Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 38 (2014) 84 – 91

Procedia Computer Science

10th Italian Research Conference on Digital Libraries, IRCDL 2014

# A Personalized Concept-Driven Recommender System for Scientific Libraries

D. De Nart, C. Tasso*

*Artificial Intelligence Lab, Department of Mathematics and Computer Science, University of Udine, Italy*

**Abstract**

Recommender Systems can greatly enhance the exploitation of large digital libraries; however, in order to achieve good accuracy with collaborative recommenders some domain assumptions must be met, such as having a large number of users sharing similar interests over time. Such assumptions may not hold in digital libraries, where users are structured in relatively small groups of experts whose interests may change in unpredictable ways: this is the case of scientific and technical documents archives. Moreover, when recommending documents, users often expect insights on the recommended content as well as a detailed explanation of why the system has selected it, which cannot be provided by collaborative techniques. In this paper we consider the domain of scientific publications repositories and propose a content-based recommender based upon a graph representation of concepts built up by linked keyphrases. This recommender is coupled with a keyphrase extraction system able to generate meaningful metadata for the documents, which are the basis for providing helpful and explainable recommendations.



Peer-review under responsibility of the Scientific Committee of IRCDL 2014

*Keywords:* Recommender Systems; Digital Libraries; scientific publication; Computer Science; cold start problem

## 1. Introduction

Recommender systems are extremely valuable tools when providing access to catalogs of items which are too large to be browsed manually in reasonable time, such as the ones provided by current digital libraries. Scientific digital libraries host huge catalogs of publications browsed every day by thousands of researchers who seek relevant results. Due to the large availability of data, this activity is extremely time consuming and therefore recommender systems can provide a valuable support. Most of the today recommender systems are based upon Collaborative Filtering (CF) techniques, i.e. they filter resources according to user ratings [1]. However two issues must be considered before applying CF to a particular domain:

- *User base*: collaborative recommenders assume that the number of users who rate products is much larger than the number of items and that users who expressed similar interests will maintain similar interests over time. These assumptions hold in the domain of e-commerce, where there is a specialized catalog, focused on a single domain, for instance books. In this scenario users are clients who are very

*Corresponding author

*Email address:* c.tasso@uniud.it (C. Tasso)



Peer-review under responsibility of the Scientific Committee of IRCDL 2014

doi:10.1016/j.procs.2014.10.015

likely to interact with the system several times, often consuming similar items, and therefore ratings are quite easy to obtain.

- *Cold start problem and long tail*: new items or old items that received very few ratings are unlikely to be recommended in a collaborative system. In the domain of e-commerce these problems are not a critical issue since new products are already pushed by advertising and most of the profits are driven by a few blockbuster items, with the so-called long tail of the catalog generating a relatively small fraction of the income.

However, domains exist in which the above assumptions do not hold and the above issues become critical. One of them is the domain of scientific publications, where users are relatively few with respect to the available documents, information needs and interests easily change in an unpredictable way over time due to evolving professional needs, there is no advertising pushing new items, and the long tail of infrequently read articles may contain the so-called *sleeping beauties*, that are documents containing extremely relevant results, but that remain unknown to most researchers for a very long time[2]. On the other hand, content-based recommenders, typically build attribute vector representations of contents and user preferences and generate recommendations according to the degree of similarity between user interests and items. This approach does not require particular assumptions over the size and the activity of the user base, nor penalizes items that have not been rated or consumed yet by many users as long as satisfactory metadata are available. Moreover, the presence of such metadata allows detailed explanations. These advantages over Collaborative Filtering techniques make this approach particularly attractive to the purpose of providing recommendation in the domain of scientific publications. State-of-the-art content-based recommenders, however, need a reliable source of metadata and expect every user to build a detailed profile specifying a set of desired item characteristics, which may be a time-consuming activity, therefore pushing the problem of cold start from items to users. In this work we propose a novel content-based recommender technique based on a network, rather than vector, user model built upon sets of concepts automatically extracted from documents. By using concepts as features, we have developed a concept-based recommender that suggests the papers related to the concepts of interest for the active user. More specifically, concepts are identified as keyphrases automatically extracted from scientific papers. A keyphrase (KP) is a short phrase (typically constituted by up to three/four words) that indicates one of the main ideas/concepts included in a document. A keyphrase list is a short list of keyphrases that reflects the content of a single document, capturing the main topics discussed and providing a brief summary of its content. The proposed recommender system builds a user profile mainly by means of relevance feedback, i.e. by exploiting the keyphrase lists extracted from the papers that are considered and explicitly stated as relevant by the active user. Then, in order to compute the relevance of a new article, the user profile is matched against the keyphrase list extracted from that article. The automatic keyphrase extraction avoids a manual classification of papers and it still identifies a significant set of concepts as we showed in[3]. Our objective is to provide accurate recommendations with little cold start issues exploiting the same cognitively plausible model for both document contents and user interests, allowing the system to offer intuitive and expressive ways to build a detailed user profile and to later provide satisfactory explanations of recommendations. In order to support our claim, a test system, providing access to a large library of scientific publications was built and experimentally evaluated. The paper is organized as follows: Section 2 reviews related work, Section 3 presents a brief architectural overview of the system, Section 4 describes the proposed recommendation method, Section 5 presents the evaluation performed so far, and Section 6 concludes the paper.

