Towards Full End-Users Control of Social Recommendations

Gabriela Bosetti[1](✉), Sergio Firmenich[1,2], Alejandro Fernández[1,3], Martin Wischenbart[4], Gustavo Rossi[1,2], and Damiano Distante[5]

[1] LIFIA, Facultad de Informática, UNLP, 50th St. and 120th St., La Plata, Argentina
{gabriela.bosetti,sergio.firmenich,alejandro.fernandez, gustavo}@lifia.info.unlp.edu.ar
[2] CONICET, Ciudad Autónoma de Buenos Aires (CABA), Argentina
[3] CIC, Buenos Aires, Argentina
[4] Josephinum Research, Wieselburg, Austria
martin.wischenbart@josephinum.at
[5] University of Rome Unitelma Sapienza, Viale Regina Elena, 295, 00161 Rome, Italy
damiano.distante@unitelma.it

Abstract. Recommender systems are integrated with a significant number of Web applications helping users to find what may be of their interest. However, these systems do not always satisfy the users' expectations regarding *when* and *where* recommendations are available, *what* is recommended, the *source* of the recommendations or *how* they are presented. Moreover, users may expect recommendations on sites that do not offer them. This work presents an approach for empowering users to generate a recommendation layer above the presentation layer of any existing (even third-party) Web application. The approach considers recommender systems as user-constructed artifacts, pluggable to any website and offering users full control at both stages of the recommendation process: the creation and the provision of recommendations. This paper presents preliminary work on the approach, concretely, a toolset to support our claims. Generalization tests on popular websites show no apparent restrictions regarding domain applicability.

Keywords: Recommender systems · Web augmentation
User customization · Social web

1 Motivation

Research on recommender systems (RS) has actively evolved since the mid-90s when the first articles on collaborative filtering appeared [1]. More recently, social RS [2] have begun to target social media, ultimately helping users to cope with the challenges of the social overload.

In this context, recent works also point to the importance of *controllability* in RS. Knijnenburg et al. [3] show that giving the user –who have significant domain knowledge– explicit control over the weights of the items and their friends in collaborative filtering increases the quality of recommendations and the user satisfaction. Hijikata et al. [4] observed that some factors, as the time and effort required by the user for his intervention, play an essential role in the user satisfaction. Harper et al. [5] noted a preference from users for recommendations received after they had a certain level of control. Ekstrand et al. [6] observed that a substantial portion of users choose to switch among recommendation algorithms until they find the one which satisfies them the most.

However, controllability has been studied in the context of applications that were designed with a recommendation service in mind. But there is a much more fundamental problem not directly related to the user: not all websites that could profit from having a RS are equipped with one, and many of those having one offer items from a single domain (e.g. only movies, only books) and a single source (mostly items from within the site). However, cross-domain recommendations are possible and can be beneficial for the user [7], and there are situations in which the resources of interest to the users are spread over the Web on multiple websites.

It is a fact that many sites offer recommendations and users can use such specialized RS. Nevertheless, these resources are typically isolated from other platforms and kinds of content (e.g. just videos, just slides). A potential solution comes with Web augmentation (WA) techniques [8], which allow adding new features on the top of the UI of existing (usually third-party) Web applications according to the user' s needs. Zemanta[1] is an example of this kind of application: it offers images and links from trustworthy sources as recommendations for improving blog entries. Nevertheless, like other related works, it lacks the capability of offering heterogeneous content from any Web page, especially if they are not considered a formal source (i.e., it does not provide a public API).

This article proposes an approach to put end-users in control of where, how, and what recommendations to present, as well as which content to recommend to other users. The approach rests on two pillars. First, users are empowered to (collaboratively) conceptualize arbitrary Web content into domain-specific objects employing semantic tagging. Consequently, resulting objects from various websites are extracted, collected, and become the source of recommendations. Second, any website can then be augmented with a recommendation layer supported by a configurable recommendation service. This way, any open media website can become a source of content to be recommended and have an augmented version with heterogeneous recommendations.

