

A Federated Recommender System for Online Learning Environments

Lei Zhou¹, Sandy El Helou², Laurent Moccozet³, Laurent Opprecht³,
Omar Benkacem³, Christophe Salzmann², and Denis Gillet²

¹Tongji University,
200092 Shanghai, China

lake.zhou.sh@gmail.com

²Ecole Polytechnique Fédérale de Lausanne (EPFL),
1015 Lausanne, Switzerland

{sandy.elhelou, denis.gillet, christophe.salzmann}@epfl.ch

³University of Geneva (UNIGE),

Geneva, Switzerland

{laurent.moccozet, laurent.opprecht, omar.benkacem}@unige.ch

Abstract. From e-commerce to social networking sites, recommender systems are gaining more and more interest. They provide connections, news, resources, or products of interest. This paper presents a federated recommender system, which exploits data from different online learning platforms and delivers personalized recommendation. The underlying educational objective is to enable academic institutions to provide a Web 2.0 dashboard bringing together open resources from the Cloud and proprietary content from in-house learning management systems. The paper describes the main aspects of the federated recommender system, including its adopted architecture, the common data model used to harvest the different learning platforms, the recommendation algorithm, as well as the recommendation display widget.

Keywords: Technology-Enhanced Learning, Personal Learning Environments, Federated Recommender System, Web 2.0.

1 Introduction

Recommender systems have led to a shift from the user-active mode in search engines to the information-active mode in information discovery where interesting items are channeled to the user without (necessarily) an explicit request [1].

From e-commerce (e.g. Amazon¹) to social networking sites (e.g. Facebook², LinkedIn³), recommender systems are gaining more and more popularity whether to recommend connections, news, resources to access, or products to purchase.

¹ <http://www.amazon.com>, 06-06-2012

² <http://www.facebook.com>, 06-06-2012

³ <http://www.linkedin.com>, 06-06-2012

With the development of computer technology, learning is no longer limited to classrooms. Technology-Enhanced Learning (TEL) aims at designing, developing and testing socio-technical innovations to support learning practices at both the individual and the organizational level [2]. It helps the learners to learn anywhere and anytime. More and more online platforms, such as Symbaloo⁴ and Graasp⁵, enable users to access, share, and organize learning resources of all kinds, and manage their own Personal Learning Environments (PLEs). PLEs give students the opportunity to create, organize, repurpose and package their learning content and tools, increasing by that the learning efficiency and effectiveness [3]. Unlike traditional LMS (Learning Management Systems), PLEs are user-centered rather than teacher or course-centered. The shift from LMS to PLEs regarding educational resources introduces an overflow and a distortion: with course-centered LMSs, students are directly provided with a limited amount of selected and dedicated resources whereas with learner-centered PLEs, students have to face a large amount of resources for which they have to perform their own selection. Recommender systems therefore play a key role in open environments such as PLEs helping learners find what matches their interests from a pool of resources which, far from being preplanned and limited, can be added, augmented, and repurposed at run time [4].

Throughout their studies in Swiss universities, students are often exposed to more than one online learning platform. As a matter of fact, the mobility of students across different universities is rising. At the same time, Swiss universities do not all rely on the same online learning platform, even though they have adopted a common single sign-on mechanism. In addition, within a single university, there could be more than one platform used [5]. It becomes essential to establish a system that integrates resources from different platforms, so that students can construct a richer learning environment and access additional distributed resources.

In this paper, we propose a federated recommender system that relies on people's interaction with distributed resources in order to deliver rich and relevant recommendations. Such recommendations can serve as a central access point for learners, enrich platforms used locally (within one institution) by bringing data from other universities and help create learning networks across Switzerland. The platforms themselves consist of LMS and PLEs serving different learning communities from Swiss Universities and including Graasp, Moodle⁶, and Mahara⁷.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the main components of the federated recommender system. Section 4 concludes the paper and discusses future work.