## 2. Related Work

Several works in the literature deal with the problem of supporting access and navigation in large scientific publications libraries, mostly from an Information Retrieval perspective, such as in[4], where CiteSeer is presented. However there are several authors who have taken into account more personalization-based approaches to the problem, leading to the development of recommender systems rather than search engines. For example, in CiteUlike, two collaborative filtering mechanisms are exploited: (i) an item-based CF recommender system where the tags provided by the users are exploited for identifying the resources similar to

those the active user previously liked and (ii) a user-based recommender system where the resources liked by the users who share papers with the active user are recommended[5]. Other works take into account the textual contents of papers in order to provide recommendations. Some of them[6] take into account specific sections of the papers such as the bibliography which can be used to build, navigate, and, moreover, mine the citation graph (i.e. the directed graph in which each vertex represents a publication and each edge represents a citation from one publication to another) in order to generate recommendations. Others, consider the relations involving users, publications, tags, and other metadata: in[7] this information is used to produce a graph according to which personalized suggestions are computed by means of the FolkRank algorithm[8]. On the other hand, our work aims at extracting from the documents the main ideas and concepts in order to describe from a more semantic perspective the interests of users who consumed and liked that paper. Similarly, the feedback of the users of social systems, such as CiteUlike and BibSonomy, has been also used for identifying the concepts of interests for the user. The authors of[9], for example, extract the tags provided by the users of CiteUlike for generating a dictionary which can be used for identifying relevant concepts present in the abstract of scientific publications. The precision of these approaches depends on the active participation of the users whereas the content-based recommender system described in this paper is solely based on the automatic extraction of the main concepts from a scientific resource. The textual content of scientific papers is also analyzed in a concept-based recommender system proposed in[10], where authors and papers are modeled by trees of concepts: using the ACM Computing Classification System (CCS), the authors trained a vector space classifier in order to associate concepts of the CCS classifications to documents. The hierarchical organization of the CCS allows the system to represent user interests and documents by trees of concepts. A user profile and a paper representation are then compared by a tree edit-distance which computes a similarity measure among trees. Our approach, on the other hand, does not need a training phase and it also does not depend on specific ontologies for identifying relevant concepts (represented as keyphrases constituted by n-grams) in the papers. Finally, In[11], the authors propose a content-based filtering system based on a simple, unsupervised, keyphrase extraction technique to identify relevant concepts and entities by considering their frequency in the document. Extracted keyphrases are then clustered according to their Google distance[12] and then a vector of related terms is used as document model.

## 3. System Overview

In order to support our claims and to test our approach we have developed a specific recommender system for scientific publications, called *Recommender and Explanation System* (*RES*)[13][1], described in the following. The main goal of RES is providing personalized access to documents retrieved from CiteSeerX. The overall architecture of the system, showed in Figure 1, includes a database called *Scientific Paper Collection (SPC )*, a repository for user profiles, and the following three main modules:

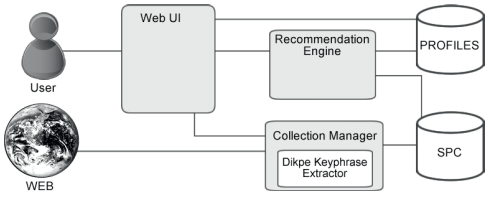


Figure 1. System Architecture Overview.