2 Related Work

There are several works demonstrating that users are expecting to have an active role in the process of recommendations, and that they can understand and configure certain aspects of a RS. E.g. in relation to HCI [9], or controllability [1,3,5]. However, recommendation systems are not present in every Web application,

[1] Related Content by Zemanta: https://goo.gl/dqKLzC.

even if they would be a perfect target for this kind of systems. Fortunately, customization of sites or adding a RS is possible without involvement of website providers. For instance there is a userscript[2] that augments Reddit articles with recommendations of other articles. Wischenbart et al. [10] propose empowering users to create their own augmentations for enhancing existing Web applications with personalized recommendations, but it does not contemplate elements of different nature nor from different sources. Reform [11] is a library promoting the enhancement of third-party Web applications through implementations that do not to depend on the UI of a specific website. Instead, developers define which data may fulfil the enhancement needs and end users must tell the system where in the page such information can be found as well as which page they want to augment. However, the users still depend on developers for implementing and setting up a specialized RS with the library.

Web Objects Ambient [12] allow users to create personlized Web experiences by extracting and materializing existing Web content, annotating it with classes of a particular ontology and finally wrapping it with some specialized behavior. Although this allows to generate a common space of information with the collected objects, such information items are not considered as content to be recommended to a group of people.

WikiLens [13] allows creating a community-maintained space of recommendations. Users can introduce new items from any website into the recommender and they can manually add new attributes to it. Other users can rate those items and find information of their interest, but the approach presents recommendations in a different context than where the user would need such information.

3 The Approach in a Nutshell

Universal Web design principles are used to structure Web pages for clearly presenting a group of related information items [14]. Web pages presenting similar information usually share the same set of properties and UI widgets, and Firmenich et al. [12] pointed out that users can recognize and define a conceptualization layer over the presentation of any Web page.

In this work, users identify which elements in any DOM should be considered as content to recommend to others. Web Scraping and Web Augmentation are presented as a means for enabling multi-origin recommendations on the top of any third-party Web site. It is about expanding the limits of technology but also about enhancing the user experience in any Web page by the addition of social recommendations. For that purpose, users should be able to:

1. Indicate how to interpret the existing Web content (e.g. a video in Youtube), s.t. other elements sharing the same structure can be automatically extracted.
2. Collect objects of their interest from the Web pages in a shared ecosystem. These objects (e.g. videos, slides) and their available properties (e.g. title, duration) will be stored in a shared repository while users browse these sites.

[2] XPLR Reddit Recommender: https://github.com/xplr/xplr_reddit_recommend.

3. Configure recommenders and use their content to improve any website. These recommenders will offer recommendations from the universe of objects in the shared repository, depending on the available information and the configuration of the widgets (e.g., type of content, sources).

Figure 1 depicts the fundamental architectural elements of the platform supporting the approach, called CoRA (Collaboration for Recommendations in Augmented Web applications). The toolset is composed of three client-side applications –browser extensions– that interact with a recommendation server. The following sections explain the functionality and interplay of these components in detail.

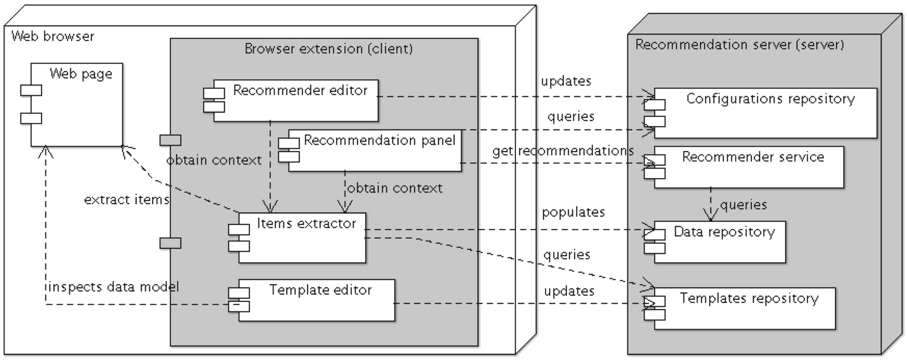
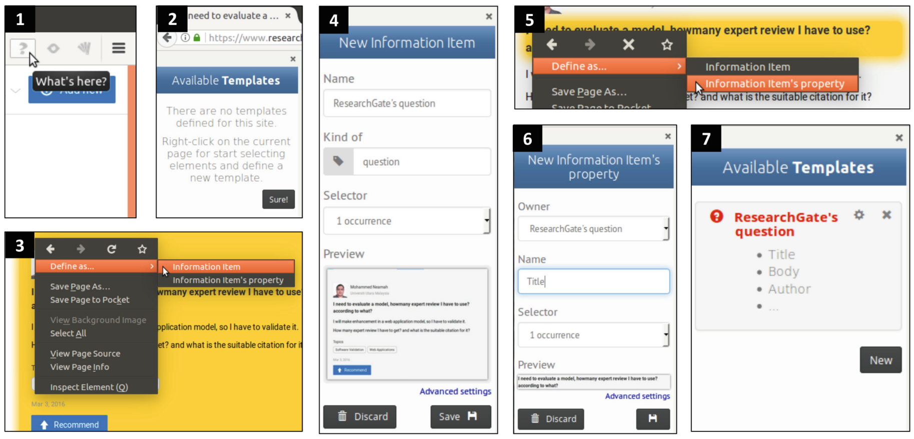


