Private Cross-page Movie Recommendations with the Firefox add-on OMORE

Tobias Bannwart, Amancio Bouza, Gerald Reif, and Abraham Bernstein

Department of Informatics, University of Zurich
Binzmuehlestrasse 14, 8050 Zurich, Switzerland
{tobias.bannwart@alumni,bouza@ifi,reif@ifi,bernstein@ifi}.uzh.ch

Abstract. Online stores and Web portals bring information about a myriad of items such as books, CDs, restaurants or movies at the user's fingertips. Although, the Web reduces the barrier to the information, the user is overwhelmed by the number of available items. Therefore, recommender systems aim to guide the user to relevant items. Current recommender systems store user ratings on the server side. This way the scope of the recommendations is limited to this server only. In addition, the user entrusts the operator of the server with valuable information about his preferences. In this paper we introduce the private, personal movie recommender OMORE, which learns the user model based on the user's movie ratings. To preserve privacy, OMORE is implemented as Firefox add-on which stores the user ratings and the learned user model locally at the client side. Although OMORE uses the features from the movie pages on the IMDb site, it is not restricted to IMDb only. To enable cross-referencing between various movie sites such as IMDb, Amazon.com, Blockbuster, Netflix, Jinni, or Rotten Tomatoes we introduce the movie cross-reference database LiMo which contributes to the Linked Data cloud.

1 Introduction

On the Web, a myriad of items is in competition for the users attention. This items can either be products in an online store (e.g. movies, music CDs and books) or events or places to visit such as parties, restaurants or bars. Since the users are overwhelmed by the amount of possible choices, some online stores and portal sites use recommender systems to help the user to find items that match their personal interest. Users are typically asked to explicitly rate items. Data mining techniques are then applied on these ratings to learn either a user or item model to predict a rating for an unknown item. Or, in a more simplified system, the user ratings are only used to display the average user rating across all users applying the Wisdom of Crowds phenomenon [6].

These systems face two main problems. First, the user has to trust the owner of the portal, that his data is protected from unauthorized access and it is not misused and shared with third parties. For example, the kind of books or movies a user considers interesting, might give a hint on the political or sexual orientation

of the user. Second, ratings the user did in one online store or portal are stored on server side and cannot be reused on other Web applications. Therefore, users have to rate the same items over and over again on each Web site they want to take advantage from recommendations.

In this paper we introduce OMORE, which provides a solution to these problems. OMORE is a personal movie recommender, that runs locally as Firefox[1] add-on and let the user rate movies, when he browses a supported Web page. The user ratings are stored locally which tackles the privacy issue. The ratings are used to learn the user model which adapts the movie preferences of a user. In the configuration plane of the add-on the user can choose among different learning algorithms. The user model is then used to generated personal recommendations which are directly visible within the supported Web sites the user is currently browsing.

The current implementation supports cross-page recommendations for the following movie pages: Amazon.com[2], IMDb[3], Netflix[4], Rotten Tomatoes[5], Blockbuster[6], DBpedia[7], LinkedMDB[8] and Jinni[9]. To enable the cross-referencing between these movie sites we defined a movie ontology and an URI to uniquely identify movies across different movie sites that is key to provide cross-page recommendations. We published the movie URIs and the references of the supported movie sites in a database following the Linked Data guidelines [1].

2 IMDb as Data source and Movie Cross-Reference Database

On the Web we find many Web sites that provide information about movies. To help the user to find an interesting movie, some of these sites also provide a rating for each movie. This rating can either be the average rating over all user ratings, or/and a personalized rating for each user that is based on the movies the user has rated so far. As discussed in the introduction, these ratings are stored on server side, which causes privacy issues, and hinders the reuse across Web sites. Therefore, OMORE stores user ratings and the user model on client side. In this section, we introduce the data OMORE operates on.

In OMORE we apply machine learning algorithms to learn the model which represents the user's movie preferences. This model is further used to predict the rating if the movie is not rated yet. These algorithms require a feature vector representation of a movie where each dimension of the vector represents

[1] `http://www.mozilla.org/firefox/`
[2] `http://www.amazon.com`
[3] `http://www.imdb.com`
[4] `http://www.netflix.com`
[5] `http://www.rottentomatoes.com`
[6] `http://www.blockbuster.com`
[7] `http://www.dbpedia.org`
[8] `http://www.linkedmdb.org`
[9] `http://www.jinni.com`

a particular feature. Therefore, OMORE requires a comprehensive movie data base which provides detailed descriptions of each movie. Additionally, a cross-page movie recommender system also requires cross-references between the same movie across different Web sites. A first attempt exists in form of the Linked Movie Database (LinkedMDB) [2], but the available data is not as comprehensive as required for OMORE. Thus, we came up with another approach to establish the link between identical movies across Web sites. In the following, we explain our collaborative approach to establish the link between the same movies across Web sites and describe the data source we use to build the feature vector.

