X-REC: Cross-Category Entity Recommendation

Dragan Milchevski
Max Planck Institute for Informatics
Saarbrücken, Germany
dmilchev@mpi-inf.mpg.de

Klaus Berberich
Max Planck Institute for Informatics
Saarbrücken, Germany
kberberi@mpi-inf.mpg.de

ABSTRACT
We demonstrate X-REC, a novel system for entity recommendation. In contrast to other systems, X-REC can recommend entities from diverse categories including goods (e.g., books), other physical entities (e.g., actors), but also immaterial entities (e.g., ideologies). Further, it does so only based on publicly available data sources, including the revision history of Wikipedia, using an easily extensible approach for recommending entities. We describe X-REC’s architecture, showing how its components interact with each other. Moreover, we outline our demonstration, which foresees different modes for users to interact with the system.

Categories and Subject Descriptors
H.3.3 [Information Search & Retrieval]: Information filtering

Keywords
Entity Recommendation, Open Data, Wikipedia

1. INTRODUCTION
Recommender systems are commonplace today and at the heart of many businesses. In fact, the bottom line of companies like Amazon or Netflix crucially depends on their ability to recommend goods (e.g., books or movies) to customers. Recently, in their move toward showing more structured results to users, web search engines also have started to recommend entities to users. Google’s Knowledge Graph, as a concrete example, recommends entities similar to the query entity, arguably based on query logs. Likewise, social networks like Facebook suggest users and pages based on what users liked in the past.

Systems like the aforementioned suffer from at least two shortcomings. First, they are typically restricted to a narrow set of entity categories (e.g., buyable goods) and do not exploit correlations across diverse entity categories – for instance, they can usually not recommend actors based on books or vice versa. Second, they are commonly based on proprietary data sources (e.g., purchase histories or query logs), making the process of how recommendations are generated non-transparent to users.

The X-REC system that we propose in this work takes a different approach and differs in the following aspects:

- it recommends a wide variety of entities from the YAGO2 knowledge base [6] including goods (e.g., books and movies), other physical entities (e.g., actors and cities), but also immaterial entities (e.g., ideologies or brands);
- it exploits correlations and recommends entities across diverse categories, allowing it to recommend, for instance, brands based on movies;
- it uses only publicly available data sources, including the revision history of Wikipedia and the YAGO2 knowledge base, combining them in an easily extensible approach to recommend entities.

More precisely, X-REC uses three publicly-available data sources to generate entity recommendations. The co-editing behavior of Wikipedia articles, as reflected in their revision history, unveils subtle connections between the corresponding entities. This allows us to connect, for instance, Dave Grohl and Californication (TV series) based on the fact that there is substantial overlap in the Wikipedians who edited the corresponding articles. Likewise, by comparing the contents of Wikipedia articles, X-REC can relate entities having similar descriptions, for instance, Freddie Mercury and Love of My Life (a song by Queen (band)). Finally, by comparing their sets of Wikipedia categories and WordNet classes contained in the YAGO2 knowledge base, X-REC can relate entities in terms of their semantics – for instance, Eric Clapton and Pete Townsend (both among other commonalities– English rock guitarists).

Entity recommendations are generated based on a simple yet easily extensible approach. Each of the above data sources is incorporated via an entity similarity function. X-REC then recommends entities having high combined similarity to entities highly rated by the user. By using multiple data sources, which complement each other, X-REC is robust and effective, i.e., able to recommend good entities to users even when they have rated only few or exotic entities.

Organization. Section 2 provides details on our entity recommendation approach and how we make use of the different data sources. We describe the architecture of our prototype implementation in Section 3. Our demonstration is outlined in Section 4. We discuss relations between X-REC and prior work in Section 5 and summarize in Section 6.
2. RECOMMENDING ENTITIES

We now describe X-Rec’s approach to recommend entities based on publicly available data sources.

Model. Let $\mathcal{U}$ denote the set of users, $\mathcal{E}$ be the set of entities, and $\mathcal{C}$ denote the set of categories. We further assume a partial function $ra : \mathcal{U} \times \mathcal{E} \not\rightarrow [0, m]$, so that $ra(u, e)$ reflects the rating given by user $u$ to entity $e$, and use $\text{dom}(ra) \subseteq \mathcal{U} \times \mathcal{E}$ to refer to the set of user-entity pairs for which ratings are known.