Fig. 1. Architecture of CoRA

3.1 Information Extraction

As a means to express a family of similarly structured Web content under a common URL pattern we propose to use *extraction templates*. Such templates are used to extract *information items* from diverse Web pages, and to push them to a shared data repository. A *template editor* offers the user a visual mechanism to select portions of any DOM and to define reusable extraction templates, which are then stored in another shared repository. Whenever a user visits a Web page, the information *items extractor* checks whether any of the available templates in the repository applies to the visited page and asks the user whether he wants to extract and submit the available items in the page to the data repository.

Extraction templates are modeled using a combination of JSON and XPath, but the editor hides this complexity from the user. The tool is materialized as a button in the browser's UI, shown in step (1) of Figure 2. To inspect or create extraction templates, the user opens the template editor via the *items* toolbar button (1). As response, the tool lists the templates from the repository that match the page's URL (7). If there is no matching template, a message is presented instead (2). The user can start creating a new template by selecting an



There are no templates defined for this site.
Right-click on the current page for start selecting elements and define a new template.

many expert review I have to use?

pplication model, so I have to validate it.

et? and what is the suitable citation for it?

Fig. 2. Defining a template with CoRA (Color figure online)

information item from the DOM (3) and filling the required data for defining it (4). Templates are labeled by a user-provided name and the type of information that it extracts. Structuring an information item also requires properties to be defined in a similar way (5, 6). The properties are stored as an association between the property name and the XPath matching the proper DOM element. Once there is a public template matching a URL (7), it is presented to any user clicking the «templates» button (1). The user can create several templates for a single page to extract different types of information items.

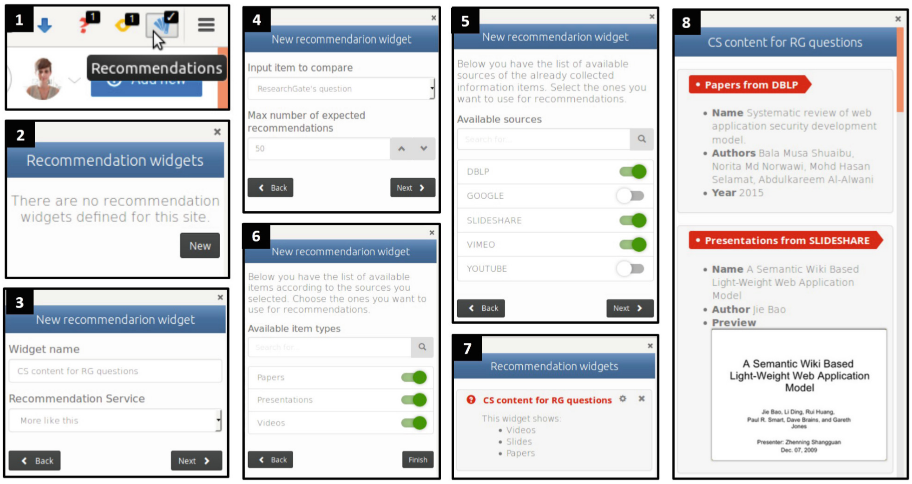
On each Web page change, the *items extractor* loads all templates with URL patterns matching the current URL from the repository. A red label on the toolbar button serves as visual feedback for the result of the extraction process on the current page. For each matching template, the tool checks that all the referenced DOM elements exist and extracts the items with all their properties. If the user agrees to save the item, the item is uploaded to the repository.