2 Related Work

In this section, we discuss existing research around the two essential aspects of federated search or recommendation in learning contexts: the federation approach and

⁴ <http://www.symbaloo.com>, 06-06-2012

⁵ <http://graasp.epfl.ch>, 06-06-2012

⁶ <http://moodle.unifr.ch>, 06-06-2012

⁷ <https://mahara.org>, 06-06-2012

architecture. The federation approach concerns about how to get effective data, filter and merge data, while the federation architecture pays more attention to the structure with which the system can be efficiently implemented.

The study summarized in [6] focuses on ranking and merging search results from different resources. In order to efficiently combine different search results and improve performance, several algorithms are tested and compared on ObjectSpot⁸, a vertical federated search engine for academic papers and other learning objects. In [7] search results from different platforms are collected and re-ranked by a federated search service. Ranked search results are sent to a widget and displayed to the user. Re-ranking is performed as follows: the widget sends search result URLs (Uniform Resource Locators) to the recommendation service. This service returns personalized recommendations after analyzing attention metadata stored in the database [7]. Attention metadata consists of recorded users' interaction with the recommended items (i.e. preview, like, dislike) in order to leverage user's interest and rank items' global and local popularity (within the target users' network). By mining this metadata, recommended lists for similar queries is improved in subsequent search requests.

Several papers discuss the architecture adopted for federated search or information retrieval from distributed repositories. The multi-agent based architecture proposed in [8] has a master agent that interacts with users and dispatches search requests to service agents of different repositories. Each of the repositories has a service agent that collects and updates metadata, and processes requests sent by the master agent. Once the service agents return their results, the master agent combines the results and displays them to users. In AIREH (Architecture for Intelligent Recovery of educational content in Heterogeneous Environments) [9], an intermediary communication point is proposed for retrieving learning objects from heterogeneous environments. The intermediary communication point is responsible for sending user requests to different repositories and then merging the returned ranked results. In Ariadne [10], a harvester gets metadata belonging to new resources from different learning repositories regularly and incrementally. The efficiency of processing search requests is improved by checking metadata. The OpenScout system [11] follows the same approach.

Our federated engine also relies on a harvester to gather data across different platforms. However, we not only store metadata related to learning objects (or resources) but also interaction data linking users and learning objects. As a result, the federated can offer a personalized recommendation, where personal interests are inferred from user actions. In addition, recommended items are not limited to learning objects, but also include activity spaces and users.

As in [7], we developed a portable widget in order to display recommendation results. A widget is a small Web application that is embeddable in a Web page and allows easy mashups [11]. This choice leads to a lightweight interoperability and an easy integration of the recommendation display service in the online learning platforms involved. The developed widget allows learners to classify recommended items by type (e.g. resources, users, or activity spaces) and gives an overview of the recommended items, whilst providing a link pointing to its physical location in one of the harvested learning platforms.

⁸ <http://www.objectspot.org>, 06-06-2012

3 Proposed Federated Recommender System

3.1 Common Data Model

The federated recommender system exploits interaction data stored in different online learning platforms. “Considering that data models, naming conventions, and formats vary from user to user and network to network” [12], it becomes crucial to adopt a standard data representation and exchange format across different platforms.

Graasp is one of the online learning and collaboration platforms considered. It was built on top of the 3A interaction model illustrated in Fig. 1. We adopt this lightweight model, as a reference model to exchange data across the different learning platforms involved. The 3A interaction model is based on the following main constructs [13]:

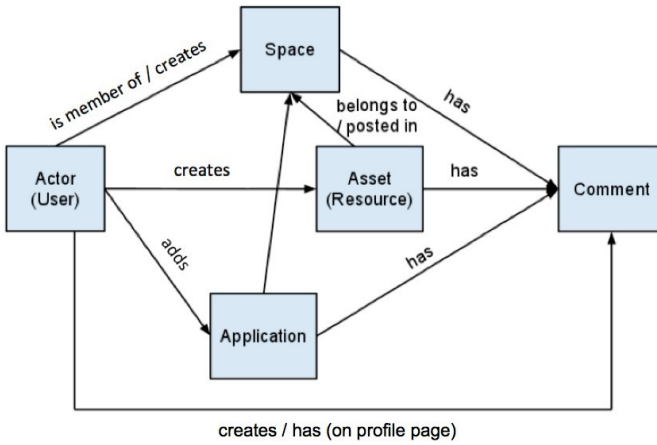


Fig. 1. Common Data Model

- Actors: refer to users or people. Initially, the concept of actors integrated also the notion of applications (and more generally any entity capable of initiating an event). Later, a clear separation between human actors and applications was adopted.
- Assets or resources: refer to different kinds of resources or digital artifacts (files, presentations, videos, wikis and etc.) created and shared among actors.
- Activity spaces: refer to online contexts or containers where one or more actors share resources and applications, under the umbrella of an activity space serving an explicitly or implicitly stated purpose. For instance, for every course in a formal learning context, a main activity space can be created to which all students, tutors, and teaching assistants are invited to join. In the same way, a community could create an activity space to which all its members are invited.

Each learning platform involved (e.g. Moodle, Mahara) maps its data into the 3A model and offers REST (Representational State Transfer)⁹ APIs that are called by the harvester to get data in XML or JSON (JavaScript Object Notation)¹⁰ formats.

⁹ <https://www.ibm.com/developerworks/webservices/library/ws-restful/>, 06-06-2012

¹⁰ <http://www.json.org/>, 06-06-2012

3.2 Architecture

The adopted architecture for the federated recommender system is illustrated in Fig. 2. Its main building blocks are:

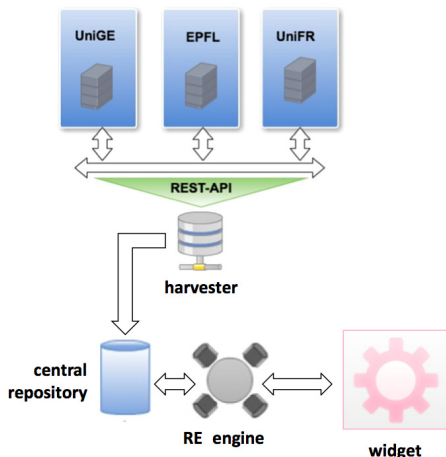


Fig. 2. Architecture of the federated recommendation system

- A harvester, which periodically retrieves the metadata from different repositories through REST APIs, parses retrieved data, and stores it in the central repository;
- A recommendation engine (or RE), which runs the recommendation algorithm regularly and then stores the results into the central repository;
- A widget, which queries the central repository using REST APIs and displays recommended items to target users. This widget can be embedded in any platform that implements the adopted REST APIs, such as Graasp, ELGG¹¹, Liferay¹² or Moodle.

3.3 Data Collection: The Harvester

The harvester serves as a middleware between the learning platforms and the central repository. Using the common REST APIs implemented by the learning platforms, the harvester regularly retrieves the metadata, parses it based on the 3A model taxonomy, and updates the central repository. For the sake of privacy, only public and closed items are retrieved. Public items consist of actors, spaces, resources and applications accessible to everyone without any restriction. The content of closed spaces on the other hand, are only accessible to authorized members, while everyone else can only see basic information (e.g. description, title, keywords).

¹¹ <http://www.elgg.org>, 06-06-2012

¹² <http://www.liferay.com>, 06-06-2012

During the harvesting operation, two tables are used to store the gathered data.

- An indexing table which stores the URLs of the harvested items;
- A queue table which stores the URLs that are in the waiting list to be harvested;

We describe hereafter the steps undertaken by the harvester and the role of the above tables in the data retrieval and storage process.

Step 1: Call the REST APIs, get the metadata and check item status.

