Augmenting a Feature Set of Movies Using Linked Open Data

Jaroslav Kuchař

Web Intelligence Research Group, Faculty of Information Technology
Czech Technical University in Prague, Prague, Czech Republic
jaroslav.kuchar@fit.cvut.cz

Abstract. Augmenting a feature set using mappings to the Web of data is an up-and-coming way to enrich data in the original dataset. Those enrichments are valuable especially for the recent preference learning algorithms and recommender systems. In this paper, we describe the process of mapping and augmenting the movie ratings dataset *MovieTweetings* from the perspective of *RecSysRules 2015 Challenge*. The ad-hoc queries to DBpedia are used as an underlying concept. To the best of our knowledge, there is no existing mapping dataset of movies for *MovieTweetings*. We also provide a brief discussion about the benefits of the augmented feature set for an elementary rule-based representation of the user preferences.

Keywords: web of data, mapping, user preferences, association rules

1 Introduction

In this paper, we are focused on a new type of problems which uses the Web of data to augment the feature set. Data in the original dataset are automatically mapped to the Linked Open Data (LOD) identifiers, and then additional features are generated from public knowledge bases such as DBpedia. The huge amount of achievable additional features can provide valuable information for various applications. Recommender systems and their preference learning algorithms have adopted the augmenting of the feature sets. The main goal is to overcome the issues with low granularity of available content descriptions on the one hand and data volume on the other hand [8]. Since association rules are recognized as one of the most suitable and understandable forms to represent knowledge and relations in data, we place emphasize on the benefits of enrichments for the user preferences represented by a set of rules. Rule-based representations of user preferences can thus provide a desirable balance between the quality of the representation and the understandability of the explanation for the human user [7].

The main contribution of this paper is that it presents an approach how to map an existing movie ratings dataset *MovieTweetings* [3] to the DBpedia, makes the mapping dataset available and discusses its benefits for rule-based user preferences. The presented approach is focused on ad-hoc SPARQL queries instead

of "guessing" *URIs* [10] or downloading all possible data to a local database and processing the data locally [2],[11]. To the best of our knowledge, there is no existing mapping dataset of movies for *MovieTweetings* to the Web of Data.

This paper is organized as follows. Section 2 examines a connection to *RecSysRules* challenge and provides an overview of dataset used for the challenge. Section 3 presents automatically generated mappings to the LOD cloud for an existing dataset, including the details on results. Section 4 briefly discusses the benefits of mappings for rule-based representation of user preferences. Finally, Section 5 summarizes the results.

2 Connection to RecSysRules 2015

The challenge *RecSysRules 2015*[1] has two focus areas: 1) rule learning algorithms applied on recommender problems 2) using the linked open data cloud for feature set extension. Since the mappings for the *MovieTweetings* dataset (as described in the rest of this paper) were not available at the time of organizing this challenge, the challenge uses a semantically enriched version of the MovieLens dataset [6]. As a mapping of MovieLens to Linked Open Data *DBpedia mappings to MovieLens1M dataset* [2] were used. Please note that due to the unavailability of all movies in DBpedia, the mapping for a fragment of movies is missing. For each movie in the mapping dataset the organizers extracted a set of categories and datatype properties (e.g. release data or gross) as an example of the augmented feature set. The *URI* identifiers to DBpedia were used to extract those features. In order to facilitate the distribution, the organizers do not provide the final dataset. Nevertheless, a Python script to download and build the dataset is available. This script downloads all necessary dependencies and creates the train CSV file as follows:

1. Download all dependencies including MovieLens ratings, mappings to DBpedia, augmented feature sets and configurations.
2. Filter ratings - select only ratings that correspond to a predefined set of users (randomly selected 1000 users by challenge organizers). There were also removed last 10 ratings for each selected user and moved to a test set. Test set was used for an evaluation of results submissions.
3. Augment a feature set of movies - for each movie that appeared in the filtered ratings, merge the movie with categories and properties from DBpedia. Entries without any available mapping are removed.
4. Export the train dataset as a CSV file.

The rest of this paper is focused on a way to provide mappings of movies to DBpedia for another dataset: *MovieTweetings*. The linking of movies is performed in a similar way as mappings for MovieLens. The paper also discusses the benefits of available links for preference learning.