3.2 Recommendations

Once a user has configured a reusable *recommender service* they can be turned into *recommendation widgets* that can be attached to specific Web pages. Back to Fig. 1, the *recommender editor* is an end-user tool for attaching and configuring one or many recommendation widgets for a given website matching a specific URL pattern. A widget is configured for a concrete recommendation algorithm– one of the extension points of our architecture, which may require different parameters to work properly. By default, we provide a simple content-based recommender service based on TF-IDF. Information items are interpreted as documents by concatenating their properties into a single text. As multiple items can be extracted from the same page, a recommender may ask the user to choose an item to compare against stored ones at server-side.

The *recommendation panel* is in charge of querying the *configurations repository* on every Web page change, to load all the matching configurations at client-side. Each configuration queries the *recommender service* to start presenting recommendations to the user.

Recommendations are finally presented to the user in a sidebar panel that can be opened/closed by clicking on the «recommendations» button, as shown in step (1) of Figure 3. Recommendations are presented by the widgets (7). However, if there is no widget defined yet (2), the user can start creating a new one (3). Then, he can configure the specific parameters of such a service (4, 5, 6) and the template will be available in the widgets' view. When he clicks the widget item (7) he accesses the recommendations as shown in the last step (8).



Recommendation widgets

There are no recommendation widgets defined for this site.

New

New recommendarion widget

Widget name: CS content for RG questions

Recommendation Service: More like this

Back Next

New recommendarion widget

Input item to compare: ResearchGate's question

Max number of expected recommendations: 50

Back Next

New recommendarion widget

Below you have the list of available sources of the already collected information items. Select the ones you want to use for recommendations.

Available sources

DBLP ✓
GOOGLE
SLIDESHARE ✓
VIMEO ✓
YOUTUBE

Back Next

New recommendarion widget

Below you have the list of available items according to the sources you selected. Choose the ones you want to use for recommendations.

Available item types

Papers ✓
Presentations ✓
Videos ✓

Back Finish

Recommendation widgets

CS content for RG questions

This widget shows:
- Videos
- Slides
- Papers

CS content for RG questions

Papers from DBLP
- Name Systematic review of web application security development model.
- Authors Bala Musa Shuaibu, Norita Md Norwawi, Mohd Hasan Selamat, Abdulkareem Al-Alwani
- Year 2015

Presentations from SLIDESHARE
- Name A Semantic Wiki Based Light-Weight Web Application Model
- Author Jie Bao
- Preview

A Semantic Wiki Based Light-Weight Web Application Model

Jie Bao, Li Ding, Rui Huang, Paul R. Smart, Dave Brains, and Gareth Jones

Presenter: Zhenning Shangguan
Dec. 07, 2009

Fig. 3. Recommending related items collected by other users

4 Validation

This preliminary work assesses the feasibility of the approach through validation by construction and in applicability tests for a selection of popular web-sites. The tests focused especially on the robustness of the item extraction approach. Having a good strategy for extracting information items by reusing a template definition is crucial, since generating wrong XPaths or choosing properties not present in all the documents may result in an exception when trying to extract information items at the harvesting stage.

Templates and recommender widgets were defined for a subset of representative websites to assess generalization. The list of test sites comprised the first three sites with open and free-of-charge content as reported by Alexa in seven categories. A single template was created for each site, matching a single information item with standard properties for the templates in the same category.

The «Video Sharing» category comprised YouTube, Vimeo, and Dailymotion. The extracted conceptual item was a video with the following properties: title, author, publication date, description, views and the video itself. The «Presentation Sharing» category contained 12 sites, of which only two met the requirement of open and free access: SlideShare and Authorstream. There templates were defined to extract slides with: title, author, presentation, likes and description. The «Ask an expert» category included StackOverflow, Yahoo! Answers and Answers.com. Their templates specify how to extract questions with two common properties: the question itself and their categories.