- A *Web User Interface* devoted to (i) let the user create and manage profiles, (ii) specify one or more documents of interest, to be used as positive relevance feedback, either by browsing a list of articles within the SPC or uploading new ones, (iii) query CiteSeerX, and (iv) request recommendations. These are presented as a ranked list of documents where the top items are those that better match the

[1] The novelty of the presented paper with respect to[13] is an extended evaluation activity

user profile. For each document a tag cloud of concepts contained in the document is shown. This information, shown in Figure 2, serves two goals: it briefly explains why a document was recommended by highlighting its main concepts and, secondly, offers the user a way to provide relevance feedback. Users can explicitly adjust the weight of concepts already included in their model, deleting them or adding new ones.

- A *Collection Manager Module*, devoted to: (i) execute queries on CiteSeer and crawl results, (ii) preprocess articles by extracting KPs from full text, and (iii) store their representations, as a list of KPs, into the SPC. This module has been developed using the Dikpe KP extraction algorithm [2] described in[14], which has proven to perform significantly better than other known systems. The Dikpe KP extractor provides, as output, a list of KPs extracted from the document where each KP has a weight called *Keyphraseness* that summarizes the several linguistic, statistical and social indicators exploited in the extraction process. The higher the Keyphraseness, the more relevant is the KP in the document.
- A *Recommendation Engine Module* devoted to: build and maintain individual user profiles; retrieve query results from the SPC, and then recommend the most promising papers.

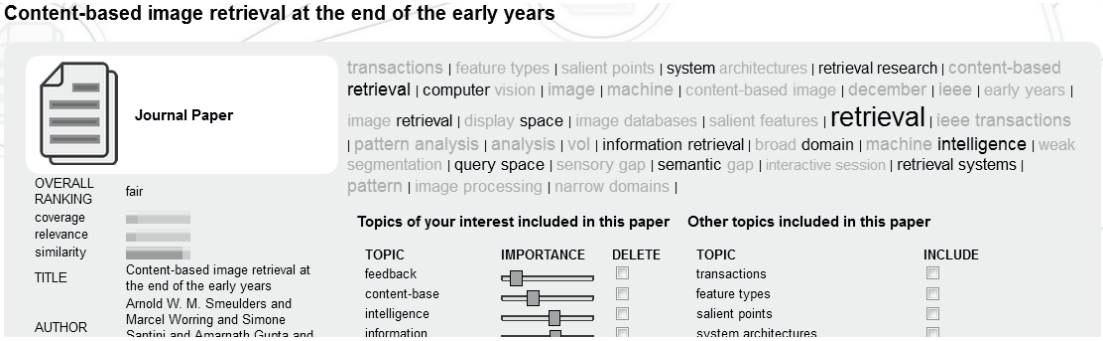


Figure 2. Recommendation screenshot.

The SPC is an important part of the system since Keyphrase Extraction is, computationally, a demanding task and a set of hundreds of query results cannot be processed in an interactive way. In order to address this issue, we decided to let RES process retrieved documents only once, in an asynchronous way, and cache their representation for later use. On the other hand, when the document KPs are known, the recommendation algorithm proposed is very efficient and it is able to rank large sets of documents in a short time.

## 4. Proposed Method

In the RES system, both user profile and document content are represented by a network structure called *Context Graph* (CG). For each document stored in the SPC, a CG is built by processing a weighted list of KPs. In the current system such list is automatically extracted from full texts and the used weight is keyphraseness; however, to the extents of the recommendation algorithm, KPs may come from metadata or be manually generated as well and any weight metric could be used. User profiles are represented by CGs built from KPs belonging to SPC documents marked by the user as interesting and, possibly, enriched with other KPs gathered via relevance feedback, for example by providing a fragment of text or a specific paper not previously included in the SPC, or a specific list of KPs or keywords. CGs are built by taking into account each single term belonging to each KP; each term is stemmed and then represented as a node of the graph; if two terms belong to the same KP, their corresponding nodes are connected by an arc. Both nodes and arcs are assigned a weight which is computed as the sum of the weights of the KPs that generated them and then normalizing such sum. In Figure 3 is shown a small CG formed by five KPs.