[1] `http://2015.ruleml.org/recsysrules-2015.html`

3 Dataset Mapping

The goal is to provide a one-to-one mapping of movies from *MovieTweetings* dataset [3] to Linked Open Data cloud as *URI* identifiers. The dataset contains movie ratings extracted from Twitter for movies released from 1900s to the presence. Each movie is represented by a title, release date and a set of assigned genres (Example: *Rocky (1976), Drama | Sport*). The main advantage, compared to other existing datasets (MovieLens [6], Last.fm [1], Jester [4] or Book-Crossing [12]), is an availability of updates on a daily basis. Because the dataset is based on extraction of ratings from Twitter users around the world and it is daily updated, we have to deal with the following issues: multilingualism in titles, freshness, inaccuracies and incompleteness of data.

3.1 URI Alignment

Our proposed approach is designed to query the DBpedia using a set of predefined SPARQL queries performed in the following order:

Perfect match of a title: Listing 1.1 presents a SPARQL query to perform the perfect matching of the title and year according to the existing conventions for titles of movies in DBpedia (Example: *Rocky, Rocky (film) and Rocky (1976 film)*).

```
SELECT DISTINCT ?movie ?title ?category WHERE {
?movie rdf:type dbpedia-owl:Film ;
rdfs:label ?title .
?movie dcterms:subject ?category .
?category rdfs:label ?year .
FILTER (
  (
    (str(?title)="%s" || str(?title)="%s (film)")
   &&
    regex(?year,"^%s film", "i")
  )
  ||
  str(?title)="%s (%s film)"
  )
}
ORDER BY ASC(?movie)
```

Listing 1.1. SPARQL query - Perfect match of the title and year

Partial match of a title: Listing 1.2 describes a modification of the FILTER condition as a relaxation of the patterns in titles.

```
...
FILTER regex(?title,"%s", "i") .
FILTER regex(?year,"%s", "i")
...
```

Listing 1.2. SPARQL query - Partial match of the title and year

Pattern-based match of an abstract: Based on the nature of DBpedia abstracts formatting we use an abstract as a possible candidate for the pattern matching. The common format of an abstract is: *Rocky is a 1976 film ...* or *... Rocky ... released 1976 ....*

```
...
FILTER (
  regex(?abstract,"^%s is a %s", "i")
  ||
  regex(?abstract,"^%s .* releas.* %s", "i")
)
...
```

Listing 1.3. SPARQL query - Pattern-based match of the abstract

Any match of an abstract Last case is when there is no match to any previously described patterns. For foreign languages, abstract usually contains textual mentions about titles of the movie in foreign languages (Example: *... also known as ...* or *... (Italian: ..., German: ...)*)

```
...
FILTER regex(?abstract,"%s", "i").
FILTER regex(?year,"%s", "i")
...
```

Listing 1.4. SPARQL query - Any match of the abstract

3.2 Confidence Values

To express a basic relevance of the mapping to *URI* identifiers from *DBpedia*, we provide a set of confidence values. *Title confidence(tc)* is computed using Levenshtein distance of titles, *Year Confidence (yc)* is computed as a simple distance of years and *Genre Confidence (gc)* uses number of common genres. Those values are available in the final mapping dataset and can be used together with a method name for filtering of results. The setting of the filtering is left to the end-user of the mapping dataset.

3.3 Results and Statistics

In this section we will briefly describe results of the mapping. We use a snapshot of the dataset downloaded on June 1, 2015. It contains over 21000 movies. At the time of publishing of this paper, the mapping provides URIs for 71.3% movies. The remaining movies were not mapped due to the issues mentioned at the beginning of this section.

Figure 1 depicts distribution of years for movies that were not successfully mapped to any *URI*. There is a large amount of movies from recent time that were not successfully mapped due to their unavailability in DBpedia. The reason is that the current version of DBpedia was published on September 9, 2014

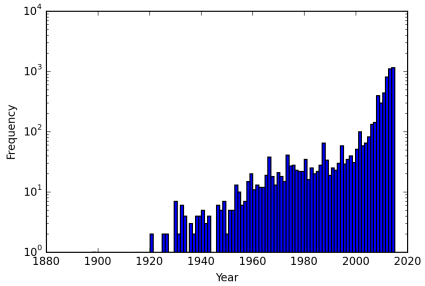

Fig. 1. Distribution of years for unmapped movies

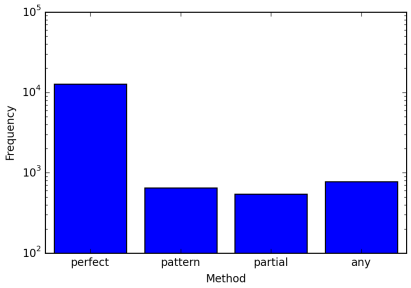

Fig. 2. Overview of methods used for successful mapping

(based on Wikipedia dumps from April/May 2014)[2]. Figure 2 demonstrates usage of methods for successful mapping of movies. The method that performs the perfect match of a title and a year is the most frequent (perfect: 86.61%, pattern: 4.38%, partial: 3.66%, any: 5.35%). Figure 3 provides an overview of language distribution in titles.[3] This summary presents the availability of mappings to *DBpedia* for various languages.