The e-book «Titles» category presented the Gutenberg, Bookboon, and ManyBooks.net as the top sites. But, there was just a single shared property: the title. Gutenberg also contemplated preview, author, and downloads; Bookboon also contemplated the preview and description; ManyBooks the author and description. Sciencedirect, Nature, and Arxiv were the first sites in «Science/Publications». Templates for Article items were defined with the following properties: title, authors, abstract, publication date, DOI, journal, and keywords. In the case of Nature, it was necessary to define different templates for the articles of each Journal, since although its search engine integrates its data, their presentation changes and the selectors are not applicable to every article on the site. The same happens with each blog in the «Weblog» category: WordPress, Tumblr, and Blogger. Their templates vary according to different sites. The extracted items represented blogs with: title, body, publication date, picture, and related tags.

No technical issues were registered in the visual definition of the templates for any of the mentioned sites. Recommender widgets were configured for the same websites, but these are generalizable by design; they operate in an independent floating panel, not being affected by the Web page design.

5 Conclusions and Future Work

This article presents an approach for allowing end users to populate a shared repository of user-collected data over the Web for later retrieving recommendations. This way, users can create an environment of shared information that helps them to achieve a common goal. This work is preliminary; a formal experiment is being currently designed to assess its usability and potential of adoption.

Future work will focus on automatically completing information items' data when something is missing on a Web page. Recommendation algorithms will be extended and made available as configuration options for the widgets. Finally, further work is required to enable collaborative editing of existing templates and to enable recommendations from different groups or repositories.

References

1. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press, New York (2010)
2. Guy, I.: Social recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 511–543. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_15
3. Knijnenburg, B.P., Bostandjiev, S., Donovan, J.O., Kobsa, A.: Inspectability and control in social recommenders. In: Proceedings of the Sixth ACM Conference on Recommender Systems, pp. 43–50 (2012)
4. Hijikata, Y., Kai, Y., Nishida, S.: A study of user intervention and user satisfaction in recommender systems. MIS Q. **22**(4), 669–678 (2014)
5. Harper, F.M., Xu, F., Kaur, H., Condiff, K., Chang, S., Terveen, L.: Putting users in control of their recommendations. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 3–10. ACM, New York (2015)
6. Ekstrand, M.D., Kluver, D., Harper, F.M., Konstan, J.A.: Letting users choose recommender algorithms: an experimental study. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 11–18. ACM, New York (2015)
7. Cantador, I., Fernández-Tobías, I., Berkovsky, S., Cremonesi, P.: Cross-domain recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 919–959. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_27
8. Díaz, O., Arellano, C., Azanza, M.: A language for end-user web augmentation: caring for producers and consumers alike. ACM TWEB **7**(2) (2013). https://dl.acm.org/citation.cfm?id=2460388. Article No. 9
9. Verbert, K., Parra, D., Brusilovsky, P., Duval, E.: Visualizing recommendations to support exploration, transparency and controllability. In: Proceedings of the 2013 International Conference on Intelligent User Interfaces, pp. 351–362. ACM (2013)
10. Wischenbart, M., Firmenich, S., Rossi, G., Wimmer, M.: Recommender systems for the people - enhancing personalization in web augmentation. In: Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, pp. 53–60 (2015)
11. Toomim, M., Drucker, S.M., Dontcheva, M., Rahimi, A., Thomson, B., Landay, J.A.: Attaching UI enhancements to websites with end users. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1859–1868. ACM, New York (2009)
12. Firmenich, S., Bosetti, G., Rossi, G., Winckler, M., Barbieri, T.: Abstracting and structuring web contents for supporting personal web experiences. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 77–95. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_5
13. Frankowski, D., Lam, S.K., Sen, S., Harper, F.M., Yilek, S., Cassano, M., Riedl, J.: Recommenders everywhere: the wikilens community-maintained recommender system. In: Proceedings of the 2007 International Symposium on Wikis, pp. 47–60. ACM, New York (2007)
14. Sharp, H., Rogers, Y., Preece, J.: Interaction Design: Beyond Human Computer Interaction. Wiley, Chichester (2007)