Automatic Classification of Springer Nature Proceedings with Smart Topic Miner

Francesco Osborne[1], Angelo Salatino[1], Aliaksandr Birukou[2], Enrico Motta[1]

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

Abstract. The process of classifying scholarly outputs is crucial to ensure timely access to knowledge. However, this process is typically carried out manually by expert editors, leading to high costs and slow throughput. In this paper we present Smart Topic Miner (STM), a novel solution which uses semantic web technologies to classify scholarly publications on the basis of a very large automatically generated ontology of research areas. STM was developed to support the Springer Nature Computer Science editorial team in classifying proceedings in the LNCS family. It analyses in real time a set of publications provided by an editor and produces a structured set of topics and a number of Springer Nature Classification tags, which best characterise the given input. In this paper we present the architecture of the system and report on an evaluation study conducted with a team of Springer Nature editors. The results of the evaluation, which showed that STM classifies publications with a high degree of accuracy, are very encouraging and as a result we are currently discussing the required next steps to ensure large-scale deployment within the company.

Keywords: Scholarly Data, Ontology Learning, Bibliographic Data, Scholarly Ontologies, Data Mining, Conference Proceedings, Metadata.

1 Introduction

The process of classifying and annotating scholarly publications is crucial to enable scholars, students, companies and other stakeholders to easily discover and access this knowledge. To facilitate this classification process, a number of scholarly ontologies (e.g., SWRC[1], BIBO[2], BiDO[3], PROV-O[4], AKT[5], FABIO[6]) and bibliographic repositories in the Linked Data Cloud [1, 2, 3] have been proposed in the past decade, while at the same time the major publishing companies are starting to adopt richer data models [4, 5].

[1] http://ontoware.org/swrc/

[2] http://bibliontology.com

[3] http://purl.org/spar/bido

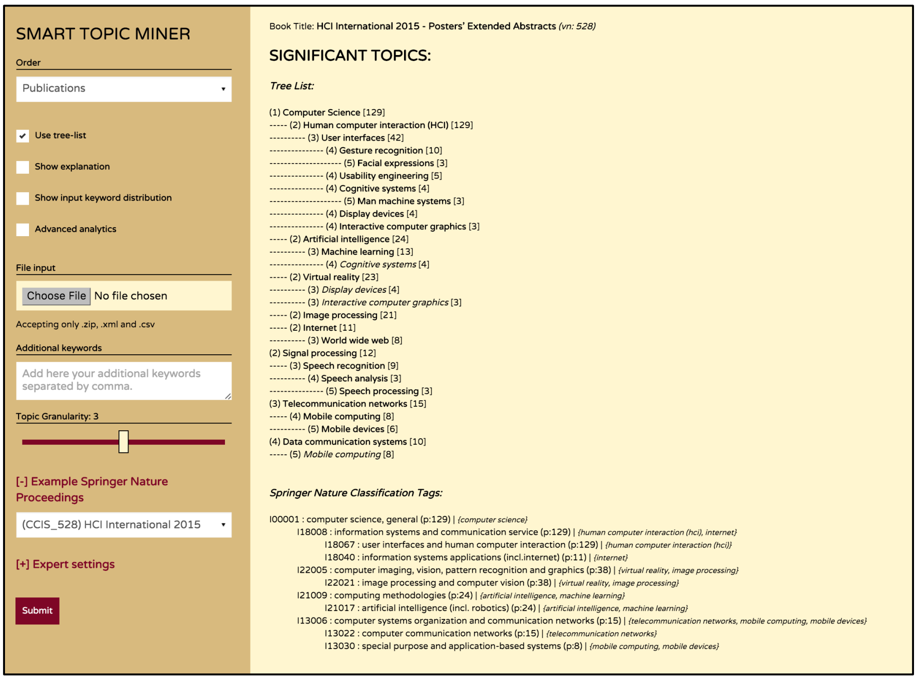
[4] https://www.w3.org/TR/prov-o/

[5] http://www.aktors.org/publications/ontology

[6] http://purl.org/spar/fabio

In this paper, we present Smart Topic Miner (STM), a novel application, developed in collaboration with Springer Nature (SN), which classifies scholarly publications according to an automatically generated ontology of research areas.

STM analyses in real-time a collection of publications and returns a description of the given corpus in terms of i) a taxonomy of research topics drawn from a large scholarly ontology and ii) a set of Springer Nature Classification tags – see Figure 1. This information is then used for a variety of tasks such as: i) classifying proceedings in digital and physical libraries; ii) enhancing semantically the metadata associated with publications and consequently improving the discoverability of the proceedings in both the Springer digital library, SpringerLink, as well as third-party sites such as Amazon.com; iii) deciding where and when to market a specific book; and iv) detecting novel and promising research areas that may deserve more attention from the publisher.



Order: Publications

☑ Use tree-list

□ Show explanation

□ Show input keyword distribution

□ Advanced analytics

File input: No file chosen

Accepting only .zip, .xml and .csv

Additional keywords: Add here your additional keywords separated by comma.

Topic Granularity: 3

[-] Example Springer Nature Proceedings: (CCIS_528) HCI International 2015

[+] Expert settings

Book Title: HCI International 2015 - Posters' Extended Abstracts *(vn: 528)*

SIGNIFICANT TOPICS:

Tree List:

```
(1) Computer Science [129]
----- (2) Human computer interaction (HCI) [129]
---------- (3) User interfaces [42]
--------------- (4) Gesture recognition [10]
-------------------- (5) Facial expressions [3]
--------------- (4) Usability engineering [5]
--------------- (4) Cognitive systems [4]
-------------------- (5) Man machine systems [3]
--------------- (4) Display devices [4]
--------------- (4) Interactive computer graphics [3]
----- (2) Artificial intelligence [24]
---------- (3) Machine learning [13]
--------------- (4) Cognitive systems [4]
----- (2) Virtual reality [23]
---------- (3) Display devices [4]
---------- (3) Interactive computer graphics [3]
----- (2) Image processing [21]
----- (2) Internet [11]
---------- (3) World wide web [8]
(2) Signal processing [12]
----- (3) Speech recognition [9]
---------- (4) Speech analysis [3]
--------------- (5) Speech processing [3]
(3) Telecommunication networks [15]
----- (4) Mobile computing [8]
---------- (5) Mobile devices [6]
(4) Data communication systems [10]
----- (5) Mobile computing [8]
```