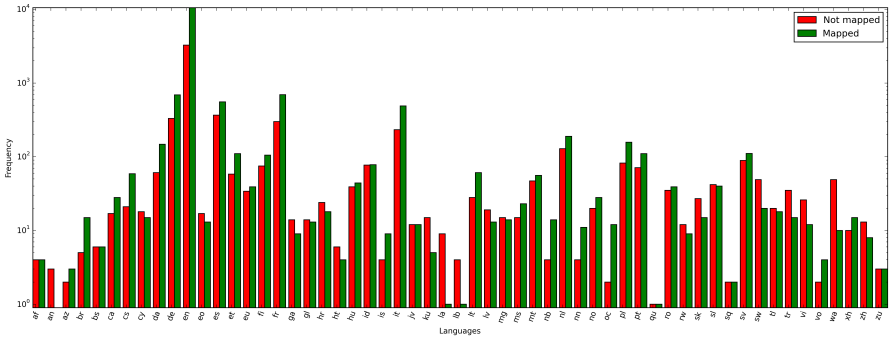

Fig. 3. Distribution of mapped/unmapped movies with respect to languages detected in movie titles

We also evaluated our approach using another existing mapping dataset for MovieLens [2]. We selected this dataset because both original datasets (MovieLens and MovieTweetings) are provided in the same format and the authors of the mapping dataset for MovieLens deal with the same task: mapping of movies

[2] `http://wiki.dbpedia.org/news/dbpedia-version-2014-released`
[3] Languages detected in titles using *LangID*: `https://github.com/saffsd/langid.py`

to DBpedia. Furthermore, the dataset was manually corrected, therefore we can use it as a ground truth. We launched the proposed mapping algorithm and compared to available mappings. Our approach achieved over 98.5% match, where the incorrectly mapped values were either missing *URIs* or incorrect links that can be filtered using the confidence values.

4 Rule-based User Preferences

In this section we will briefly discuss the benefits of the augmented dataset from the perspective of the challenge *RecSysRules 2015*.

Association rules are recognized as one of the most suitable and understandable forms to represent knowledge and relations in data. Rule-based representations of user preferences can thus provide a desirable balance between the quality of the representation and the understandability of the explanation for the human user. The user preferences may be used in different scenarios or use cases from elementary user profile representations to rating predictions and recommendations. In this paper we consider a subset of association rules, called *class association rules (CARs)* [9]. Those rules are in the specific format, where a right-hand side of a rule (consequent) contains only one attribute and this attribute is a classification class attribute.

4.1 Illustrative Example

Let consider the domain of movies and information about ratings provided by users from MovieTweetings dataset. The presence of a user rating for a specific movie can be considered as an interest clue - the implicit information about the positive user preference for the movie. For ratings prediction tasks, the provided ratings can be considered as a level of interest. However, it is beyond the scope of this paper to elaborate on all possible tasks. The rest of this illustrative example is focused on the positive-only feedback and the item recommendation task. Each movie is basically represented by a set of features - associated genres. Table 1 provides example for one user from the MovieTweetings dataset.

Table 1. Example of input data from MovieTweetings dataset (User Id: 455)

MovieId	Title	Features (Genres)	Interest
468569	The Dark Knight (2008)	"Action", "Crime", "Drama", "Thriller"	positive
1345836	The Dark Knight Rises (2012)	"Action", "Crime", "Thriller"	positive
1951264	The Hunger Games: Catching Fire (2013)	"Action", "Adventure", "Sci-Fi", "Thriller"	positive

The elementary rule-based user preferences can be mined using an association rule mining algorithm (e.g R arules package [5]). Example of extracted rules, that

represents the user preferences for one specific user (User Id: 455, minConfidence: 0.1, minSupport: 0.1):

- $\{Action\} \rightarrow \{positive\}$ (support=1.0, confidence=1.0)
- $\{Action\&Thriller\} \rightarrow \{positive\}$ (support=1.0, confidence=1.0)
- $\{Crime\} \rightarrow \{positive\}$ (support=0.67, confidence=1.0)

The drawback of the previously described preferences is that they consider only genres as a key component. It is a limiting factor of this representation since those genres are too general. The total number of unique genres in the dataset is 28. In case we would like to use those rules to find candidates for other interesting movies to the user, the rules match too many movies as a set of possible candidates (2952, 1130 and 2717 matched movies respectively).

