fronts. First, we will gather feedback from the community of users and improve the interface and functionalities accordingly. We also intend to further improve the performance of the classifier by taking advantage of the data collected by R-Classify and integrating recent NLP solutions based on transformers. Finally, we plan to expand the classifier coverage by considering other classification schemes both in Computer Science (e.g., ACM Computing Classification System[4]) and in other fields. As now, we are currently working on new models for classifying research papers in the fields of Engineering and Biomedicine.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Santo Fortunato, et al., Science of science, Science 359 (6379) (2018) eaao0185, http://dx.doi.org/10.1126/science.aao0185, URL https://www.science.org/doi/abs/10.1126/science.aao0185, arXiv:https://www.science.org/doi/pdf/10.1126/science.aao0185.

[2] Daan Odijk, et al., Struggling and success in web search, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450337946, 2015, pp. 1551–1560, http://dx.doi.org/10.1145/2806416.2806488.

[3] Angelo A. Salatino, et al., Improving editorial workflow and metadata quality at springer nature, in: Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, Fabien Gandon (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham, ISBN: 978-3-030-30796-7, 2019, pp. 507–525.

[4] Thiviyan Thanapalasingam, et al., Ontology-based recommendation of editorial products, in: Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, Elena Simperl (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing, Cham, ISBN: 978-3-030-00668-6, 2018, pp. 341–358.

[5] Angelo A. Salatino, Francesco Osborne, Enrico Motta, AUGUR: Forecasting the emergence of new research topics, in: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL '18, ACM, New York, NY, USA, ISBN: 978-1-4503-5178-2, 2018, pp. 303–312, http://dx.doi.org/10.1145/3197026.3197052.

[6] Angelo A. Salatino, et al., The CSO classifier: Ontology-driven detection of research topics in scholarly articles, in: Antoine Doucet, Antoine Isaac, Koraljka Golub, Trond Aalberg, Adam Jatowt (Eds.), Digital Libraries for Open Knowledge, Springer International Publishing, Cham, ISBN: 978-3-030-30760-8, 2019, pp. 296–311.

[7] Angelo A. Salatino, et al., The computer science ontology: A large-scale taxonomy of research areas, in: Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, Elena Simperl (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing, Cham, ISBN: 978-3-030-00668-6, 2018, pp. 187–205.

[8] Serafeim Chatzopoulos, et al., Artsim: improved estimation of current impact for recent articles, in: ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, Springer, 2020, pp. 323–334, http://dx.doi.org/10.1007/978-3-030-55814-7_27.

[9] Marc Beck, et al., From automatic keyword detection to ontology-based topic modeling, in: International Workshop on Document Analysis Systems, Springer, 2020, pp. 451–465, http://dx.doi.org/10.1007/978-3-030-57058-3_32.

[10] Xiaoyu Zhang, Senthil Chandrasegaran, Kwan-Liu Ma, ConceptScope: Organizing and visualizing knowledge in documents based on domain ontology, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–13.

[11] Angelo Salatino, Francesco Osborne, Enrico Motta, CSO classifier 3.0: A scalable unsupervised method for classifying documents in terms of research topics, Int. J. Digit. Libr. (2021) http://dx.doi.org/10.1007/s00799-021-00305-y.

[12] GROBID, 2008–2021.

[13] Laurent Romary, Patrice Lopez, GROBID - Information Extraction from Scientific Publications, ERCIM News 100 (2015) Scientific Data Sharing and Re-use, URL https://hal.inria.fr/hal-01673305.

[3] CSO Portal - https://cso.kmi.open.ac.uk

[4] ACM Computing Classification System - https://dl.acm.org/ccs

[14] Francesco Osborne, Enrico Motta, Paul Mulholland, Exploring scholarly data with rexplore, in: Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, Krzysztof Janowicz (Eds.), The Semantic Web – ISWC 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 460–477.

[15] F. Löffler, et al., ScholarLensViz: A visualization framework for transparency in semantic user profiles, in: Kerry Taylor, Rafael Gonçalves, Freddy Lecue, Jun Yan (Eds.), Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas To Industrial Practice Co-Located with 19th International Semantic Web Conference (ISWC 2020), Globally Online, November 1-6, 2020 (UTC), 2020.

[16] Francesco Osborne, Giuseppe Scavo, Enrico Motta, Identifying diachronic topic-based research communities by clustering shared research trajectories, in: European Semantic Web Conference, Springer, 2014, pp. 114–129.

[17] Francesco Osborne, Giuseppe Scavo, Enrico Motta, A hybrid semantic approach to building dynamic maps of research communities, in: International Conference on Knowledge Engineering and Knowledge Management, Springer, 2014, pp. 356–372.

[18] Thanasis Vergoulis, et al., Veto: Expert set expansion in academia, in: Mark Hall, Tanja Merčun, Thomas Risse, Fabien Duchateau (Eds.), Digital Libraries for Open Knowledge, Springer International Publishing, Cham, ISBN: 978-3-030-54956-5, 2020, pp. 48–61, http://dx.doi.org/10.1007/978-3-030-54956-5_4.

[19] Marcos Vinícius Macêdo Borges, Julio Cesar dos Reis, Semantic-enhanced recommendation of video lectures, in: 2019 IEEE 19th International Conference on Advanced Learning Technologies, Vol. 2161, ICALT, IEEE, 2019, pp. 42–46, http://dx.doi.org/10.1109/ICALT.2019.00013.

[20] Anderson Rossanez, Julio Cesar dos Reis, Ricardo da Silva Torres, Representing scientific literature evolution via temporal knowledge graphs, 2020.

[21] Simone Angioni, et al., AIDA: A knowledge graph about research dynamics in academia and industry, Quant. Sci. Stud. (ISSN: 2641-3337) 2 (4) (2022) 1356–1398, http://dx.doi.org/10.1162/qss_a_00162.

[22] Danilo Dessì, et al., AI-kg: an automatically generated knowledge graph of artificial intelligence, in: International Semantic Web Conference, Springer, 2020, pp. 127–143, http://dx.doi.org/10.1007/978-3-030-62466-8_9.

[23] Danilo Dessì, et al., CS-KG: A large-scale knowledge graph of research entities and claims in computer science, in: International Semantic Web Conference, ISWC, 2022.

[24] Amparo Elizabeth Cano-Basave, Francesco Osborne, Angelo Antonio Salatino, Ontology forecasting in scientific literature: Semantic concepts prediction based on innovation-adoption priors, in: Eva Blomqvist, Paolo Ciancarini, Francesco Poggi, Fabio Vitali (Eds.), Knowledge Engineering and Knowledge Management, Springer International Publishing, Cham, ISBN: 978-3-319-49004-5, 2016, pp. 51–67.

[25] Francesco Osborne, Enrico Motta, Pragmatic ontology evolution: Reconciling user requirements and application performance, in: International Semantic Web Conference, Springer, 2018, pp. 495–512.

[26] Francesco Osborne, Andrea Mannocci, Enrico Motta, Forecasting the spreading of technologies in research communities, in: Proceedings of the Knowledge Capture Conference, 2017, pp. 1–8.

[27] Francesco Osborne, Hélène de Ribaupierre, Enrico Motta, TechMiner: Extracting technologies from academic publications, in: European Knowledge Acquisition Workshop, Springer, 2016, pp. 463–479.