Springer Nature Classification Tags:

```
I00001 : computer science, general (p:129) | {computer science}
    I18008 : information systems and communication service (p:129) | {human computer interaction (hci), internet}
        I18067 : user interfaces and human computer interaction (p:129) | {human computer interaction (hci)}
        I18040 : information systems applications (incl.internet) (p:11) | {internet}
    I22005 : computer imaging, vision, pattern recognition and graphics (p:38) | {virtual reality, image processing}
        I22021 : image processing and computer vision (p:38) | {virtual reality, image processing}
    I21009 : computing methodologies (p:24) | {artificial intelligence, machine learning}
        I21017 : artificial intelligence (incl. robotics) (p:24) | {artificial intelligence, machine learning}
    I13006 : computer systems organization and communication networks (p:15) | {telecommunication networks, mobile computing, mobile devices}
        I13022 : computer communication networks (p:15) | {telecommunication networks}
        I13030 : special purpose and application-based systems (p:8) | {mobile computing, mobile devices}
```

Figure 1. The STM interface.

Traditionally, when classifying proceedings, editors choose a list of related terms and categories according to their own experience, a visual exploration of titles and abstracts, and, optionally, a list of keywords given by the curators or derived by calls for papers. However, this is a complex and time-consuming process and it is easy to miss the emergence of a new topic or assume that some topics are still popular when this is no longer the case. In addition, the keywords used in the call of papers are often a reflection of what a venue aspires to be, rather than the real contents of the proceedings. For these reasons, there is a real need for more objective and scalable methods for identifying the research areas relevant to a proceedings book.

In this kind of scenario, it is critical for the editors to build confidence in the tool by being able to analyse the rationale behind the outcomes and understand why a certain research area or classification tag was chosen. Hence, we designed STM to produce intuitive explanations and to give the user full control over the granularity and nature of the topic characterization. Actually, one of the main advantages of adopting semantic web technologies is that they make it easier to generate a user-friendly explanation, as discussed in section 2.3. Of course, the final decision of which topics and tags to associate with the proceedings still rests on SN editors.

In this paper, we describe STM in terms of its knowledge bases, algorithm and user interface. We also report the outcome of an evaluation study performed with eight Springer Nature editors with expertise in a variety of different fields, as well as a coverage study on a set of 200 proceedings. Finally, we conclude by discussing the steps required for large-scale deployment of the technology within the company.

2 Smart Topic Miner

Smart Topic Miner (STM) was designed to automatically classify proceedings and more in general any collection of articles by tagging them with a number of research areas and SN classification labels. It can be used for supporting editors in classifying new books and for quickly annotating a large number of proceedings, thus creating a comprehensive knowledge base to assist the analysis of venues, journals and topic trends. In this paper, we focus on the classification/annotation task.

STM can take as input either an XML file containing metadata about a publication or a ZIP including multiple XML files. Each XML file represents a paper in a proceedings volume published in the LNCS family of book series and contains title, abstract, the keywords provided by the authors, section title and book title. Springer books are thus usually represented as collections of XML files.

STM analyses the publication metadata and returns:

- A taxonomy (or optionally a plain list) of the most significant topics annotated with the number of relevant papers/chapters, structured according to an automatically generated ontology of research areas;
- A taxonomy of Springer Nature Classification tags;
- A number of analytics to allow the editors to further analyse the content of a proceedings volume, including the list of terms and topics associated to each paper;
- Optionally, an explanation for each topic, in term of the keyword distributions that triggered the topic recognition. For instance, the Semantic Web may have been inferred as a research area for a book by recognizing terms such as “linked data”, “ontology matching”, and “semantic web services”.

Figure 1 shows the main interface and how the tool classified the Springer Nature book “HCI International 2015 - Posters’ Extended Abstracts”. Figure 2 shows the STM architecture, which consists of four main components: 1) the user interface, 2) the parser, which elaborates the input files, 3) the back-end API, and 4) the knowledge bases. Every time the user uploads a file and submits it to the system using the GUI, the parser analyses the XML files and extracts the relevant metadata. This data are sent as a JSON file to the background API via a POST query. The API analyses the data and returns the results either as a JSON or HTML file, which is in turn visualized by the interface. The API and the parser are realized in PHP and save cached data in a MariaSQL[7] database, while the front-end uses HTML5 and Javascript.

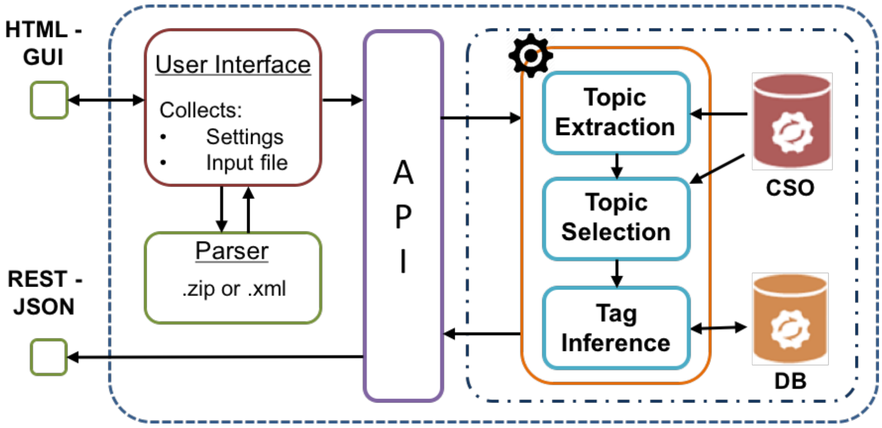


Figure 2. The STM architecture.

In the next sections we will discuss the system in detail. In section 2.1 we will elaborate on the knowledge bases, in section 2.2 we will discuss the approach to infer research areas and Springer Nature Classification tags from the metadata and finally in section 2.3 we will describe the user interface and the options available to the users.

2.1 Background data

STM uses two knowledge sources: the Klink-2 Computer Science Ontology (CSO) and the Springer Nature Classification for Computer Science (SNC).