Table 2. Excerpt from an augmented feature set for MovieTweetings (User Id: 455)

Title	Features
The Dark Knight (2008)	"American action thriller films", "Batman films", "Films directed by Christopher Nolan", ...
The Dark Knight Rises (2012)	"Batman films", "Warner Bros. films", ...
The Hunger Games: Catching Fire (2013)	"2010s science fiction films", "The Hunger Games (film series)", "American fantasy adventure films", ...

The mappings of movies to the Linked Open Data (See previous section for more details) can help to overcome this issue. Linked Open Data cloud contains relevant information to augment the feature set and increase the granularity. The *URI* as an identifier of data related to the associated movie can be used to extract additional features; a set of assigned categories for this example[4]. Table 2 demonstrates excerpt of an augmented feature set for the movies from our example. We use a basic SPARQL query to extract all categories associated with the specific movies.

Sample of three representative rules mined on the augmented feature set (User Id: 455, minConfidence: 0.1, minSupport: 0.1):

- $\{Warner\_Bros.\_films\} \rightarrow \{positive\}$ (support=0.67, confidence=1.0)
- $\{Batman\_films\} \rightarrow \{positive\}$ (support=0.67, confidence=1.0)
- $\{The\_Hunger\_Games\_(film\_series)\} \rightarrow \{positive\}$ (support=0.33, confidence=1.0)

Using the Linked Open Data Cloud we get more granular features for representations of movies. In total there are 10 950 unique categories for all movies in the dataset. The availability of a set of more granular categories assigned to each movie and rule-based user preferences considering those categories, the number

[4] Categories are identified by predicate http://purl.org/dc/terms/subject

of movies that match preferences should be decreased. For our illustrative experiment, the number of matching movies are as follows: 859, 9, 4. The first rule contains more general category, but the remaining two are able to provide adequate number of candidates based on the preferences.

5 Conclusion and Future Work

In this paper we demonstrate the approach to augment the existing movie ratings dataset *MovieTweetings* from the perspective of the *RecSysRules 2015* challenge. We provide the dataset as a mapping of movies to DBpedia for further experiments. It is available for download on the Github[5]. It can be used for other content-based recommender systems as well. We also discussed the benefits of augmented feature sets for the elementary rule-based representations of user preferences. We plan to perform extensive experiments with rule-based user preferences boosted by the augmented feature set. Last but not least, we plan to improve the mapping patterns, offer the mappings to other knowledge bases and provide updates of mapping dataset on a regular basis.

Acknowledgments. This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS14/104/OHK3/1T/18.

References

1. Oscar Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer Publishing Company, Incorporated, 1st edition, 2010.
2. Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 1–8, New York, NY, USA, 2012. ACM.
3. Simon Dooms, Toon De Pessemier, and Luc Martens. Movietweetings: a movie rating dataset collected from twitter. In *Workshop on Crowdsourcing and Human Computation for Recommender Systems, CrowdRec at RecSys 2013*, 2013.
4. Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2):133–151, July 2001.
5. Michael Hahsler, Bettina Grün, and Kurt Hornik. arules - a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, 9 2005.
6. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA, 1999. ACM.

[5] `http://github.com/jaroslav-kuchar/MovieTweetingsMappings`

7. Tomáš Kliegr and Jaroslav Kuchař. Orwellian eye: Video recommendation with Microsoft Kinect. In *Proceedings of the Conference on Prestigious Applications of Intelligent Systems (PAIS'14) collocated with European Conference on Artificial Intelligence (ECAI'14)*, pages 1227–1228. IOS Press, 2014.
8. Jaroslav Kuchař and Tomáš Kliegr. Bag-of-entities text representation for client-side recommender systems. In *First Workshop on Recommender Systems for Television and online Video (RecSysTV), ACM RecSys*, 2014.
9. Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In Piatetsky-Shapiro G. Agrawal R., Stolorz P., editor, *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 80–86, 1998.
10. Heiko Paulheim and Johannes Fümkranz. Unsupervised generation of data mining features from linked open data. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, New York, NY, USA, 2012. ACM.
11. Matthew Rowe. Semanticsvd++: Incorporating semantic taste evolution for predicting ratings. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*, WI-IAT '14, pages 213–220, Washington, DC, USA, 2014. IEEE Computer Society.
12. Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 22–32, New York, NY, USA, 2005. ACM.