2.1 IMDb as Source for Features

Machine learning algorithms are applied to learn the user's individual movie preferences. Because such algorithms demand a feature vector representation of movies, we need (1) an exhaustive data source of almost all movies and that (2) provides adequate features for the machine learning task. The structured information as well as the content information about a movie differs among different Web pages. This heterogeneous data makes it impossible to extract a meaningful feature vector representation of a movie. Therefore, we have to relay at least on one good movie data source to build feature vectors from. On the one hand, the LinkedMDB provides semantic movie informations about 38k movies. The LinkedMDB provides the data as semantic data and interlinks the movies and their details to the Linked Open Data Cloud and other movie Web stites such as Internet Movie Database (IMDb) or Rotten Tomatoe. On the other hand, the IMDb provides movie information about over 1.5 million movies with structured and detailed feature information, which makes the IMDb the most "comprehensive" movie archive available on the Web. We decided to use the IMDb as data source because we wanted to ensure that OMORE is able to provide accurate recommendations for a wide range of movies. The IMDb is used as data source, even if the user surfs a different supported movie Web site. Therefore, we have to establish a link from each Web site that is supported by OMORE to the IMDb site. How these links are established is discussed in Sect. 2.2.

Unfortunately, IMDb does not provide an ontology-based version of their database or direct database access to retrieve the features of a specific movie. In addition, for copyright reasons we were not able to transform the IMDb data into RDF and publish it on the Web. Therefore, we wrote a wrapper that extracts the features from the HTML version of an IMDb movie detail page. Not all features are available for each movie. To avoid computational complexity and the problem of missing values that bias the preference learning, we considered features that are provided by most movies and that adequately describe the movie's content. The list of features used for the feature vector is presented in Table 1.

Feature	Description
Genre	Categorization of a movie; each movie can have more than one genre assigned (e.g., Action, Fantasy, Animation, etc.)
Country	Country where the movie has been released; each movie can have more than one country assigned
IMDb Rating	Average rating of the IMDb users that indicates the overall quality of the movie
Movie runtime	length of the movie in minutes
Release date	Year when the movie has been released
Color	Color the movie is presented such as colorized or black and white

Table 1. Attributes used by OMORE

2.2 Interlinking Movies across Web Pages

We use a collaborative approach to establish the link between identical movies across Web sites and store these cross-references in our LiMo database[10]. Our LiMo database is based on D2R [5]. The LiMo database provides an unique URI for each movie, the URL to the IMDb movie page, the movie title, the release year, and the cross-references to all movie sites that are supported by OMORE. OMORE uses the URL of the currently viewed Web page as key to look up the IMDb URL in the LiMo database. As discussed above, the IMDb URL is used to build the feature vector of a movie.

When a user browses a movie Web page where the cross-reference cannot be found in our LiMo database, OMORE uses our movie lookup service MOLookup[11] to retrieve the IMDb URL. OMORE uses the movie title, release year, and the URL of the Web page for the request to MOLookup. This information is extracted by OMORE from the currently viewed Web page. The title and the year are used for the lookup. The URL is used to store the new cross-reference in the LiMo database. This way the LiMo database "learns" about the existence of a new movie page and stores the cross-reference. The user benefits from the movie recommendation and we gain a comprehensive set of movie cross-references. This approach seems to be promissing since it follows the Pareto improvement principle [4].

Establishing the links among movies from different Web pages is challenging due to the heterogeneous presentation of movies and the fact that Web pages may misspell, transform or extend movie titles in various ways. Especially on online shops, we experienced that the movie titles are extended with information about many variants of special or collector's edition and the type of medium the movie is provided. Instead of trying to extract the original title from the unpurified title, we decided to apply fuzzy search over movie titles and release year to retrieve the linkage among identical movies from different Web pages.

[10] `http://seal.ifi.uzh.ch/limo`
[11] `http://seal.ifi.uzh.ch/molookup/`

For this purpose, MOLookup uses the fuzzy search facility of Apache Lucene[12]. Lucene is a scalable and popular open-source search software that enables fast search within textual data. Lucine uses the Levensthein distance to compute the similarity between the given title and the titles stored in the LiMo database. In addition to the JSON interface which used by OMORE, MOLookup also provides a Web interface (`http://seal.ifi.uzh.ch/molookup/`).