CSO was created and subsequently updated every 6 months by applying the Klink-2 algorithm [6] on the Rexplore dataset [7], which consists of about 16 million publications, mainly in the field of Computer Science. The Klink-2 algorithm combines semantic technologies, machine learning and knowledge from external sources (e.g., DBpedia, calls for papers, web pages) to automatically generate a fully populated ontology of research areas, which uses the Klink data model[8]. This model is an extension of the BIBO ontology[9] which in turn builds on SKOS[10]. It includes three semantic relations: *relatedEquivalent*, which indicates that two topics can be treated as equivalent for the purpose of exploring research data – e.g., Ontology Matching and Ontology Mapping; *skos:broaderGeneric*, which indicates that a topic is a sub-area of another one – e.g., Linked Data is considered a sub-area of Semantic Web; and c*ontributesTo*, which indicates that the research outputs of one topic significantly contribute to research into another. For instance, research in Ontology Engineering contributes to the Semantic Web, but arguably Ontology Engineering is not a sub-area of the Semantic Web – that is, there is plenty of research in Ontology Engineering

[7] https://mariadb.org/

[8] http://technologies.kmi.open.ac.uk/rexplore/ontologies/BiboExtension.owl

[9] http://purl.org/ontology/bibo/

[10] http://www.w3.org/2004/02/skos/

outside the context of Semantic Web research. The current version of STM uses the first two relationships.

An important characteristic of the CSO ontology is that it allows for a research topic to have multiple super-areas – i.e., the taxonomic structure is a graph rather than a tree. This is a very important difference with respect to other taxonomies of research areas notably because research topics often derive from multiple areas and can be categorized under a variety of fields. For example, it can be argued that a topic such as Inductive Logic Programming should be a sub-area of both Machine Learning and Logic Programming. Hence, a representation that forces a research area to be subsumed by only one other area fails to capture adequately the network of relationships between research topics.

The current version of the CSO ontology comprises about 17k topics linked by 70k semantic relationships and includes 8 levels of granularity. The main root is the topic Computer Science, however CSO includes also a number of secondary roots, such as Geometry, Semantics, Linguistics and so on. STM uses CSO for a variety of tasks, including i) inferring a list of well-defined and human readable semantic topics from the very large distributions of terms extracted from publications, ii) supporting the set-covering algorithm, and iii) structuring the outcome as a taxonomy, in order to help the editors to understand the relationships between research areas.

CSO presents two main advantages over the classic manually crafted categorizations used in Computer Science, such as the well-known 2012 ACM Classification[11]. Firstly, it is able to recognize a very large number of terms which do not appear in these other classifications. In fact, it is about seventeen times larger than ACM in terms of number of concepts and about seventy times larger in terms of number of relationships. For this reason, it is able to characterize higher-level research areas by means of hundreds of sub-topics and related terms, which allows STM to effectively map specific terms from research publications to higher-level research areas. Secondly, the ontology can be easily updated to include novel research areas simply by adding the most recent publications to the dataset and running Klink-2 over again. Conversely, human crafted classifications cannot keep up with the evolution of the research domain and tend to age very quickly, especially in rapidly changing fields such as Computer Science. A more comprehensive discussion of the advantages of adopting an automatically generated ontology in the scholarly domain can be found in [6].

The Springer Nature Classification for Computer Science is a three level classification, containing 76 categories characterizing both research fields (e.g., *I23001 – Computer Applications*) and domains (e.g., *I23028 - Computer App. In Social and Behavioral Sciences*). It is an internal company classification, which is used in order to categorize proceedings, books, and journals. This helps to appropriately channel the contents. For instance, users browsing the Springer Nature website can retrieve all contents on Computer Science or its sub-disciplines. These codes are also used in the metadata describing the contents for third parties (libraries, bookshops).

We integrated CSO and SNC by means of 349 relationships, so that every SNC tag is now associated to a set of related topics. For example, we mapped the *systems and*

[11] http://www.acm.org/about/class/2012

data security category to topics such as Cryptography, Security Of Data, Network Security, Computer Crime, Data Privacy and so on.

The mapping was performed in three phases. First, we used Klink-2 to generate automatically a number of *relatedEquivalent* and *skos:broaderGeneric* relationships between the SN label and the topics. Then, we manually cleaned these links and created additional ones by analysing the 158 topics at the first two levels of the CSO ontology. Finally, these links were revised by a Springer editor with extensive experience in using SNC for classifying conference proceedings.

```
function STM (metadata, CSO, SNC)
    Result: topics, tags
    /* Topic extraction                                              */
    metadata ← extractKeywordsFromText(metadata);
    topics_ini ← inferTopicsFromKeywords(metadata);
    /* Topic selection, via greedy set-covering algorithm            */
    foreach level in KCS do
        topics_in_level ← getTopicsInLevel(topics_ini, level, CSO);
        while count(topics[level]) < limit for level do
            weights ← computePubWeight(metadata, topics[level]);
            /* selects the topic covering the publications with
               highest weight                                        */
            topics[level] ← selectTopic(topics_in_level, weights);
        end
    end
    /* Tag inference                                                 */
    tags ← inferTags(topics, CSO, SNC);
    return topics, tags
end
```

Algorithm 1. The STM algorithm

2.2 The STM Approach

The STM approach for generating topics and tags associated to a set of publications consists of three phases:

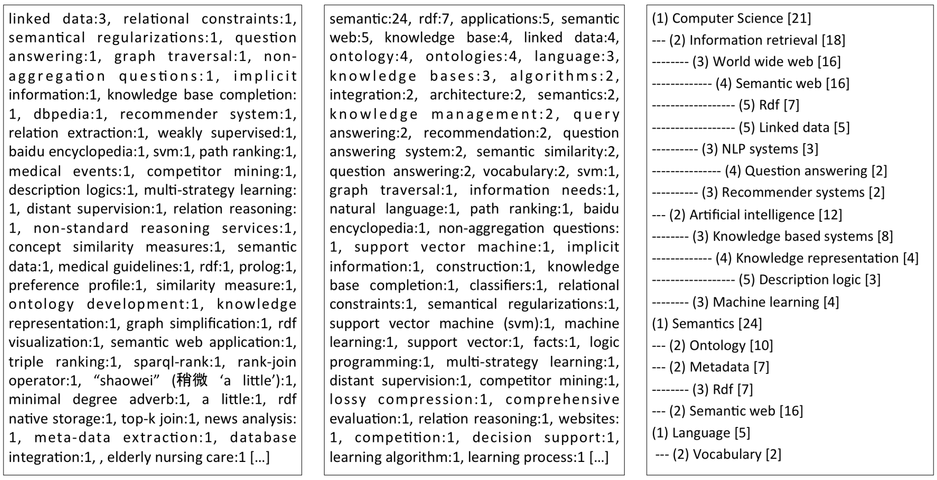
-*Topic extraction*, in which the metadata of the publications are analysed and each publication is mapped to a list of semantic topics in the CSO ontology;
-*Topic selection*, in which a greedy set-covering algorithm is used to reduce the topics to a user-friendly number, usually 10-20;
-*Tag inference*, in which the selected topics are used to infer a number of SNC tags, using the mapping between CSO ontology and SNC.

