Report: From Recommender Systems to Personalized Academic Search Engines

1. Introduction

I have applied for a COST Keystone scholarship in order to travel to the University of Cyprus in Nicosia on Cyprus from 4th June 2014 to 3rd July 2014. With the help of my host Dr. Georgia Kapitsaki, I planned on implementing a search engine for academic papers into the open source literature management software *Docear*. For this we wanted to re-use large parts of the existing academic paper recommender system of Docear and recommend search queries to users. Our goal is to study, whether search engines with integrated query recommender system can be used as augmentations or even substitutions for recommender systems.

Implementing Docear's academic paper recommender system took us several years and in our last evaluation[1] we analyzed 240,948 recommendations delivered to 4,153 users between April 2013 and June 2014. As a matter of course, my four weeks stay was not enough to collect similar amounts of data. It was, however, enough to implement the academic search engine into Docear's, release it as a *semi-open* beta version and to present first data on the usage of the search engine.

2. The Academic Search Engine

I spent most of my time on Cyprus with designing and implementing the search engine and creating the data structures necessary in order to collect usage data. Hence, this report is mainly of technical substance. The limited amount of presented research results are only based on a very limited amount of data, collected by a beta version of Docear and results should be considered with caution.

Anyway, I want to state, that the help and experience of Dr. Kapitsaki were essential factors for the success of the search engine's realization. I will very gladly analyze the search engine's usage together with her and publish the results, as soon as we have gathered the necessary amounts of data.

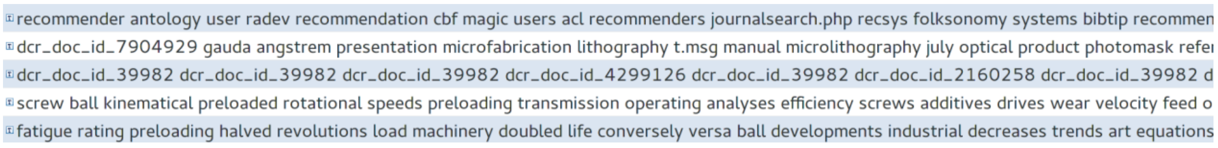
2.1. Plans and Adjustments

In order to implement the search engine prototype I planned the following (see the application for details on these tasks):

- To adjust the algorithms of the recommender system in order to create user models matching the user's interests and fitting as queries for a search engine
- To implement the search engines UI components and re-use the recommender system's UI for the presentation of the search results
- To create the data model in which the usage based data can be collected

[1] I submitted the evaluation of Docear's academic paper recommender system with the title "On Docear's Users and the Comparability of Recommender Systems Evaluation" to the *ACM RecSys Workshop on Recommender Systems Evaluation: Dimensions and Design (REDD 2014)*. It is currently in review.

Docear’s recommender system already creates user models matching a user’s interest. Docear user models contain both terms and citation data. Figure 1 shows an excerpt of user model examples as created and used by Docear’s recommender system.



recommender antology user radev recommendation cbf magic users acl recommenders journalsearch.php recsys folksonomy systems bibtip recommen
dcr_doc_id_7904929 gauda angstrem presentation microfabrication lithography t.msg manual microlithography july optical product photomask refe
dcr_doc_id_39982 dcr_doc_id_39982 dcr_doc_id_39982 dcr_doc_id_4299126 dcr_doc_id_39982 dcr_doc_id_2160258 dcr_doc_id_39982 d
screw ball kinematical preloaded rotational speeds preloading transmission operating analyses efficiency screws additives drives wear velocity feed o
fatigue rating preloading halved revolutions load machinery doubled life conversely versa ball developments industrial decreases trends art equations