[2]The keywords of this article were generated from its full text with the Dikpe KP extraction algorithm.

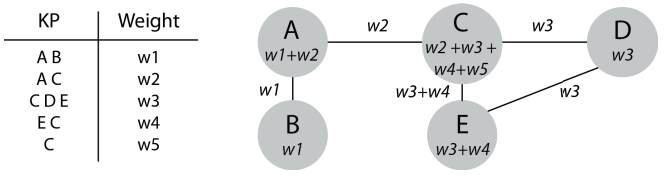


Figure 3. A simple Context Graph.

As new KPs are added to the CG, either by direct article insertion or relevance feedback, both provided by the user, related concepts tend to link together, creating, in such a way, extensive networks of terms. Consider for example the profile CGs shown in Figure 4, the one on the left has been built from four articles dealing with 'Content-based Recommender Systems' and 'Information Extraction'; on the right-hand side two unrelated articles (the first dealing with Machine Learning, the second with Mechanical Engineering) are fed into a profile showing how unrelated concepts form different, non-connected groups. If a user expresses multiple domains of interest in his profile, they will form different groups in the corresponding CG. CGs

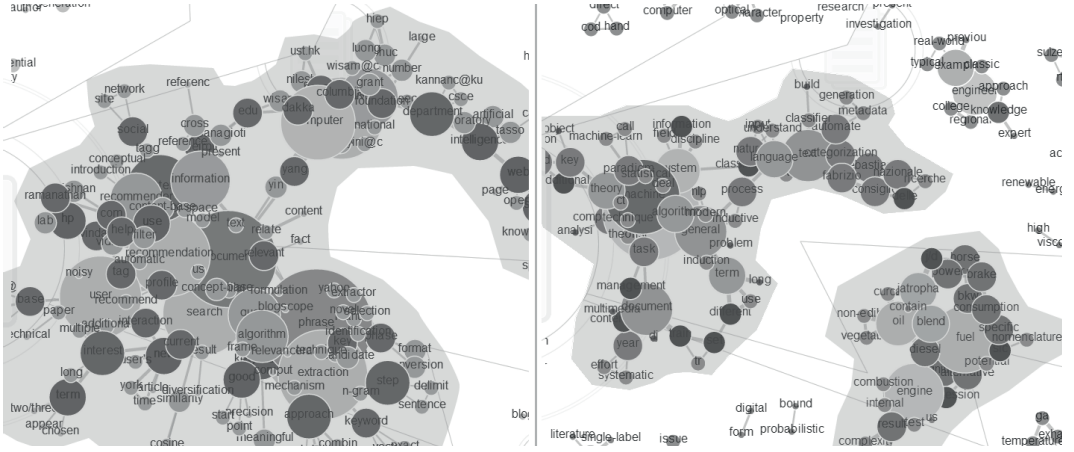

Figure 4. A comparison of a CG built from 4 articles dealing with related topics and one built with 2 unrelated articles.

allow to create, for each term, a meaningful context of interest by simply checking its adjacency list. If, in two different documents, the same term is used in similar contexts (i.e. in the two respective CGs the same nodes are connected in the same or similar way), it reasonably refers to the same concept, proving a certain degree of similarity between the two items. This mechanism also represents our solution to the problem of disambiguating polysemic terms. When, as result of a user-specified query, a set of documents is retrieved from CiteSeer, RES extracts a list of KPs from each one of the retrieved texts, builds a CG for each KP list and then generates a recommendation. Recommendations are generated in three steps: Matching/Scoring, Ranking, and Presentation. In the first step every document (D) in the SPC is matched against the user profile (U) by calculating the following parameters: *Coverage (C)*, *Relevance (R)*, and *Similarity (S)*. C represents the fraction of all the concepts present in D (referred as *totalTerms(D)*) which are also of interest for the user, since they are already included in the profile U (referred as *sharedTerms(D, U)*).