As an initial dataset for the LiMo database, we added all movies from the IMDb dataset and the movie titles, release years, and IMDb URLs (about 1.4 million movies). In addition we have the cross-references to the Rotten Tomatoes site and to about 12.000 Amazon.com pages. The cross-references to the other supported movie pages (Netflix, Jinni, Blockbuster, LinkedMDB) will be established as discussed above, when OMORE is used by the community.

3 The Architecture

In this section we introduce the architecture of the OMORE Firefox add-on that is presented in Figure 1. The central part of OMORE is the machine-learning component. It consists of the *Learner*, which is responsible for learning the user model and the *Classifier* for generating recommendations based on the user model [3]. The user can choose between several machine learning algorithms from the WEKA library [7].

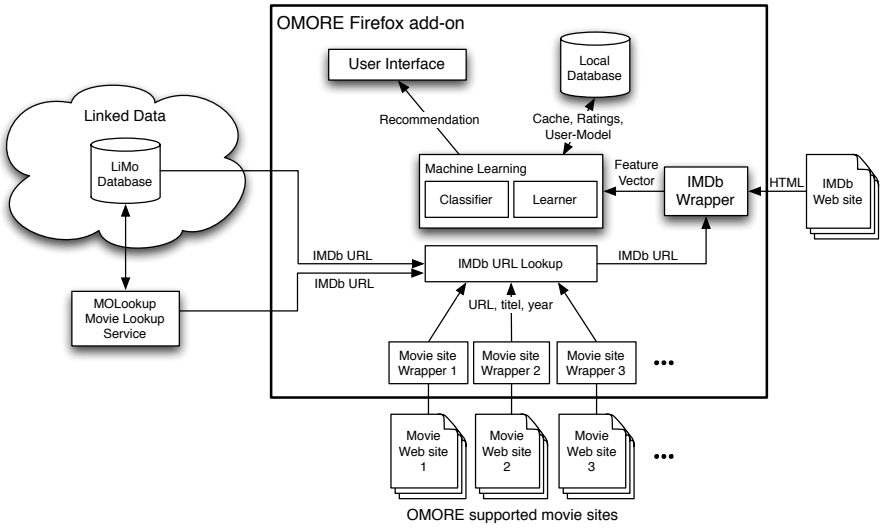

Fig. 1. Overview of the OMORE Architecture

[12] `http://lucene.apache.org`

When a user browses a Web page the URL is checked by the OMORE Firefox add-on if the current URL is a movie page that is supported by OMORE. This is done by analyzing the URL of the Web page. The URL also indicates which *Movie Site Wrapper* should be used. The wrapper extracts the URL, the title, and release year of the movie. This information is then used to retrieve the IMDb URL from the LiMo database or the MOLookup service (see section above). The URL is then handed over to the *IMDb Wrapper* to build the feature vector.

When a user rates a movie the feature vector and the rating are stored in the local SQLite database that comes with Firefox. When a user has rated enough movies (threshold can be set in the configuration) the *Learner* retrieves the feature vectors and the user ratings from the *Local Database* and starts the learning of the user model. Generally, the learning of a model is computationally expensive in contrast to the use of the model. Therefore, we store the learned user model locally and reuse it for the generation of new recommendations. This way, OMORE scales over the movies rated by a user. The user model is recomputed from in regular intervals to improve the adaptation to the user's preferences. The feature vector is also used by the *Classifier* to generate the recommendation which is displayed to the user.

The wrapper is not only responsible for extracting information from the HTML Web page, but also for displaying the OMORE user interface. The user interface is integrated in the movie page by manipulating the DOM tree of the page. It displays the rating and gives the possibility to rate the movie. A screenshot is shown in Fig. 2. OMORE easily be extended to support additional movie pages by simply adding a new Movie Site Wrapper.

4 OMORE in Action

This section describes the installation and usage of the private, personal movie recommender system OMORE and how the user benefits from rated movies on different Web sites.

When a user browses a movie Web page, the OMORE add-on is triggered and renders the user interface directly within the movie Web page. Figure 2 shows how OMORE's user interface is nicely integrated within two different Web pages from different providers even imitating their look and feel. The user interface is highlighted with a red border. The user interface is divided into two vertical sections providing two rating bars. The left rating bar indicated with "User Rating" enables the user to rate the current movie even if the provider does not provide such a facility. The rating is automatically stored locally without further effort. If the user already rated the particular movie already, his rating is rendered in the left rating bar.