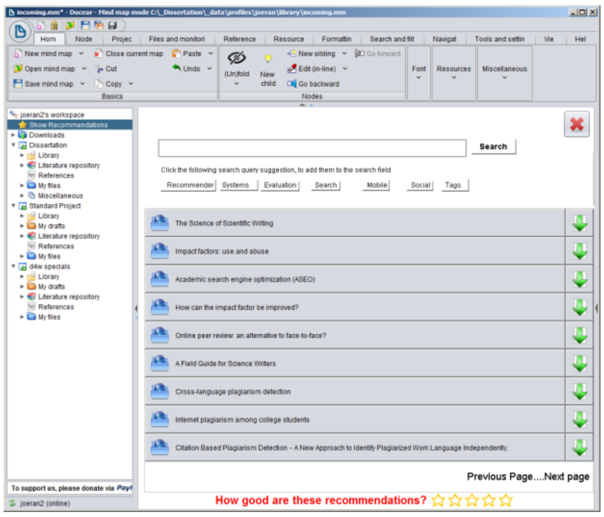
Figure 1: Data Models created and used by Docear's recommender system

User models can contain up to 1,500 items, which are stored according to their relevance in descending order. Thus, the most important items of a user models are at the beginning, while less important items can be found near the end. Citations have the prefix “dcr_doc_id_” followed by an ID of documents cited by the user. Their content is not understandable by humans without any additional information. They are hence not fit to be recommended as search terms to users.

Since terms in user models are retrieved from user data based on their relevance, they do not necessarily describe the same concept. For instance, the terms “fatigue” and “life” are probably not chosen by users to be searched together with “ball”, “trends”, “art” and “equation” (see the last example of Figure 1).

When consulting with Dr. Kapitsaki we agreed on the following changes to take the described facts into account:

1. Instead of using algorithms to create user models that can be used as search queries, we just prune and cut user models that are created for users by the recommender system anyway. Citations are removed from user models and the user model is subsequently cut to a length of 20 terms or less.
2. Instead of showing a full generated search query to the user, which he would need to edit, we just recommend concepts, i.e. terms, to the user. He can decide to add these concepts to his search query.



Click the following search query suggestion, to add them to the search field

The Science of Scientific Writing
Impact factors: use and abuse
Academic search engine optimization (ASEO)
How can the impact factor be improved?
Online peer review: an alternative to face-to-face?
A Field Guide for Science Writers
Cross-language plagiarism detection
Internet plagiarism among college students
Citation Based Plagiarism Detection – A New Approach to Identify Plagiarized Work Language Independently.

How good are these recommendations? ☆☆☆☆☆

Figure 2: Draft of the search feature's user interface

We then drafted the user interface (Figure 2). The search panel contains a text box for the *search query*, and buttons showing the concepts of the *search model*. Clicking a concept adds it to the text box. Clicking the “Search” button starts the search using the search query inside the text box.

2.2. Research Questions and the Data Model

At the beginning of the STSM, I asked the following three questions I wanted to answer based on my work on Cyprus:

- Do researchers use both the search module and the recommender system mutually or do they favor one over the other?
- Is searching with manually modified search queries more effective (in terms of the click through rate and the overall clicks on returned documents) than the automatic recommender system?
- The computation time for recommendations heavily depends on the chosen algorithm. Especially the comparison between user model and digital library seems to increase strongly with the size of the user model. The overall computation time of recommendations ranges from a few seconds up to a few minutes, which renders ad-hoc creation of recommendations on a user’s request impossible. Is ad-hoc creation of helpful search terms for the search engine possible within a reasonable amount of time?

In order to answer these questions, data on the usage and parameters of the search engine has to be collected. A first draft of the *Entity Relationship* (ER) model can be seen in Figure 3.