Figure 3 illustrates the steps of STM for inferring significant topics. The first panel shows the keywords provided by the authors, the second one shows the set of enriched keywords that include also the keywords extracted from titles and abstracts, the third one shows the output taxonomy.

In the next sections we will discuss the details of each step.

2.2.1 Topic Extraction

In the first step, STM extracts the title, the abstract, the list of keywords and the chapter name from the XML denoting each publication. It analyses the text and extracts frequent keyphrases and the terms that coincide with the topic labels in the CSO ontology. The publication ID is then associated to a set of keywords which include these terms, the original keywords, and optionally some keywords suggested by the editor.



linked data:3, relational constraints:1, semantical regularizations:1, question answering:1, graph traversal:1, non-aggregation questions:1, implicit information:1, knowledge base completion: 1, dbpedia:1, recommender system:1, relation extraction:1, weakly supervised:1, baidu encyclopedia:1, svm:1, path ranking:1, medical events:1, competitor mining:1, description logics:1, multi-strategy learning: 1, distant supervision:1, relation reasoning: 1, non-standard reasoning services:1, concept similarity measures:1, semantic data:1, medical guidelines:1, rdf:1, prolog:1, preference profile:1, similarity measure:1, ontology development:1, knowledge representation:1, graph simplification:1, rdf visualization:1, semantic web application:1, triple ranking:1, sparql-rank:1, rank-join operator:1, “shaowei” (稍微 ‘a little’):1, minimal degree adverb:1, a little:1, rdf native storage:1, top-k join:1, news analysis: 1, meta-data extraction:1, database integration:1, , elderly nursing care:1 […]

semantic:24, rdf:7, applications:5, semantic web:5, knowledge base:4, linked data:4, ontology:4, ontologies:4, language:3, knowledge bases:3, algorithms:2, integration:2, architecture:2, semantics:2, knowledge management:2, query answering:2, recommendation:2, question answering system:2, semantic similarity:2, question answering:2, vocabulary:2, svm:1, graph traversal:1, information needs:1, natural language:1, path ranking:1, baidu encyclopedia:1, non-aggregation questions: 1, support vector machine:1, implicit information:1, construction:1, knowledge base completion:1, classifiers:1, relational constraints:1, semantical regularizations:1, support vector machine (svm):1, machine learning:1, support vector:1, facts:1, logic programming:1, multi-strategy learning:1, distant supervision:1, competitor mining:1, lossy compression:1, comprehensive evaluation:1, relation reasoning:1, websites: 1, competition:1, decision support:1, learning algorithm:1, learning process:1 […]

```
(1) Computer Science [21]
--- (2) Information retrieval [18]
-------- (3) World wide web [16]
------------ (4) Semantic web [16]
------------------ (5) Rdf [7]
------------------ (5) Linked data [5]
---------- (3) NLP systems [3]
--------------- (4) Question answering [2]
---------- (3) Recommender systems [2]
--- (2) Artificial intelligence [12]
-------- (3) Knowledge based systems [8]
------------ (4) Knowledge representation [4]
------------------ (5) Description logic [3]
-------- (3) Machine learning [4]
(1) Semantics [24]
--- (2) Ontology [10]
--- (2) Metadata [7]
-------- (3) Rdf [7]
--- (2) Semantic web [16]
(1) Language [5]
 --- (2) Vocabulary [2]
```

Figure 3. Example of author keywords, enriched keywords and topics from CSO.

In this phase, the proceedings can also be represented as a distribution of keywords, as shown in the second panel of Figure 3. However, this representation is usually very noisy: many terms are redundant and consist of different labels for the same topics and the keyword distribution contains a long tail of terms associated with a single paper. The editors who tried STM (see section 3.1) usually considered this representation unfriendly and very time consuming to browse.

For this reason, STM uses the CSO ontology to infer a list of semantic topics from these keywords. To do so it normalizes the terms, by eliminating plural, genitive forms and common affixes and postfixes [8], and then it identifies the terms with the same label as the ontology concepts and associates to each publication tagged with them also all the relevant super areas. For example, a publication associated with the term SPARQL will be tagged with higher-level topics such as RDF, Linked Data, Semantic Web, World Wide Web, and Computer Science. Finally, it generates the topic distribution of all input publications.

The keywords for which it was not possible to find a related concept in the ontology are not included in the topic distribution, unless the user checks the “Include keywords not in the ontology” checkbox in the GUI (see section 2.3).

The drawback of this method is that an erroneous semantic connection in the ontology can sometimes lead to inferring a wrong topic and the error will then be propagated to the higher-level topics. For example, if the ontology were to state that Genetic Algorithms is a sub area of Genetics, the resulting high-level topics may include Biology, even if the proceedings do not address Natural Sciences at all. Although Klink-2 is actually able to infer semantic relationships with very high precision (> 90%, see [6]), incorrect links may still be present. However, the probability of having multiple incorrect links to the same node is quite low and the probability of multiple errors regarding all nodes in the path to the roots is extremely low. We thus addressed this problem by discarding from the topic distribution any research area which is not supported by at least *n* direct sub-topics. This prevents isolated errors in the lower levels of the ontology from easily propagating to the upper nodes. However, as *n* grows, the result set becomes smaller, since many high-level topics may be discarded. Hence, in a realistic setting it makes sense to adopt either *n*=1, equivalent to switching off this functionality, or *n*=2, potentially sacrificing recall for precision. We labelled this functionality ‘robust mode’. Editors preferred *n*=2 as default, but they also have the option of turning it off in the user interface.

The output of this process is a large set of topics associated with the relevant papers.

2.2.2 Topic Selection

The list of topics returned in the previous step is richer and more human-friendly than the term distribution, but in most cases will still suffer from prolixity, being composed of a very large number of topics. For this reason, we apply a greedy covering-set algorithm with the aim of selecting a smaller set of topics that could be easily handled by Springer Nature editors.