User Similarity. As motivated above, the Wikipedia revision history informs us about connections between entities from potentially very different categories or textual descriptions (e.g., Black Sabbath and Gibson SG). Let $rc(u, e)$ denote the number of revisions that user $u$ has contributed to the Wikipedia article corresponding to entity $e$. We measure the similarity between two entities as

$$sim_u(e, e') = \frac{\sum_{u \in \mathcal{U}} rc(u, e) \cdot rc(u, e')}{\sqrt{\sum_{u \in \mathcal{U}} rc(u, e)^2} \cdot \sqrt{\sum_{u \in \mathcal{U}} rc(u, e')^2}},$$

which captures the correlation between users’ tendency to contribute to the Wikipedia articles corresponding to entities $e$ and $e'$, respectively. To eliminate noise from the Wikipedia revision history, we apply a number of pre-processing steps. Thus, we ignore all revisions likely to stem from bot users, which can be identified based on the user name. Also, we filter out revisions that only reverted acts of vandalism, which can be identified from the accompanying comment. Our user similarity ranges in $[0, 1]$.

Content Similarity. The contents of Wikipedia articles allow us to relate entities with similar textual descriptions (e.g., Lord of the Rings and Gandalf). To compare them, we use the cosine similarity between their corresponding $tfidf$ vectors, formally

$$sim_c(e, e') = \frac{\sum_{v \in \mathcal{V}} w_{tfidf}(e, v) \cdot w_{tfidf}(e', v)}{\sqrt{\sum_{v \in \mathcal{V}} w_{tfidf}(e, v)^2} \cdot \sqrt{\sum_{v \in \mathcal{V}} w_{tfidf}(e', v)^2}},$$

with $\mathcal{V}$ as the vocabulary of words and $w_{tfidf}(e, v)$ as the $tfidf$ weight of word $v$ in the Wikipedia article corresponding to entity $e$. Our implementation uses a length-normalized $tf$-weight and ignores terms having a negative $idf$-weight, which are quasi stopwords, ensuring that our measure of content similarity ranges in $[0, 1]$.

Semantic Similarity. The YAGO2 knowledge base integrates Wikipedia categories (e.g., American rock guitarists) and classes from Wordnet (e.g., guitarist) in a common taxonomy. It allows us to relate two entities in terms of their semantics (e.g., Lord of the Rings and His Dark Materials). Let $ca(e)$ denote the set of Wikipedia categories to which the entity $e$ belongs. We use the average Wu & Palmer [8] similarity

$$sim_s(e, e') = \frac{1}{|ca(e)||ca(e')|} \sum_{c \in ca(e), c' \in ca(e')} \frac{2d(lca(c, c'))}{d(c) + d(c')}$$

as a measure of semantic similarity between entities. Here, $lca(c, c')$ is the lowest common ancestor of categories $c$ and $c'$ in the taxonomy, and $d(c)$ denotes the depth of category $c$ with the root category owl:Thing having depth 1. We thus assign maximal similarity 1 to two entities only if they belong to the same set of Wikipedia categories. Otherwise, our measure of semantic similarity reflects how far one has to generalize their categories on average to arrive at a common supercategory, yielding a value in $[0, 1]$.

Generating Recommendations. We combine the entity similarity functions described above as

$$sim(e, e') = w_u sim_u(e, e') + w_c sim_c(e, e') + w_s sim_s(e, e')$$

with weights $w_u, w_c$, and $w_s$ learned from training data. Note that this formulation makes it straightforward to incorporate other publicly available datasets in our approach — via additional similarity functions. Entities are then recommended according to the score

$$s(u, e') = \sum_{(u, e') \in \text{dom}(v)} ra(u, e) \cdot sim(e, e').$$

Our approach thus favors entities that are highly similar to entities rated highly by the user. In practice, the different similarity measures complement each other, increasing the robustness of our approach. As a concrete example, Table 1 shows the entities most similar to Queen (band) under each of our three similarity measures. Here, user similarity brings up a person, an album, and a song related to the band; content similarity yields albums including two by other artists; semantic similarity ranks other artists at the top.

<table>
<thead>
<tr>
<th>User Similarity</th>
<th>Content Similarity</th>
<th>Semantic Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Deacon</td>
<td>Live at Wembley '86</td>
<td>Duran Duran</td>
</tr>
<tr>
<td>A Night at the Opera</td>
<td>The Happy Club</td>
<td>Etta James</td>
</tr>
<tr>
<td>Too Much Love Will Kill You</td>
<td>Anthems (Kerry Ellis album)</td>
<td>Nick Drake</td>
</tr>
</tbody>
</table>

Table 1: Top-3 entities most similar to the entity Queen (band) according to our three similarity measures

3. ARCHITECTURE

We now describe the architecture of our prototype implementation, which is depicted in Figure 2.