Each harvested item is identified with a unique URL. The harvester calculates the hash value of the URL and the API response, and stores them in the index table. When the harvester calls the REST APIs, it will first check if the item's URL exists in the index table. If the URL exists and the hash value of the API response is still the same, nothing is done; if the URL exists but the hash value has changed, the item is updated; otherwise, it is a new item and it gets added into the table. If calling the defined APIs returns 404 errors for an existing item, it means that the item has been deleted from the platform and, as a result, it is also deleted from the index table. This check is important, as in some platforms such as Moodle, it is not possible to query the harvester every time an item is updated or deleted.

Step 2: Parse the response and store information in central repository.

For each URL stored in the index table, the response is parsed based on the 3A model taxonomy and the item's metadata (e.g. name, thumbnail image, description) are stored into the central repository.

Step 3: Check item's child elements.

Spaces might contain members, resources and subspaces as child elements. Fig. 3 shows the XML response data for a space item containing a subspace, a resource and an actor, including their metadata. For each child element, the URL is stored in the queue table.

Step 4: Explore items stored in the queue table.

For each item stored in the queue table, the harvester goes through steps 1 to 3. The top-down recursive process ends when there are no more URLs in the queue table.

```

▼<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aaa="http://ple.unige.ch/re/1#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://localhost/mahara/artefact/graaasp/service/data/"
  ▼<rdf:Description rdf:about="36e7e81b9d12dababcbe2febaba8b521-2">
    <aaa:id>36e7e81b9d12dababcbe2febaba8b521-2</aaa:id>
    <aaa:dataset>http://localhost/mahara/</aaa:dataset>
    <aaa:url>http://localhost/mahara/group/view.php?id=2</aaa:url>
    <aaa:name>test for creating space</aaa:name>
    ▶<aaa:description rdf:parseType="Literal">...</aaa:description>
    <aaa:type>space</aaa:type>
    <aaa:spaces rdf:resource="7e14ed175ce447742753df686bdadaa0-2"/>
    <aaa:assets rdf:resource="63bdeaa8f804a116dd038e61b1452b1c-6"/>
    <aaa:actors rdf:resource="857c9a58e87a91b1a6044fe1e6dbaf26-1"/>
    <aaa:kind>group</aaa:kind>
    <rdf:type rdf:resource="http://ple.unige.ch/re/1#space"/>
    <rdfs:label>test for creating space</rdfs:label>
    ▶<rdfs:comment>...</rdfs:comment>
    ▶<dc:description rdf:parseType="Literal">...</dc:description>
  </rdf:Description>
</rdf:RDF>

```

Fig. 3. Example of metadata related to a space

3.4 Data Processing: Recommendation Engine

The recommendation engine (or RE) uses the 3A ranking algorithm to compute the relative importance of existing items for a target user. The 3A ranking algorithm is derived from the original pagerank algorithm, which was designed by Page and Brin for ranking hypertext documents for Google [14]. Its main idea can be summarized as follows: if the owner of page j links to a page i , he/she is indicating that page i is important to check. The more incoming links a page has, the more important or authoritative it is. In addition, pages referred to by authoritative pages are considered as more important, and therefore ranked higher than pages linked by non-authoritative ones. Following this idea, item ranks are computed by applying a random walk algorithm on an adjacency graph connecting pages together, with two main parameters: one deciding the probability to choose a random page and the other deciding the probability to follow a page link.

The 3A ranking algorithm follows the same idea except that its graph consists of nodes representing heterogeneous items, connected by edges representing different relation types with equal or different importance weights. For instance, for a space member, an edge having the “membership” weight connects the user node to the space node as a sign of interest of that user in the space. The algorithm’s details and probability equation can be found here [15].

In order to apply the algorithm to the harvested data, the RE constructs a graph where each public or closed item fetched is represented with a node. Unidirectional and bidirectional relations between any two items are respectively mapped in the graph by a single edge or two edges in opposite directions connecting the corresponding nodes.