The rating bar on the right side indicated with "OMORE Recommendation", renders the recommendation that is computed by OMORE. In case that the user has not yet rated enough movies to learn a fair user model, the right rating bar is replaced with an invitation asking the user to rate more movies. By aligning

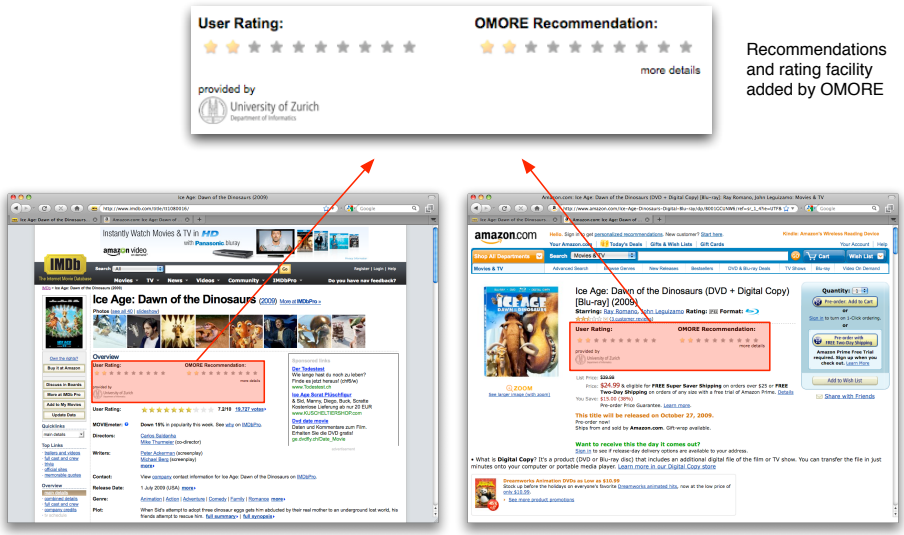

Fig. 2. Supported movie site with the custom OMORE element

the two rating bars in a horizontal plane, we enable the user to verify how well the recommendation generated by OMORE tallies with his personal rating.

For anyone interested in trying out our Mozilla Firefox add-on, we provide a binary release on our Web page at `http://seal.ifi.uzh.ch/omore`. It can be installed with a single click.

5 Conclusion

In this paper we introduced the private, personal movie recommender OMORE for accurate cross-page recommendations. OMORE is able to autonomously adapt to evolving user preferences and is able to improve the accuracy of provided recommendations with every additional movie the user rates. Additionally, we contribute to the Linked Data Open cloud with our LiMo database that provides cross-references between identical movies among different Web sites. In addition, we provide the movie lookup service MOLookup that contributes cross-references to the Linked Data cloud with every new movie page a user visits.

The key distinguishing feature of OMORE, compared to recommender systems found on most online stores and Web portals, is on the one hand, the storage of the ratings and user model on the client side and on the other hand, cross-page recommendations that liberate people from lock-in situation and barriers. This way OMORE preserves the privacy of sensitive user rating data and enables the reuse of ratings across the borders of different Web sites. The current implementation of OMORE supports the most common movie portals and online stores.

Unfortunately, none of those Web sites provide ontology-based access to their movie data. To over come the heterogeneous content from different sources, we are using wrappers to extract the needed information from the HTML Web pages and utilize the URL, the title and release year for the movie retrieval with the LiMo database and MOLookup.

Since IMDb is run by a commercial company and the terms and conditions[13] explicitly disallows the republishing of their data, we cannot build an ontology based version of IMDb and contribute it to the Linked Data cloud[14]. This way we would have a unique identifier for each movie and could build the feature vector with a single SPARQL query instead of using HTML wrapper. Once, the LinkedMDB provides information about roughly the same amount as IMDb, we may use the LinkedMDB as main source for building feature vectors. This way OMORE could present additional movie data that is provided by the Linked Data cloud. Thus, the user directly benefits from the Semantic Web and the rich semantical content provided.

References

1. Linked data - connect distributed data across the web. Homepage, Last visited Septermber 2009. `http://linkeddata.org/`.
2. Linked movie database. Homepage, Last visited Septermber 2009. `http://www.linkedmdb.org/`.
3. C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *In Proceedings of the 15th National Conference on Artificial Intelligence*, pages 714–720. AAAI Press, 1998.
4. B. S. Bernanke and R. H. Frank. *Principles of Micro Economics*. McGraq-Hill/Irwin, 2007.
5. C. Bizer and R. Cyganiak. D2r server - publishing releational databases on the semantic web. In *5th International Semantic Web Conference.*, 2006.
6. J. Surowiecki. *The Wisdom of Crowds*. Doubleday, 2004.
7. I. H. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques*. Second edition. Elsevier, San Francisco, 2nd edition edition, 2005.

[13] `http://www.imdb.com/help/show_leaf?usedatasoftware`
[14] `http://linkeddata.org/`