$$C(D,U) := \frac{|sharedTerms(D,U)|}{|totalTerms(D)|} \tag{4.1}$$

R estimates the importance of the concepts shared by the user profile (U) and the document (D). It is computed as the average tf-idf measure of the terms corresponding to the shared nodes between the user and the document CG with reference to the retrieved document set.

$$R(D,U) := \frac{\sum_{i \in |terms(D)| \cap |terms(U)|} tf\text{-}idf(i,D)}{|sharedTerms(D,U)|} \tag{4.2}$$

Finally, S is intended to assess the local overlap between the two CGs and to measure how relevant are the shared arcs, i.e. determine how similar are the contexts in which shared terms are used, the stronger the shared association, the higher the score. S is computed by considering the sub-graph of U ($\Pi U$) constituted by nodes shared with D; the parameter is evaluated as the sum of the weights (w) of the arcs in $\Pi U$ (E($\Pi U$)) which are also included in D (indicated as E(D)) divided by the overall weight of the arcs in $\Pi U$.

$$S(D,U) := \begin{cases} 0 & \text{if } E(\Pi U) = \varnothing \\ \dfrac{\sum_{i \in E(\Pi U) \bigcap E(D)} w(i)}{\sum_{j \in E(\Pi U)} w(j)} & \text{otherwise} \end{cases} \tag{4.3}$$

S ranges between 0 and 1. In this way, each document is considered a point in a 3-dimensional space where each dimension corresponds to one of the three above parameters. In the Ranking phase, the 3-dimensional space is subdivided into several subspaces according to the value ranges of the three parameters, identifying in such a way different regions in terms of potential interest for the user. For each dimension, low and high value ranges are identified. High values for all three parameters identify an excellent potential interest, while values lower than specific thresholds decrease the potential interest. According to the combination of the different ranges of the three dimensions, five subspaces have been identified from *excellent* to *not recommended* and each document is ranked according to where its three-dimensional representation is located. In the current experimental prototype, the interest threshold for each parameter can be adjusted at runtime, for fine tuning of the matching algorithm. Finally, in the Presentation step, documents are sorted by descending ranking order, and those sharing the same rank are ordered according to their distance from the origin; finally, the top ones are suggested to the user. As shown in Figure 2,all KPs are shown and the user can provide relevance feedback for fine adjustments of his profile and inclusion of serendipitous concepts indicated by extracted KPs that have not been previously included in the user profile.

## 5. Evaluation

Different evaluation activities have been performed in order to assess both system performance and user satisfaction. User tests have been performed with 30 volunteer master degree and PhD students who were asked to use the system in their ordinary research activity for a period of two weeks and to fill three questionnaires. The first questionnaire was proposed at the beginning of the evaluation period, the second after the first week and the last at the very end of the evaluation period. All of them were designed according to the ResQue evaluation framework [15]. As illustrated in Figure 5, test results highlight an overall good

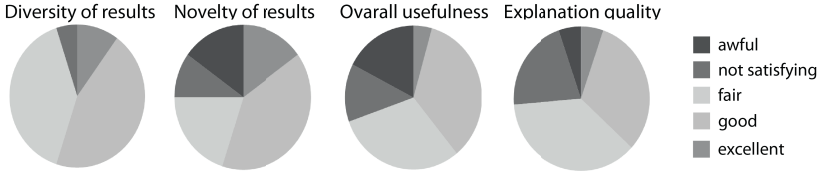


Figure 5. Summary of user perceived recommendation quality.

user satisfaction (i.e. novelty, accuracy, and diversity of recommended items, as in [15]) and the ability of the system to differentiate recommendations enough to let users discover many novel items. Particular attention was considered in assessing the quality of explanations and questionnaires results showed how most of user perceived them as sound and satisfying, while only a very small fraction found them annoying, useless, or confusing. In order to evaluate the algorithm and compare it with state of the art recommenders, we needed a data set meeting all the assumptions under which all the considered algorithms may work correctly. Unfortunately, there are no public available datasets of scientific papers including enough user ratings to make a collaborative filtering strategy able of recommending a significant part of the data set. Taking into account the lack of an adequate data set in the field of scientific literature, we decided to exploit a data set focused on a different domain, e.g. a movie data set. However, we believe that this choice is acceptable, since