Since we want a comprehensive representation of the corpus given as input, which will include both high level fields and very granular research areas, we run the algorithm separately on the set of topics at each level of the ontology. The level of a topic is computed as 1 + the shortest path from the Computer Science root to the topic in question. For example, high-level fields, such as Human Computer Interaction and Artificial Intelligence, are at level 2 in the ontology, while more specific areas, such as Gesture Recognition and Speech Analysis, are at level 4. The maximum number of topics to be selected at any level depends on the granularity preferred by the user (see Section 2.3). The keywords that were not mapped to the ontology are considered in a level of their own.

The greedy covering-set algorithm assigns an initial weight equal to 1 to each paper, and at each iteration selects the topic which covered the publications with the highest total weight and reduces the weight of every covered paper (by a 0.5 factor in the prototype).

We chose this solution because the simplest version of the greedy set-covering algorithm [9], which selects at each iteration the category which covers the largest number of uncovered items, did not work well in this domain. In fact, the proceedings of a conference tend to be related to a number of topics that are often at the intersection of two or more high level topics. For example, in a Semantic Web conference the topics Artificial Intelligence and Ontology will probably cover a very similar set of publications. Hence, an algorithm that simply selects the topic that covers the larger number of uncovered publications may discard one of them. In addition, when a prominent research area has multiple super topics, the algorithm may exclude all its super topics but one. Our implementation solves this problem, by allowing topics associated to already covered publications to be chosen when they appear in enough papers to be significant.

The output of the set-covering algorithm depends on the maximum number of topics for each level and the robust mode factor, and can be further filtered by defining a minimum number of publications that a topic should cover to be taken into consideration. The user can control these settings by switching the ‘granularity level’ in the GUI between 1 and 5. Each granularity level is associated to a number of settings that will yield a more succinct or richer topic characterization. A granularity level of 1 will result in very few high level topics, while a granularity of 5 will result in a very long and comprehensive list of the topics in the proceedings.

In some cases, an unusual input, such as a book with few chapters or associated keywords, may produce very few topics when using the normal granularity settings. For this reason, STM uses by default a mechanism for adjusting the settings to the input. It checks that the output meets some minimal requisites in terms of number of topics and number of covered publications and, if this is not the case, it automatically changes the granularity settings and re-runs the topic selection process. This modality can be disabled by changing manually the granularity or deselecting the 'automatic settings' checkbox in the user interface.

The result of the summarization process can either be represented as a plain list or a taxonomy of topics. The second solution makes it easier to understand the context of each topic and why each topic was inferred; it is therefore used as default. In both cases the topics are associated to the number of papers they cover and, optionally, they are annotated with the weight computed by the set-covering algorithm.

2.2.3 Tag inference

In the final step, STM uses the mapping between the SNC and CSO to infer the SNC tags. It does so by inferring each tag that subsumes one of the selected topics according to the previously discussed mapping. For example, if the Cryptography topic was yielded by the previous step, STM will infer the tags 'I15033 - Data Encryption' (at the third level of SN Computer Science Classification), 'I15009 - Data Structures, Cryptology and Information Theory' (second level) and 'I00001 - Computer Science, general' (root). It then associates to each tag the total number of publications covered by the associated topics, so as to help the editor to assess how representative it is.

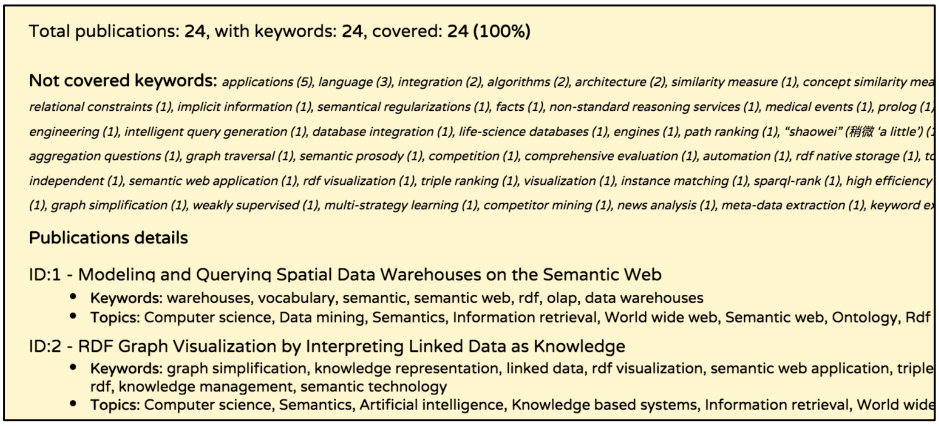
2.3 User Interface

Figure 1 shows the user interface of STM. Using the pane on the left, the user can upload the metadata, input some additional keywords and customise different settings, while the pane on the right displays the output of the process.

The interface was iteratively improved according to the feedback of experienced Springer Nature editors. In particular, the editors explained that they need a flexible tool for investigating the proceedings and for producing different kinds of annotations, rather than an automatic pipeline for annotating books. Hence, STM offers two kinds of options: those for investigating the output and those for modifying it according to the editor's needs.

The editors can control the outcome by changing the granularity of topics/tags, the metric used to order them (e.g., number of covered papers, the weight assigned by the set-covering algorithm) and the visualization style (tree list or plain list). The most used setting is the *granularity* value, which goes from 1 to 5 (default is 3) and, as discussed in section 2.2.2, allows users to choose how comprehensive should be the classification. It is mostly used to 'zoom' into the topic taxonomy, especially when the editor suspects that some significant topics may have been left out by the default visualization. In addition, the editors can choose to allow in the classification also frequent keywords that could not be mapped to the CSO ontology. This functionality allows STM to take in consideration also terms outside the Computer Science field or terms that are not strictly research areas, but may be important for assessing the content of a book, such as “commercial applications” or “empirical evaluation”. The output becomes noisier, but potentially more informative.

The main tool for exploring a proceedings book and assessing the quality of the classification is the advanced analytics functionality, which shows 1) the title of each paper/chapter, 2) the list of keywords and the percentage of publications which are not covered by the produced classification, and 3) the title and ID of each paper and its associated list of keywords and topics. Figure 4 shows a detail of its output. The list of uncovered terms is particularly useful since it reveals how complete is the representation yielded by STM. The advanced analytics functionality is often used when editors find out that a topic that they would have normally assigned to a conference does not appear in the output. In many occasions, the resulting analysis lead to the discovery that a topic, which used to be prominent, was not so popular anymore in the conference under analysis or that some topics mentioned in the call for papers were almost absent in the proceedings.



