Smart Book Recommender: A Semantic Recommendation Engine for Editorial Products

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

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

Abstract. Academic publishers, such as Springer Nature, need to constantly make informed decisions about how and where to market their editorial products. In the field of Computer Science (CS), it is particularly critical to assess which books will be of interest to the attendees of a conference. Typically, these items are manually chosen by publishing editors, on the basis of their personal experience. To make this process both faster and more robust we have developed the Smart Book Recommender (SBR), a semantic application designed to support the Springer Nature editorial team in promoting their publications at CS venues. SBR takes as input the proceedings of a conference and suggests books, journals, and other conference proceedings which are likely to be relevant to the attendees of the conference in question. It does so by taking advantage of a semantic representation of topics, which builds on a very large ontology of Computer Science topics; characterizing Springer Nature books as distributions of semantic topics; and approaching the problem as one of semantic matching between such distributions of semantic topics.

Keywords: Scholarly Data, Recommendation Systems, Ontology, Bibliographic Data, Scholarly Ontologies.

1 Introduction

Academic publishers need to constantly make timely and data-driven decisions to ensure that they are showcasing their editorial products to their target market. In the field of Computer Science, it is particularly critical to assess which books, journal, or proceedings will be of interest for the attendees of a conference. Typically, these items are manually chosen by publishing editors, on the basis of their personal experience. As the number of publications grows, there is an increasing need for automated and data-driven methods that can support this complex and time-consuming task by analysing large-scale data about editorial products.

In what follows we present the Smart Book Recommender (SBR), a web application developed in collaboration with Springer Nature, which recommends books, journals and conference proceedings that are likely to be relevant to the attendees of a given conference. This work stems from the ongoing collaboration between Springer Nature and the Knowledge Media Institute (KMi) of the Open University, which has produced a number of other innovative solutions, including Smart Topic Miner (STM) [1], a semantic framework for classifying academic documents, and its API, the Smart Topic API.

Since January 2017, STM is being routinely used by the SN Computer Science editorial team, halving the time for classifying conference proceeding. Similarly, SBR is in line to be adopted to support SN editors in selecting the best set of books to market to the participants of a conference. A demo of the SBR prototype is available at http://rexplore.kmi.open.ac.uk/SBR-demo.

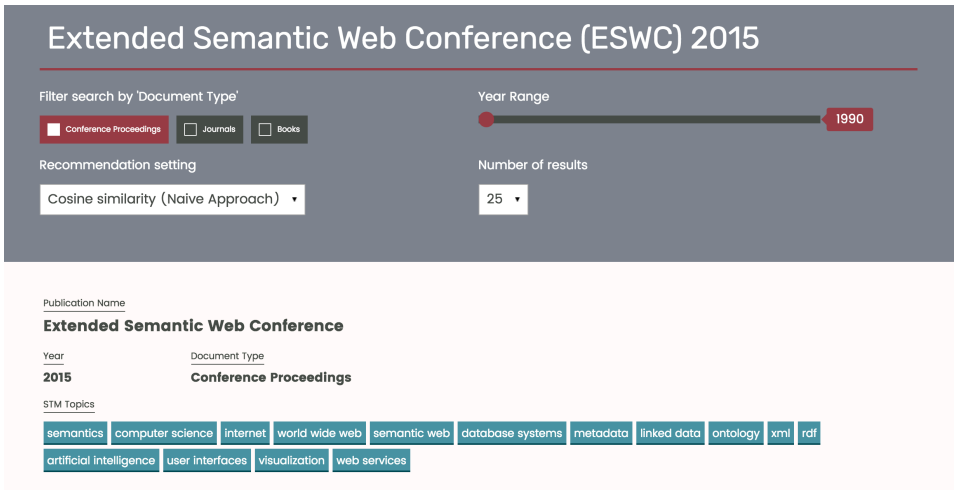

Figure 1. The SBR interface.

2 Smart Book Recommender

SBR takes as input the title of a SN book, usually the proceedings of a conference, and returns a list of books, journals, and proceedings which address topics that are likely to be relevant to the participants of the conference in question. To do so, it represents SN books in Computer Science as distributions of semantically-characterized topics, which are drawn from a large-scale ontology of Computer Science, and then computes their pairwise similarity. SBR offers a simple web interface, shown in Figure 1, to allow editors to filter the results and share their feedback. It is thus complementary, but very different in scope, to Reccomended[1], the SN recommender system, which suggest books to users on the basis of their last 100 papers read on online platforms.

SBR relies on the following background knowledge: a large database of SN book metadata and the Computer Science Ontology (CSO). The database of metadata contains titles, abstracts, keywords and other information describing the chapters of about 27K books and 270 journals in the field of Computer Science. In the case of conference proceedings and journals, each chapter is usually a research paper.

CSO is a large scale and granular ontology of research topics that has been created automatically by running the Klink-2 algorithm [2] on the Rexplore dataset [3], which consists of about 16 million publications in the field of Computer Science. It is an extension[2] of the BIBO ontology[3] which in turn builds on SKOS. The current version of CSO includes about 17K topics, which are linked by 70K semantic relationships.

[1] http://recommended.springernature.com/recommended/
[2] http://kmi.open.ac.uk/technologies/rexplore/ontologies/BiboExtension.owl

2.1 Architecture

Figure 2 shows the architecture of SBR. The computation of the pairwise similarity between SN books is performed offline. The Recommendation Engine iterates on journals, conference proceedings, and other books, and retrieves for each of them the relevant set of chapters/papers. It then sends this metadata to the Smart Topic API, which extracts frequent terms from abstracts, titles and keywords, maps them to the CSO ontology concepts, and prunes the resulting topics with a set-covering algorithm as detailed in [1]. The mappings from terms to concepts take into account both synonyms and sub-areas of a topic – e.g., all documents associated to terms such as "semantic technologies", "linked data", "RDF", "OWL" will be also tagged with the topic "Semantic Web". The result of this process is a distribution of topics from CSO, in which each topic is associated with the number of chapters/papers addressing it. The recommendation engine uses this representation for pre-computing and storing in the database the similarity scores. Presently, the prototype uses the cosine similarity of the topic vectors, but we are testing other similarity metrics.