the proposed content based approach is independent from the specific domain of the natural language texts considered. Following this line of reasoning, a subset of the Movielens dataset containing 100 items and 1113 users was considered. Each item had a set of user-generated keyphrases and each user a non-empty set of expressed ratings. At first, the RES algorithm was benchmarked against the most widespread content-based technique: TF-IDF, considered both in its simple, naive implementation (simply labelled as TF-IDF) and in a more sophisticated form, taking into account user rating normalization. These techniques were used to compute, for each item, a set of neighbour items ordered by descending similarity value. Different content-filtering techniques provided different neighbourhoods. Such neighbourhoods were then used, in an item-item fashion, to predict a personalized score for the target item for any user. Intuitively, the larger the neighbourhood, the higher the chances that all the items actually similar to the target one are included, moreover false positive neighbours may introduce noise, reducing the accuracy of the prediction. Items with an high predicted score were then recommended to users. These predictions and recommendations were then compared to the ones generated, on the same data set, by three collaborative techniques: a knn user-user filtering, an item-item filtering and an SVD collaborative recommender. All the collaborative-filtering algorithms were tuned to work with an optimal number of neighbours or latent semantic features. The implementation of the baseline systems was provided by the LensKit framework[16], and the TF-IDF implementation was provided by Apache Lucene. Hidden-data analysis was performed by taking into account accuracy and information gain metrics such as root mean squared error (RMSE), evaluated on rating and user basis, and Normalized Discounted Cumulative Gain (nDCG) evaluated on the whole set of recommended items and on the top 10 items of the list (the ones more likely to be consumed by the user). The RES algorithm proved to be able to perform well even with a small neighbourhood, converging quickly towards accuracy and information gain values that other content-based algorithms reach only when considering very large neighbourhoods (forty or more) as shown in Figure 6. Such large numbers of neighbours imply that, to generate meaningful recommendations, a large quantity of data is needed, which may be not always available. Benchmark against collaborative filtering algorithms proved RES to be on par with the most widely

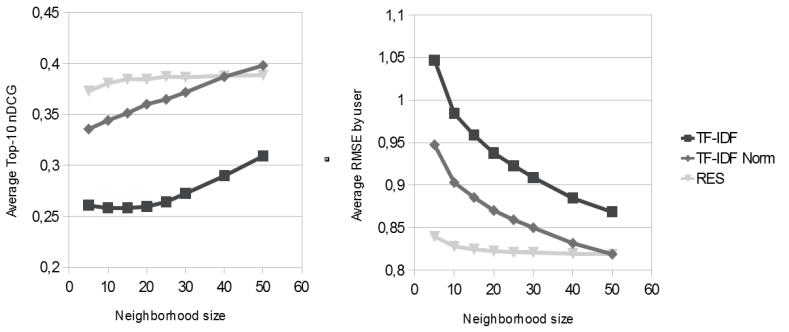


Figure 6. Comparison of accuracy (RMSE) and information gain (top-10 NDCG) between RES and two TF-IDF based techniques.

used techniques, performing slightly worse than SVD-based techniques, almost on par with item-item collaborative filtering, and slightly better than user-user filtering as shown in Figure 7. However, while showing similar levels of performance, RES has the advantage of using a scrutable user model, allowing explanation of recommendations, whose lack is one of the major drawbacks of collaborative techniques and SVD-based techniques in particular. However it is important to point out how our system, being content-based, does not need rating data to provide recommendations, i.e. does not suffer the cold start problem.

## 6. Conclusions

Recommender systems can greatly facilitate the task of searching for scientific literature, however, by just filtering collection of papers, state-of-the-art recommender systems still leave a heavy work to researchers who have to spend efforts and time for accessing the knowledge contained in scientific publications. This issue is faced in this paper, where we propose a mechanism where concepts are automatically

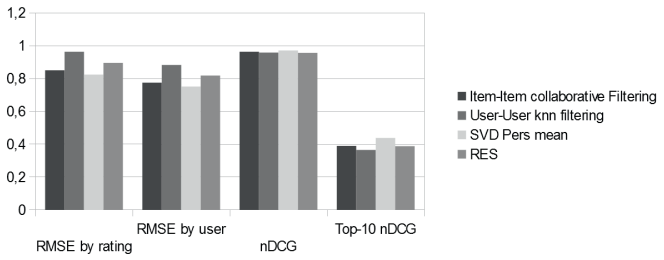


Figure 7. Comparison of accuracy and information gain between the RES algorithm and CF techniques.