Total publications: 24, with keywords: 24, covered: 24 (100%)

Not covered keywords: *applications (5), language (3), integration (2), algorithms (2), architecture (2), similarity measure (1), concept similarity mea*
relational constraints (1), implicit information (1), semantical regularizations (1), facts (1), non-standard reasoning services (1), medical events (1), prolog (1
engineering (1), intelligent query generation (1), database integration (1), life-science databases (1), engines (1), path ranking (1), “shaowei” (稍微 ‘a little’) (
aggregation questions (1), graph traversal (1), semantic prosody (1), competition (1), comprehensive evaluation (1), automation (1), rdf native storage (1), t
independent (1), semantic web application (1), rdf visualization (1), triple ranking (1), visualization (1), instance matching (1), sparql-rank (1), high efficiency
(1), graph simplification (1), weakly supervised (1), multi-strategy learning (1), competitor mining (1), news analysis (1), meta-data extraction (1), keyword e

Publications details

ID:1 - Modeling and Querying Spatial Data Warehouses on the Semantic Web
- Keywords: warehouses, vocabulary, semantic, semantic web, rdf, olap, data warehouses
- Topics: Computer science, Data mining, Semantics, Information retrieval, World wide web, Semantic web, Ontology, Rdf

ID:2 - RDF Graph Visualization by Interpreting Linked Data as Knowledge
- Keywords: graph simplification, knowledge representation, linked data, rdf visualization, semantic web application, triple rdf, knowledge management, semantic technology
- Topics: Computer science, Semantics, Artificial intelligence, Knowledge based systems, Information retrieval, World wide

Figure 4. Fragment of the *Advanced analytics* section.

Similarly, sometimes editors find some topics in the output that seem inconsistent with their previous experience of the conference and need a way to assess them. For this reason, we included the *show explanation* checkbox, which displays near each topic the full list of terms used to infer it and how many papers they cover. For example, this functionality could show that the topic Semantic Web was inferred because the system found the terms “linked data”, “semantic similarity”, “RDF” and so on. During the tests conducted in Springer Nature, the editors used often this functionality and in most cases they discovered that the proceedings actually contained a number of terms that suggested the emergence of that topic. They were able to further confirm this intuition by examining the related papers with the *advanced analytics* functionality. In addition, the user can also inspect the full keyword distribution extracted from the text of the proceedings by checking the *show input keyword distribution* checkbox.

Finally, the users can configure more complex settings by means of the ‘expert setting’ menu, which allows them to switch on and off: 1) the order of the topics according to the ontology level, 2) the text-mining from titles, abstracts and SN metadata, 3) the robust mode, 4) the automating setting, and 5) the suggestion of SNC tags.

3 Evaluation

3.1 User Study

We conducted a qualitative study on STM to assess the quality of its output, the clarity of the explanations, the impact on the editor workflows and the usability of the user interface[12]. To this end, we organized individual sessions with eight experienced SN editors. We introduced STM and its main functionalities for about 15 minutes and then asked them to use the application for classifying a number of proceedings in their fields of expertize for about 45 minutes. Every session was recorded to further analyse their interactions with the GUI, as well as their reactions and feedbacks. After the hands-on session the editors filled a three-parts survey about their experience. The first part assessed the editor background and expertise, the second part included 8 open questions, and the third part was a standard SUS questionnaire to assess the usability of the application. A demo version of STM used in this evaluation is available at http://rexplore.kmi.open.ac.uk/STM_demo. The reader can try it by using the 'Example Springer Nature Proceedings' option, which allows testing the application by using six default SN proceedings covering a variety of distinct fields.

On average, the users had 13 years of experience as editors, with seven out of eight of them having at least 5 years. All of them stated to have extensive knowledge of the main topic classifications in their fields and seven an extensive knowledge of Springer Nature Classification. Four of them considered themselves also experts at working with digital proceedings. The expertise of the editors covered a variety of Computer Science topics, including but not limited to Theoretical Computer Science, Computer Networks, Software Engineering, HCI, AI, Bioinformatics, and Security. The open question survey consisted of five questions about the strengths and weaknesses of STM and three about the quality of the results. We will first summarize the answers to the first set of questions.

Q1. How do you find the interaction with the STM interface? Five editors considered it "good and straightforward" to use, two of them found some minor issues, and one was neutral about it. The issues included the need to re-click the 'submit' button every time the user changed a setting and the fact that the checkboxes did not have explanatory tooltips.

Q2. How effectively did STM support you in classifying books/publications? Three editors stated that the application had an extremely positive effect on their work, commenting that it was "really effective", "very good, it saved me lots of time" and "it helps a lot". Four of them assessed it positively, stating it worked well for them and the result looked correct, and one was neutral. When asked to assess the accuracy of the results the estimates varied between 75% and 90%.

Q3. What were the most useful features of STM? The most useful features included: the ability to produce taxonomies of semantic topics (7 editors), the mapping to the SNC tags (5), the ability to explore topics at different granularities (2) and the speed of the analysis (1).

Q4. What are the main weaknesses of STM? The main issues suggested by the editors were: the scope is limited to the Computer Science field (2 editors), the

[12] The data collected for the evaluation and the publication coverage study are available at http://technologies.kmi.open.ac.uk/rexplore/iswc2016/stm/.

occasional noisy results when examining books with very few chapters/keywords (2), and the wrong capitalization of some topics (1). Two editors also commented that they would like to use STM on the full text of publications, while at the moment the system can only process SN metadata.

Q5. Can you think of any additional features to be included in STM? The suggested features were: being able to produce analytics about the evolution of a venue or a journal in terms of significant topics (4 editors); allowing users to find the most significant proceedings for a certain topic (3); improving the SNC (1); and mapping the topic ontology also to the ACM classification (1).