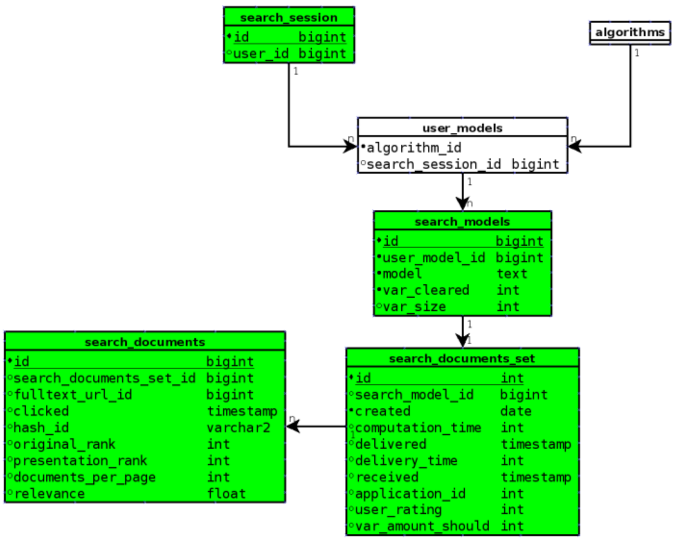


Figure 3: First Draft of the Search Engine’s ER-Model

Green entities represent the new tables used to collect the data. White entities represent existing tables used by Docear’s recommender system. Columns from existing tables, which are not important for the search engine, were left out. The idea is to inherit the *search model* (the search terms recommended to the user) from the full *user model* (see Figure 1). A *document set* is a container for all retrieved *documents* retrieved by querying the search engine. A *search session* helps to store when a user queries new documents using a new search model.

The design had some flaws:

1. The described model needed to retrieve all documents from the search engine at once. Giving a user the possibility to browse through multiple pages of the same document set, meant to retrieve a maximum of, e.g. 1,000 documents for 100 pages of documents from the full text index and store the results in the database.
2. The search model and the query string the user actually used to query the search engine are not necessarily the same. For instance, if the search model recommended to the user contained the terms “recommender”, “system” and “TF-IDF”, the user could decide to use only a part of it and add own search terms to form the query “recommender system influence of user age”. The user could also decide to issue another query “recommender system influence of gender”. Storing both queries in the same table (e.g. in another column) as the search model is stored in, is impossible without breaking basic rules of data modeling.
3. Searching the documents in Docear’s digital library must be possible even if no search model was retrieved for the user. That is important if we want to allow searching the documents from a 3rd party application or websites using Docear credentials. Hence, the *user_id* has to be stored inside the search engine’s tables which store the results of a search, as both search models and user models are not necessarily used.

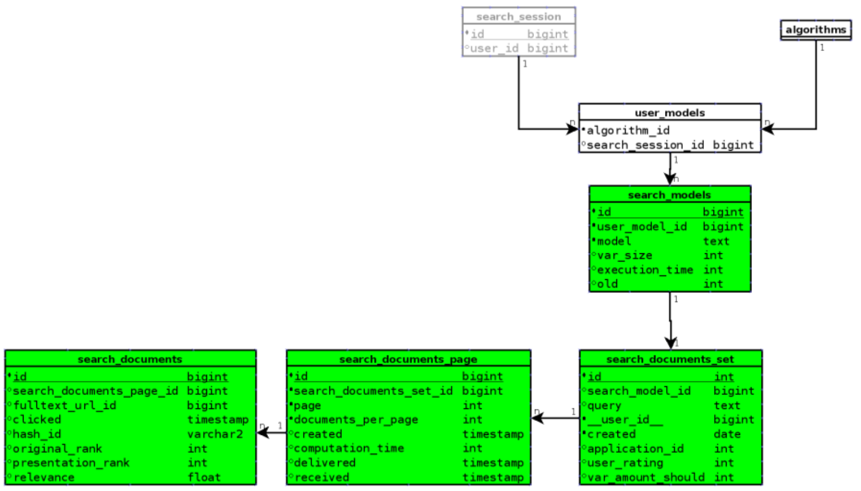


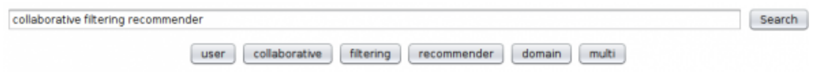
Figure 4: Final Draft of the Search Engines ER-Model

Figure 4 shows revised draft of the ER-model.