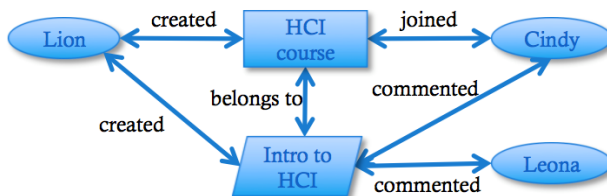


Fig. 4. Example for recommendation graph

Fig. 4 illustrates a simple example of a 3A recommendation graph. In the graph, ovals represent user nodes while rectangles represent space nodes and parallelograms represent resources. Arrows represent one way and two-way relationships between items. Since Leona commented on the «Intro to HCI », she is supposed to be interested in HCI. As the author of the resource «Intro to HCI » and the creator of the «HCI course » space, Lion clearly shares the same interests. So does the «HCI course » space member, Cindy. Running the 3A recommendation algorithm with Leona, as the target user, will help her discover spaces and people with shared interests (e.g. Lion, Cindy, and «HCI course ») by following their inter-connections.

3.5 Data Visualization: Widget

The widget (open social gadget) is used to display the recommended items and can be applied across different platforms. Its interface is shown in Fig. 5. It is implemented with JavaScript. For each user (student or tutor), the API will return a personalized list of recommendation items, displayed on different rows, according to their type.

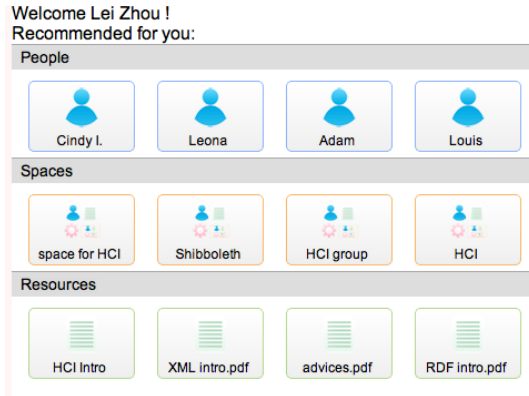


Fig. 5. The interface of the widget

When the user clicks on a recommended item, it is the responsibility of the platform where it was actually posted to check whether the user has enough rights to access the item at hand. Swiss universities have implemented Shibboleth [16] as a single sign-on (SSO) mechanism. Shibboleth allows users to login once and have a unique identity across the different platforms and the recommendation widget. Hence, the navigation between the various platforms thanks to the widget can be rendered easy. Once got privileged by the responding platform, the user can do the operations as usual.

4 Conclusion and Future Work

This paper presents a relation-based recommender system, which delivers federated recommendation. According to the common data model, the harvester collects and parses metadata from heterogeneous sources. Then, the recommendation engine runs the 3A recommendation algorithm and computes the relative importance of all items based on the metadata stored in the central repository. Each target user gets the recommendation automatically in the widget and is able to access recommended items from the harvested learning platforms.

In the near future, we plan to adopt a hybrid approach for the recommendation engine. For new users, the recommendation engine can hardly provide useful recommendation without sufficient user actions. In the hybrid approach, the 3A relation-based recommendation technique will be combined with content-based analysis, in order to alleviate the cold-start problem. Content-based analysis consists

of computing content similarity between items (by comparing different fields such as item title, description and keywords). Then the results of content-based filtering consisting of item-to-item similarity measures are fed into the 3A recommendation algorithm as an extra-relation.

An evaluation with students from different Swiss universities is also planned. The evaluation will focus on the widget's interface usability and the impact of federated search and recommendations as compared to local ones. The usefulness of recommending items from various online platforms used within one university as well as across different universities will be assessed.

Acknowledgment. The research work described in this paper is partially funded through the Personal Learning Environment (Phase 3) project of the Swiss AAA/SWITCH program and the ROLE Integrated Project; part of the Seventh Framework Program for Research and Technological Development (FP7) of the European Union in Information and Communication Technologies.