User Interface. Figure 1 shows a screenshot of X-Rec’s recommendation screen. On the left-hand side, the system displays entities for which the user has already provided ratings on a scale from 1 to 8. On the right-hand side, the system displays its entity recommendations in descending order of their determined score. Users can restrict the categories from which entities are recommended. Currently, X-Rec offers nine different domains Movies, Books, Music, Persons, Organizations, Locations, Events, Artifacts, and Misc, each corresponding to a set of Wikipedia categories and WordNet classes from YAGO2. Recommended entities are shown...
with an image and an abstract shown on image hover, and users can provide a rating for them and navigate to their Wikipedia article. When rating entities from scratch, as shown in Figure 3, users are supported by auto completion and can directly choose the desired preference value. The user interface has been implemented using the Bootstrap framework\(^1\). Auto completion, management of users’ entity ratings, etc. have been implemented in Java using the Play framework and communicate directly with the data storage backend (4).

**Recommender.** X-REC’s recommendation approach has been implemented in Java and wrapped as a RESTful web service running on Apache Tomcat. The user interface thus communicates with the recommender component via HTTP (3). To determine recommendations for a specific user, the recommender component keeps the user-item ratings in main memory for performance reasons, and communicates with the data storage back-end to determine and aggregate the different entity similarities (2).

**Data Storage.** Some of the data including users’ entity ratings, user-article revision counts, and user management are kept in a PostgreSQL relational database. All entity metadata, including the entity-category relations and the \( tf.idf \) vectors of Wikipedia articles are kept in MongoDB for fast querying. The Wikipedia revision history, including the most recent version of every article’s content, and YAGO2 are imported directly into our data sources using Java (1).

---

\(^1\)http://www.getbootstrap.com
4. DEMONSTRATION

In this section, we describe how attendees will be able to experience X-REC at our demonstration stand. We will offer three modes of interacting with X-REC, differing in how much effort, and hence time, attendees have to invest:

- **Stereotype** relies on a handful of stereotypic user profiles for which X-REC has already determined entity recommendations, namely (i) **Geek**, (ii) **Punk**, (iii) **Redneck**, (iv) **Hipster**, (v) **Snob**. Table 2 shows for each of them example entities highly rated in each of the considered domains. In this least time consuming mode, attendees can pick one of the stereotypic user profiles and check whether the entities recommended by X-REC match their expectation.

- **Wikipedian** targets attendees with a known Wikipedia user name who have contributed non-anonymously to Wikipedia in the past. In this mode, X-REC bootstraps a user profile with entity ratings from the users’ past editing history. Once the generated user profile has been reviewed and possibly adjusted by the user, X-REC recommends entities within one of the selected domains.

- **Explorer** allows attendees to create a user profile from scratch. In doing so, X-REC supports them by auto completion and filtering based on entity category. Usually, once 10-15 entities have been rated, X-REC begins producing sensible entity recommendations.

All modes of interaction will be available at our demonstration stand. The system is publicly available at:

https://gate.d5.mpi-inf.mpg.de/xrec/

5. RELATED WORK

In this section, we compare X-REC against other approaches and systems proposed in the literature.

**Recommender systems** have been an active area of research in recent years, incentivized by efforts such as the Netflix competition [1]. Prevalent approaches can be broadly categorized into (i) **collaborative filtering** as methods based on only users’ ratings, (ii) **content-based filtering** as methods based on only descriptions of the items to be recommended, (iii) **knowledge-based methods** based on formally represented domain knowledge, and (iv) **hybrid methods** combining multiple of the aforementioned aspects. According to this categorization, X-REC uses a hybrid approach to entity recommendation. For an in-depth discussion of the state of the art, we refer to the textbook by Jannach et al. [7] and the survey by Ekstrand et al. [4].

**Systems** closely related to X-REC include: (i) **SuggestBot** [3] which aims at assisting Wikipedians with their contributions by recommending articles that need revision and fall into their area of expertise, (ii) **WikiLens** [5] which provides a toolkit for creating community-based recommender systems, and (iii) **SPARK** [2] which recommends knowledge base entities relevant to the user’s query. None of them, though, is designed to recommend entities from diverse categories purely based on publicly available data sources.

6. SUMMARY

We have presented X-REC as a novel system for entity recommendation. It relies on a simple yet easily extensible approach to recommend arbitrary entities from the YAGO2 knowledge base and does so using only publicly available data sources.

As future work, we plan to (a) incorporate other features (e.g., timestamps and significance of revisions) of the Wikipedia revision history into our approach, (b) use additional data sources such as Wikipedia access statistics, and (c) extend X-REC to bootstrap user profiles from other sources (e.g., profiles from facebook.com or del.icio.us).

7. REFERENCES