A new table storing a page was introduced into the system. With this change it was possible to retrieve only one page of data from the full text index and retrieve the next one, i.e. issue a new query, when the user navigates to the next result page. Instead of waiting a long time before all (e.g. 1,000) documents have been retrieved and then navigating very quickly through the result pages, with this new design the retrieval time is always relatively short. The new design also requires less storage space, as users will probably not always browse through all result pages. Since *one* set of documents is retrieved using *one* search query, the query is stored inside the document set’s table. This table also contains the user information needed to study the usage results. The search session table was removed for now. It is only necessary if we decide to let a user retrieve multiple search models inside of the same session, i.e. without closing the search panel in Docear.

3. Status and Usage Data

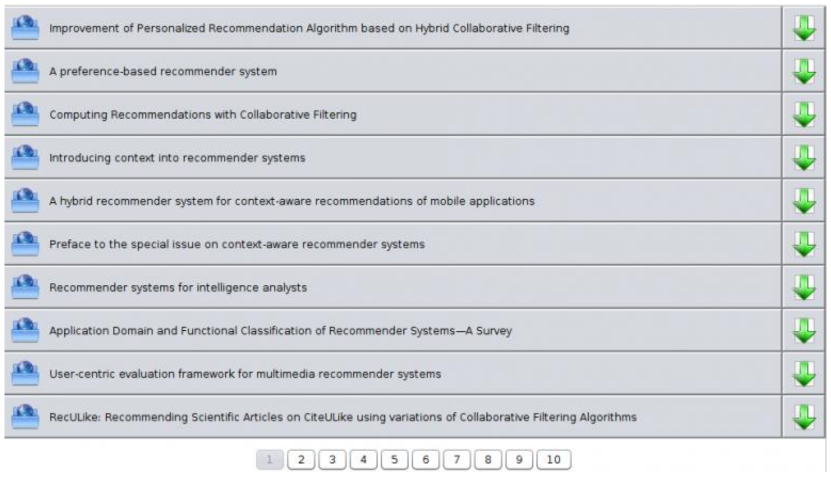
The whole search engine was planned and implemented on Cyprus. On Docear itself a total of 62 source files were changed with 2,996 inserted and 2,216 deleted lines of code[2]. In Docear's web service, which serves as interface to the digital library, a total of 52 files were changed with 2,276 inserted and 2,093 deleted lines of code. The changed versions of both Docear and the web service serve as a stable basis for the research questions we want to answer.



collaborative filtering recommender

Figure 5: Document Search Interface in Docear

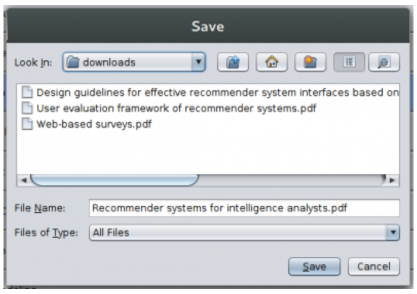
Figure 5 shows the search UI of Docear. The buttons beneath the search box show the six terms of the search model recommended to the user. Three of the buttons were clicked and the corresponding (clicked) terms were added to the search query. A user can use the search terms, add new ones or completely discard them. If he searches for documents a result list appears (Figure 6).