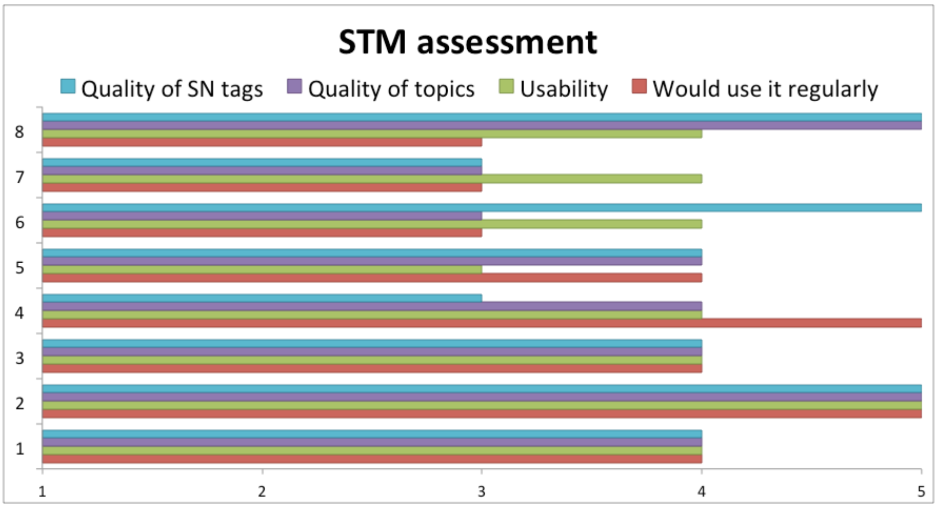


Figure 5. STM performance according to the editors (labelled 1-8).

We mapped the three remaining questions on a 1 to 5 scale, where 1 is the most negative assessment and 5 the most positive. Figure 5 shows the quality of SNC tags and topics, the usability (according to the SUS statement "I thought the system was easy to use"), and the willingness of the users to use the application regularly for their work. These features obtained a similar average score: quality of SNC tags 4.0±0.8, quality of topics 3.9±0.8, usability 4.0±0.5, and willingness to use it regularly 4.0±0.8. Interestingly, the quality of the topics was considered generally higher by the three editors working exclusively with proceedings (4.7±0.6).

The SUS questionnaire confirmed the good opinion of the editors, scoring a 77/100 (the average system scores a 68), which places STM in the 80% percentile rank in term of usability.

3.2 Assessing Publication Coverage

Editors need a topic summarization that is succinct but covers most of the publications. We thus performed a study about how the semantic topics produced by STM compare with keywords in terms of coverage.

To this end, we selected a dataset of 200 SN proceedings and generated for each of them three sets of categories: 1) the keywords defined by the authors, originally associated with each publication, 2) the enriched set of keywords, which also included additional terms extracted from abstracts, titles, and SN metadata (as discussed in section 2.2.1), and 3) the semantic topics produced by STM. Then, for each proceedings book, we computed the average number of papers covered by each member of the first n-th category, using the three sets. We used the average coverage rather than the total coverage, since the latter grows monotonically with the number of descriptors and thus the top level categories (e.g., Computer Science), that often cover most of the papers, would obscure the more fine-grained ones.

	1	**2**	**3**	**4**	**5**	**10**	**20**	**30**
Author keywords	2.83	2.42	2.18	2.04	1.92	1.57	1.33	1.25
Enrich. keywords	8.27	6.95	6.11	5.56	5.14	3.95	2.92	2.44
Topics	25.26	21.08	18.83	17.19	15.93	12.03	8.62	6.93

Table 1. Average number of papers covered by the first *n* descriptors.

Table 1 shows the average result across the proceedings. The topics produced by STM performed significantly better than the enriched keywords (the Wilcoxon test for two correlated distributions yielded $p<0.0007$), which in turn outperformed the author keywords ($p<0.0007$). Hence, we can conclude that while extracting keywords from text allows for a more representative set of categories, adding semantic to this representation produces a much more complete set of categories.

4 Plans for large scale deployment

While the tool was very well received in Springer Nature, making it part of the daily workflow of the editors requires additional steps. Before outlining these, let us take a closer look at the context of the Computer Science proceedings in Springer Nature. Every year Springer Nature publishes about 1,200 proceedings volumes. Almost 800 of these are published in Computer Science, more specifically in the Lecture Notes in Computer Science (LNCS) series family. This includes LNCS itself, its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), as well as more recently launched series, such as Lecture Notes in Business Information Processing (LNBIP), and Communications in Computer and Information Science (CCIS). Last but not least, there are also two series in cooperation with the IFIP and ICST/EAI societies (IFIP-AICT and LNICST, respectively).

In order to deploy STM at such a large scale (classifying 800 proceedings/year) we have to connect STM with existing production systems. In an ideal case, STM could receive inputs already from the conference submission system used by the conference. This could happen during the preparation of the material for publication by the conference chairs. In practice, however, the diversity of the submission systems and the lack of commonly accepted standards used beyond the Semantic Web community makes it difficult to expect each of the existing conference management systems to adopt a standard for exporting the data about abstracts, titles, and keywords required by STM. Therefore, we are going to explore the integration of STM with the Springer Nature's own submission system, OCS (Online Conference Service[13]).

For the proceedings not using OCS, the data required for STM will be provided during the production process, after the metadata about individual papers have been

[13] https://ocs.springer.com/ocs/

finalized and before the proceedings are published. Depending on how STM performs and the overall metadata strategy of Springer Nature, STM might be used for annotating already published contents (roughly 10,000 proceedings volumes). In combination with the data already available at the Springer LOD portal,[14] this would allow editors to analyse the evolution of conference topics.

We are also looking into how STM could be used to improve the existing Springer Nature Classification. One possible way of approaching this problem would be to set up periodic updates of the SNC based on the most recently published material. During such updates new categories could be added, corresponding to the emerging topics, while the categories corresponding to disappearing topics could be deprecated.

Finally, we also plan to expand the scope of the research area ontology to fields other than Computer Science, to support the classification of books and proceedings in other domains as well.

5 Related Work

STM identifies research areas from a corpus of metadata by using an automatically generated ontology of topics. In this regard, it can be considered a classic name-entity linking approach. In particular, many historical approaches focus on linking entities to general knowledge bases, such as Wikipedia or DBpedia. For example, Mihalcea and Csomai [10] and Bunescu and Pasca [11], introduced some of the first approaches for mapping text to Wikipedia pages. Since then, we saw the creation of a number of systems for name-entity linking which exploited DBpedia or Wikipedia, including DBPedia Spotlight [12], Microsoft Entity Linking[15], BabelFly [13], Illinois Wikifier [14], KORE [15], AGDISTIS [8] and many others. DBpedia Spotlight is also used by the Klink-2 [6], the algorithm which generated the CSO ontology, for linking keywords to DBpedia entities and informing the identification of semantic relationships between research topics. However, using directly DBpedia as source for research areas presents some issues, since the research fields taxonomy in DBpedia is quite coarse-grained: it does not contain some of the most recent or specific topics and lacks a number of links between them. Another alternative is the Machine Aided Indexer[16], a rule-based document indexer that can map the full text of a document to a taxonomy of topics. However, this method requires the manual definition of a number of rules for the mapping.