References

1. Guy, I., Jaimes, A., Agulló, P., Moore, P., Nandy, P., Nastar, C., Schinzel, H.: Will Recommenders Kill Search? Recommender Systems – an Industry Perspective. In: Proceedings of the 4th ACM Conference on Recommender Systems, pp. 7–12 (2010)
2. Manouselis, N., Drachler, H., Vuorikari, R., Hummel, H., Koper, R.: Recommender Systems in Technology Enhanced Learning. In: Recommender Systems Handbook, Part 2, pp. 387–415 (2011)
3. Dabbagh, N., Kitsantas, A.: Personal Learning Environments, Social Media, and Self-Regulated Learning: A Natural Formula for Connecting Formal and Informal Learning. *The Internet and Higher Education* 15, 3–8 (2012)
4. Tang, T.Y., Mccalla, G.: Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment. *International Journal on E-Learning* 4, 105–129, <http://www.editlib.org/p/5822> (retrieved March 2012)
5. LMS Installations 2010 at Swiss Institutions of Higher Education, <http://eduhub.ch/info/lms-installations10.html> (retrieved April 2012)
6. Drbálek, Z., Dulík, T., Koblischke, R.: Developing components for distributed search engine ObjectSpot. In: Proceedings of the 8th WSEAS International Conference on Distance Learning and Web Engineering, pp. 82–85 (2008)
7. Govaerts, S., El Helou, S., Duval, E., Gillet, D.: A Federated Search and Social Recommendation Widget. In: Proceedings of the 2nd International Workshop on Social Recommender Systems in conjunction with the 2011 ACM Conference on Computer Supported Cooperative Work, pp. 1–8 (2011)
8. Kaushik, S., Kollipalli, D.: Multi-Agent based Architecture for Querying Disjoint Data Repositories. In: International Conference on Machine and Web Intelligence, pp. 28–34 (2011)
9. Gil, A.B., De la Prieta, F., Rodríguez, S.: Automatic Learning Object Extraction and Classification in Heterogeneous Environments. In: Pérez, J.B., Corchado, J.M., Moreno, M.N., Julián, V., Mathieu, P., Canada-Bago, J., Ortega, A., Caballero, A.F. (eds.) *Highlights in Practical Applications of Agents and Multiagent Systems*. AISC, vol. 89, pp. 109–116. Springer, Heidelberg (2011)

10. Ternier, S., Verbert, K., Parra, G., Vandeputte, B., Klerkx, J., Duval, E., Ordoez, V., Ochoa, X.: The Ariadne Infrastructure for Managing and Storing Metadata. *IEEE Internet Computing* 13, 18–25 (2009)
11. Ha, K.-H., Niemann, K., Schwertel, U., Holtkamp, P., Pirkkalainen, H., Boerner, D., Kalz, M., Pitsilis, V., Vidalis, A., Pappa, D., Bick, M., Pawlowski, J., Wolpers, M.: A Novel Approach towards Skill-Based Search and Services of Open Educational Resources. In: García-Barriocanal, E., Cebeci, Z., Okur, M.C., Öztürk, A. (eds.) *MTSR 2011*. CCIS, vol. 240, pp. 312–323. Springer, Heidelberg (2011)
12. Grewe, L., Pandey, S.: Quantization of Social Data for Friend Advertisement Recommendation System. In: Nagamalai, D. (ed.) *PDCTA 2011*. CCIS, vol. 203, pp. 596–614. Springer, Heidelberg (2011)
13. El Helou, S., Li, N., Gillet, D.: The 3A interaction model: towards bridging the gap between formal and informal learning. In: *Proceedings of the Third International Conferences on Advances in Computer-Human Interactions*, pp. 179–184 (2010)
14. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the Web. Technical report. Stanford: Stanford Digital Library Technologies Project (1999)
15. El Helou, S., Salzman, C., Gillet, D.: The 3A Personalized, Contextual and Relation based Recommender System. *Journal of Universal Computer Science* 16(16), 2179–2195 (2010)
16. Shibboleth Architecture Technical Overview, <http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf> (retrieved March 2012)