Improvement of Personalized Recommendation Algorithm based on Hybrid Collaborative Filtering
A preference-based recommender system
Computing Recommendations with Collaborative Filtering
Introducing context into recommender systems
A hybrid recommender system for context-aware recommendations of mobile applications
Preface to the special issue on context-aware recommender systems
Recommender systems for intelligence analysts
Application Domain and Functional Classification of Recommender Systems—A Survey
User-centric evaluation framework for multimedia recommender systems
RecULike: Recommending Scientific Articles on CiteULike using variations of Collaborative Filtering Algorithms

Figure 6: Search Results Displayed in Docear

A user can open a document from the result, by clicking the title. The default web browser of the system will open the link to the document's URL. He can also directly download a document to Docear's literature folder, by clicking the green arrow icon next to a document. If a user clicks the green arrow icon a file chooser dialog will open (Figure 7). The document's title is used as file name by default. Users can, of course, change the name before saving the document.



Look In: downloads

Design guidelines for effective recommender system interfaces based on
User evaluation framework of recommender systems.pdf
Web-based surveys.pdf

File Name: Recommender systems for intelligence analysts.pdf
Files of Type: All Files

Figure 7: File Chooser Dialog to Save A Document

[2]Docear is open source. The source code, its changes and the commit messages can be viewed on Github: https://github.com/Docear/Desktop/commits/master

On July 1st 2014, an experimental version was released in Docear's forum (the experimental version board)[3]. Since no serious bugs were reported by Docear's users this version was published as *semi-open* beta version in our blog[4]. That is, users were informed about the new Docear version, that it is still a test version. The download links were only provided in the blog post and not officially offered through the download section of Docear's website. This makes sure, that in case of errors, only users who want to test new features, are affected. We plan to officially release the beta version with minor code changes in the next weeks.

While this procedure assures a maximum of trust of users in the stability of Docear, it takes time until many users use a new version and thus, data can be collected. Until now, a total of 8,442 search models were created from the recommender system's user models, but only 49 were delivered to only 14 different users. The search engine has delivered 116 document sets containing to the user and 267 result pages were viewed by users. A total of 2,487 documents were delivered to the user and 58 of them were clicked. The *click through rate* (CTR), hence, is only 2.43%. The recommender system achieves a CTR of around 4% to 5% on average. Users of the new beta version of Docear are not necessarily interested in academic papers. On the one hand, we discourage the use of beta version of Docear for daily academic work. On the other hand and based on the posts in our bug reports forum, I believe, that most users who enjoy testing new features of Docear are more interested in mind mapping features of Docear. Searching the digital library takes only 7 ms on average, which is considerably faster than the recommender system, which takes 2,082 ms on average. The main reason is that search queries (with 2.3 terms on average) are usually much shorter than user models (with 621 terms on average).

4. Summary and Outlook

During the STSM, the search engine was planned, implemented and released as semi-open beta version to Docear users. The data model was planned, implemented and collects the data necessary to answer the research questions presented in section 2.2 of this document. Dr. Kapitsaki's ideas on how to study and compare the effectiveness of Docear's recommender system to the search engine were crucial for the goals achieved during the STSM. Her experience with the inner mechanics of Docear's recommender system also helped to finish the implementation part of the project.

While this is a lot of progress for only four weeks of time, there is not enough data to answer any questions about the effectiveness of the search engine yet. Therefore, we will continue evaluating the use of the search by end-users of Docear during the next months in order to gather more results. Moreover, together with Dr. Kapitsaki we are considering the extension of the search mechanism and the potential integration of semantic features. A follow-up visit during the next year would be valuable for the continuation of this work. I hope I will find both the time and the funding to visit Dr. Kapitsaki again during the next year, to analyze the data we have collected in the meantime. As a continuation of the collaboration I have established with UCY through this STSM, I will also encourage other students who might collaborate in our work on Docear's recommender system or search engine to visit the UCY in Nicosia and I will gladly accept and mentor interns from the UCY to work in Magdeburg.

[3]http://www.docear.org/support/forums/docear-support-forums-group3/experimental-releases-forum8/docear-1-1-1-0_devel_build238-introducing-online-document-search-thread980

[4] http://www.docear.org/2014/07/08/docear-1-1-1-beta-with-academic-search-feature/