Similarly to STM, a number of methods for topic detection extract topics from a corpus of documents. The best-known technique is the Latent Dirichlet Allocation (LDA) [16], which considers each document as a distribution of topics and each topic as a distribution over words. This approach is applicable to any kind of documents and has been influential in the topic detection community in the last decade. For this reason, we saw the emergence of a number of solutions tailored to the scholarly domain. For example, He at al [17] introduced an approach which makes use of the citation graph while the Author-Conference-Topic (ACT) model used by AMiner [18] exploits also information about authors, conferences and journals. However, LDA and

[14] http://lod.springer.com

[15] https://www.microsoft.com/cognitive-services/en-us/entity-linking-intelligence-service

[16] http://www.dataharmony.com/services-view/mai/

similar methods are a good solution mainly in scenarios where a very large numbers of documents need to be classified, there is no good domain knowledge, fuzzy classification is acceptable and it is not important for users to understand the rationale of a classification or customise the output. None of these tenets apply to our case.

A number of digital libraries (e.g., ACM, Springer Nature, Scopus[17]) and academic search engines (e.g., Microsoft Academic Search[18]) rely on taxonomies of topics for supporting the classification of research publications. STM uses a similar solution by adopting the CSO ontology. Indeed, ontologies of research topics have proved to be very useful to enrich semantically a number of analytics models [7], as well as supporting trend detection [19] and community detection [20, 21].

6 Conclusions

In this paper, we have presented Smart Topic Miner, a novel Semantic Web application designed to assist Springer Nature editors in classifying conference proceedings. The evaluation, performed with a number of experienced Springer Nature editors, showed that STM produces accurate and useful results. In particular, the semantic model on which STM builds was considered very helpful since it allows the editors to obtain a more concise representation, which can be easily analysed. A key lesson learned during the STM development regards the critical value of producing human-friendly explanations and the value of an explicit semantic representation for supporting this task.

We are planning to integrate the STM tool into Springer Nature workflows, in particular those used for publishing Computer Science proceedings. The use of such controlled topic vocabulary will improve discoverability and navigation of the contents of Springer Nature proceedings, as well as enable new applications. In addition, STM could be extended to indicate the emergence of new topics, as well as the fading of some traditional ones. Finally, we also plan to explore the possibility of using STM for directly supporting authors in defining the set of topics which best describe their paper.

Acknowledgements

We would like to thank the Springer Nature editors for assisting us in the evaluation of STM.

References

1. Möller, K., Heath, T., Handschuh, S., Domingue, J.: Recipes for semantic web dog food—The ESWC and ISWC metadata projects. Springer. (2007)
2. Latif, A., Afzal, M.T., Helic, D., Tochtermann, K., Maurer, H.A.: Discovery and Construction of Authors' Profile from Linked Data (A case study for Open Digital Journal). LDOW 2010. (2010)

[17] https://www.elsevier.com/solutions/scopus

[18] http://academic.research.microsoft.com/

3. Glaser, H., Millard, I.: Knowledge-enabled research support: RKBExplorer.com. Proceedings of Web Science. (2009)
4. Bryl, V., Birukou, A., Eckert, K., Kessler, M.: What's in the proceedings? Combining publisher's and researcher's perspectives. In: SePublica 2014, (2014)
5. Hammond, T., Pasin, M.: The nature. com ontologies portal. In: 5th Workshop on Linked Science 2015, colocated with International Semantic Web Conference 2015, Bethlehem, USA. (2015).
6. Osborne, F., Motta, E.: Klink-2: integrating multiple web sources to generate semantic topic networks. The Semantic Web-ISWC 2015, pp. 408-424. Springer (2015)
7. Osborne, F., Motta, E., Mulholland, P.: Exploring scholarly data with rexplore. The Semantic Web–ISWC 2013, pp. 460-477. Springer (2013)
8. Usbeck, R., Ngomo, A.C.N., Röder, M., Gerber, D., Coelho, S.A., Auer, S., Both, A. AGDISTIS-graph-based disambiguation of named entities using linked data. In The Semantic Web–ISWC 2014 (pp. 457-471). Springer International Publishing. (2014)
9. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of operations research 4, 233-235 (1979)
10. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pp. 233-242. ACM (2007)
11. Bunescu, R.C., Pasca, M.: Using Encyclopedic Knowledge for Named entity Disambiguation. EACL, vol. 6, pp. 9-16 (2006)
12. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. Proceedings of the 7th International Conference on Semantic Systems, pp. 1-8. ACM (2011)
13. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics 2, 231-244 (2014)
14. Cheng, X., Roth, D.: Relational inference for wikification. Urbana 51, 61801 (2013).
15. Hoffart, J., Seufert, S., Nguyen, D.B., Theobald, M., Weikum, G.: KORE: keyphrase overlap relatedness for entity disambiguation. Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 545-554. ACM (2012)
16. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research 3, 993-1022 (2003)
17. He, Q., Chen, B., Pei, J., Qiu, B., Mitra, P., Giles, L.: Detecting topic evolution in scientific literature: how can citations help? Proceedings of the 18th ACM conference on Information and knowledge management, pp. 957-966. ACM (2009).
18. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 990-998. ACM (2008)
19. Decker, S.L., Aleman-Meza, B., Cameron, D., Arpinar, I.B.: Detection of bursty and emerging trends towards identification of researchers at the early stage of trends. Aug-2007. [Online]. Available: http://athenaeum. libs. uga. edu/handle/10724/9958 (2007)
20. Erétéo, G., Gandon, F., Buffa, M.: Semtagp: semantic community detection in folksonomies. Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. (2011)
21. Osborne, F., Scavo, G. and Motta, E. Identifying diachronic topic-based research communities by clustering shared research trajectories. In European Semantic Web Conference (pp. 114-129). Springer International Publishing. (2014)