extracted from papers in order to generate and explain recommendations. Our semantic approach to the problem allows the creation of a user model that is both based on actual concepts of interest and understandable while maintaining performance comparable with that of state-of-the-art collaborative recommenders. The presented RES system is still a testbed and experimentation is ongoing, but results gathered so far are encouraging, proving that our concept-based, human understandable approach is able to effectively support navigation in large digital libraries. Future work will be aimed at expanding our concept-based strategy by exploiting different sources of knowledge in order to identify synonymous terms and phrases, suggesting to the users new concepts related to the ones he considers interesting, and overcome the limitations of a pure content-based approach. Finally, encouraged by the results provided by the benchmark test on the Movielens dataset, we will also address the possible advantages of utilizing our approach in other domains such as news, patents or legal documents archives.

## References

1. Jannach D, Zanker M, Felfernig A, Friedrich G, Recommender systems: an introduction, Cambridge University Press, 2010.
2. Van Raan A. F, Sleeping beauties in science, Scientometrics 59 (3) (2004) 467–472.
3. Ferrara F, Tasso C, Extracting keyphrases from web pages, in: Agosti M, Esposito F, Ferilli S, Ferro N (Eds.), Digital Libraries and Archives, Vol. 354 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2013, pp. 93–104.
4. Bollacker K. D, Lawrence S, Giles C. L, Discovering relevant scientific literature on the web, Intelligent Systems and their Applications, IEEE 15 (2) (2000) 42–47.
5. Bogers T, Van den Bosch A, Recommending scientific articles using citeulike, in: Proceedings of the 2008 ACM conference on Recommender systems, ACM, New York, NY, USA, 2008, pp. 287–290.
6. Huynh T, Hoang K, Do L, Tran H, Luong H. P, Gauch S, Scientific publication recommendations based on collaborative citation networks., in: Smari W. W, Fox G. C (Eds.), CTS, IEEE, 2012, pp. 316–321.
7. Doerfel S, Jäschke R, Hotho A, Stumme G, Leveraging publication metadata and social data into folkrank for scientific publication recommendation, in: Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web, ACM, New York, NY, USA, 2012, pp. 9–16.
8. Kim H.-N, El Saddik A, Personalized pagerank vectors for tag recommendations: inside folkrank, in: Proceedings of the fifth ACM conference on Recommender systems, ACM, New York, NY, USA, 2011, pp. 45–52.
9. Jiang Y, Jia A, Feng Y, Zhao D, Recommending academic papers via users' reading purposes, in: Proceedings of the sixth ACM conference on Recommender systems, RecSys '12, ACM, New York, NY, USA, 2012, pp. 241–244.
10. Chandrasekaran K, Gauch S, Lakkaraju P, Luong H. P, Concept-based document recommendations for citeseer authors, in: Proceedings of the 5th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 83–92.
11. Govindaraju V, Ramanathan K, Similar document search and recommendation, Journal of Emerging Technologies in Web Intelligence 4 (1) (2012) 84–93.
12. Cilibrasi R. L, Vitanyi P. M, The google similarity distance, Knowledge and Data Engineering, IEEE Transactions on 19 (3) (2007) 370–383.
13. De Nart D, Ferrara F, Tasso C, Personalized access to scientific publications: from recommendation to explanation, in: User Modeling, Adaptation, and Personalization, Springer, 2013, pp. 296–301.
14. Ferrara F, Pudota N, Tasso C, A keyphrase-based paper recommender system, in: Agosti M, Esposito F, Meghini C, Orio N (Eds.), Digital Libraries and Archives, Vol. 249 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2011, pp. 14–25.
15. Pu P, Chen L, Hu R, A user-centric evaluation framework for recommender systems, in: Proceedings of the fifth ACM conference on Recommender systems, ACM, 2011, pp. 157–164.
16. Ekstrand M. D, Ludwig M, Konstan J. A, Riedl J. T, Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit, in: Proceedings of the fifth ACM conference on Recommender systems, ACM, 2011, pp. 133–140.