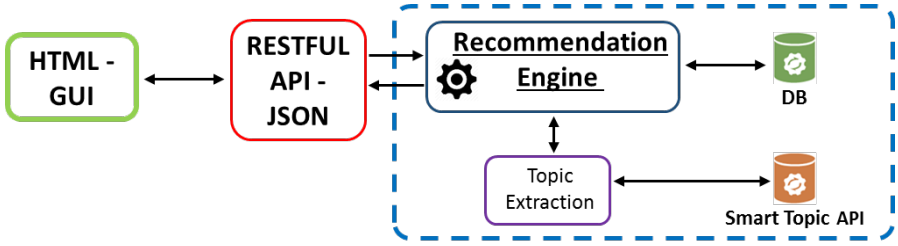

Figure 2. The architecture of SBR.

Since computing the cosine similarity of all the books in the dataset is computationally-heavy, we consider only promising pairs which obtain a jaccard similarity of at least 0.125. A data analysis revealed that this heuristic halves the number of candidate pairs while still producing very good results.

When the user submits an input book to the SBR web interface, the relevant ID and user settings are sent as JSON to the background API via a GET query. The API queries the database for selecting the most similar books and returns their descriptions. Both the API and the recommendation engine are realized in Python.

2.2 The Web Interface

The user can select specific conference proceedings by typing their name in an autocomplete field. The recommendation results are displayed in order of descending similarity scores and they can be filtered according to document types (journals, books, conference proceedings) and year range.

Figure 3 shows an example of proceedings suggested by SBR. For simplicity, proceedings of different editions of the same conferences are grouped together. Each proceedings volume is described according to its title, year and the top fifteen topics. SBR also highlights in blue the topics which are among the top fifty topics of the input book. The user has the option to provide feedback about each item using emoticons buttons. When the user clicks either button, the feedback is sent to the API and recorded in the

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

database. These data will be used to determine the quality of different similarity metrics and further enhance the recommendation process.

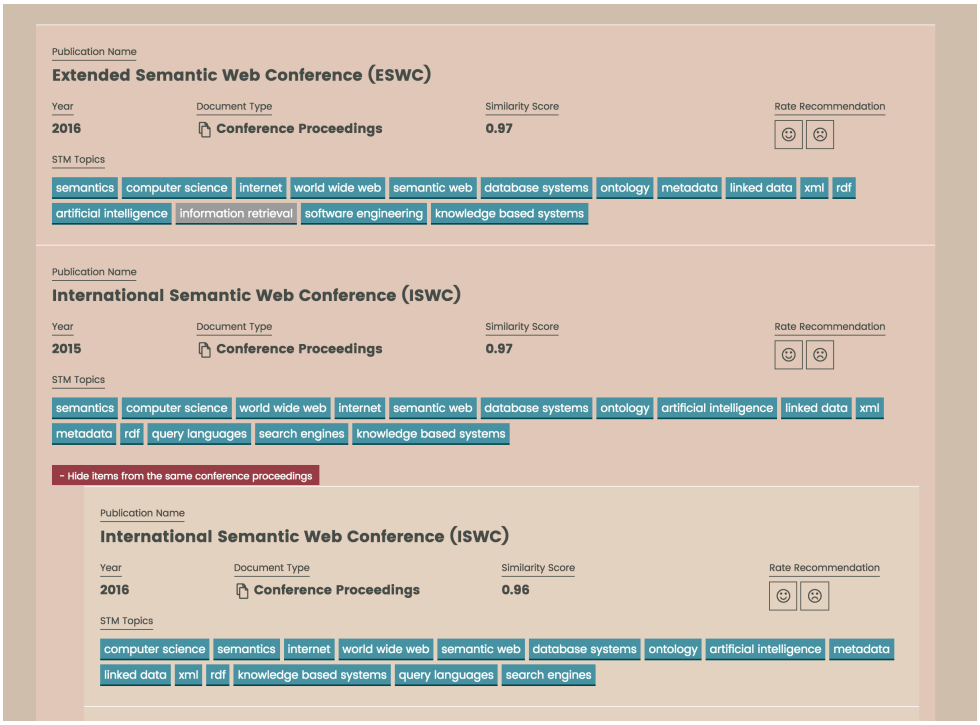

Figure 3. Example of suggested conference proceedings.

3 Conclusions

In this demo paper, we presented the prototype of SBR, a novel system for identifying related editorial products and facilitating the marketing process at SN.

As next steps, we intend to improve the recommendation process using other features (e.g., sales figures) and to conduct a formal evaluation with a group of SN editors. We are also planning to design a more advanced user interface for comparing the topics of different books and to implement a new version of the system for assisting researchers in identifying books and conferences which are relevant to their work.

4 References

1. Osborne, F., Salatino, A., Birukou, A., Motta, E.: Automatic Classification of Springer Nature Proceedings with Smart Topic Miner. In International Semantic Web Conference 2016 (pp. 383-399). Springer. (2016)
2. Osborne, F., Motta, E.: Klink-2: integrating multiple web sources to generate semantic topic networks. In International Semantic Web Conference 2015 (pp. 408-424). Springer. (2015)
3. Osborne, F., Motta, E. and Mulholland, P.: Exploring scholarly data with Rexplore. In International Semantic Web Conference 2013 (pp. 460-477). Springer. (2013)