A Hybrid Method for Rating Prediction Using Linked Data Features and Text Reviews

Semih Yumusak[1,2], Emir Muñoz[2,3], Pasquale Minervini[2], Erdogan Dogdu[4], and Halife Kodaz[5]

[1] KTO Karatay University, Konya, Turkey,
semih.yumusak@karatay.edu.tr
[2] Insight Centre for Data Analytics, National University of Ireland, Galway,
[3] Fujitsu Ireland Limited,
[4] TOBB University of Economics and Technology,Ankara, Turkey
[5] Selcuk University, Konya, Turkey

Abstract. This paper describes our entry for the Linked Data Mining Challenge 2016, which poses the problem of classifying music albums as 'good' or 'bad' by mining Linked Data. The original labels are assigned according to aggregated critic scores published by the Metacritic website. To this end, the challenge provides datasets that contain the DBpedia reference for music albums. Our approach benefits from Linked Data (LD) and free text to extract meaningful features that help distinguishing between these two classes of music albums. Thus, our features can be summarized as follows: (1) direct object LD features, (2) aggregated count LD features, and (3) textual review features. To build unbiased models, we filtered out those properties somehow related with scores and Metacritic. By using these sets of features, we trained seven models using 10-fold cross-validation to estimate accuracy. We reached the best average accuracy of 87.81% in the training data using a Linear SVM model and all our features, while we reached 90% in the testing data.

Keywords: Linked data, SPARQL, Classification, Machine Learning, #Know@LOD2016

1 Introduction

The potential of using the datasets available in the Linked Open Data (LOD) cloud[6] supporting several tasks in Data Mining (DM) has been pointed out several times (see [3] for a survey). For instance, the rich content of domain specific and general domain datasets could be used to generate semantically meaningful feature sets. The linked characteristic of the datasets in the LOD cloud allows for querying features from different sources for a given entity. The Linked Data Mining Challenge provides DBpedia URIs that we use to query the DBpedia knowledge base and extract features of the considered entity. DBpedia [2] knowledge base contains descriptive information about albums that can be extracted using SPARQL to query for relevant triple patterns. For instance, we can start from a DBpedia music album URI and access all related metadata. Furthermore, we can access extra information by navigating the links in the graph and get, for example, information about the artist(s) or band that recorded the album, number of awards of the album or artist(s), and information about producers, among others.

Although users are empowered with the ability to navigate *linked* data, they still face the same classical challenges associated to DM, such as feature selection, model selection, etc. Previous work on this task [1], highlights limitations of features coming solely from DBpedia. Extra information could come from textual critics from Metacritic[7]. Here, we follow a similar approach, enriching DBpedia to find the best set of features for distinguishing between the two classes (§ 2). Results of our experiments show that taking all considered features into account yields the best classification performance (§ 3). Conclusions and final remarks are reported in Section 4.

[6] http://lod-cloud.net/

[7] http://www.metacritic.com

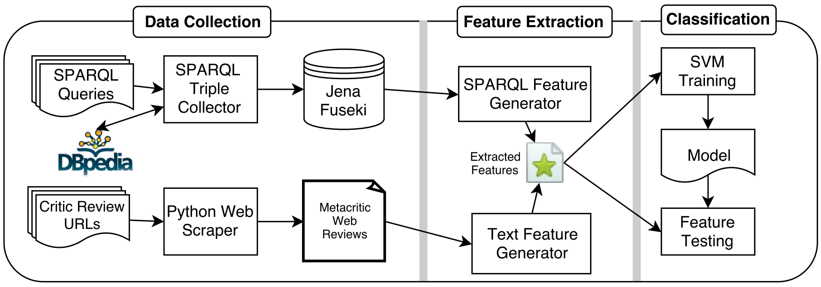

Fig. 1. System architecture

2 Methodology

We start from the DBpedia knowledge base for referencing of metadata about all albums in the training and testing datasets. By leveraging such a knowledge base, we defined a set of features which are potentially relevant to the classification task. As shown in [1], features coming from textual data (such as reviews) are also relevant for a classification problem. Therefore, in addition to pure Linked Data features, we collected the textual reviews from Metacritic website, and consider the words content as features herein. Our approach steps (as shown in Figure 1) can be summarized as follows:

Data Collection. First, we collected and analysed the DBpedia knowledge base and the Metacritic reviews. For each music album, we crawled the summaries of the corresponding Metacritic reviews for an album and artist[8]. The critic reviews were scrapped and saved as text, converted into RDF and linked to DBpedia using the `dbp:rev`[9] property in a Jena Fuseki instance.

Feature Extraction. Starting from DBpedia knowledge base, a manual selection of predicates was carried out, leaving out less frequent and irrelevant predicates. With the remaining predicates, we defined a set of questions and hypotheses that we later test (see Table 1). Based on our two sources, our features are divided into two sets: (1) Linked Data-based features, and (2) Text-based features. Set (1) is further divided into: (1-1) Linked Data object specific features, where values of specific predicates are directly used; and (1-2) aggregating features, where we use the count of values of given predicates. In the case of Metacritic reviews, we follow a Bag of Words approach for part (2) to find the most discriminant words for each class. Formally, we generate the following vectors as features: $\mathbf{x}^{(\mathsf{LD})} = (f_1, \ldots, f_m)$ to represent the (1-1) features (t_1 to t_{14}), where $m = 15009$; $\mathbf{x}^{(\mathsf{LDA})} = (f_1, \ldots, f_n)$ to represent the (1-2) features (t_{15}, t_{16}), where $n = 4$; and, $\mathbf{x}^{(\mathsf{TEXT})} = (f_1, \ldots, f_q)$ to represent the (2) features (t_{17}), where $q = 21973$ is the cardinality of the extracted vocabulary.

In order to answer each question in Table 1, we submitted SPARQL to our enriched DBpedia knowledge base. For example, the query to get a direct object feature like genre(s) of the album `<AlbumURI>`:

```
SELECT ?o WHERE {<AlbumURI> dbo:genre ?o.}
```

Similarly, we get the aggregation features, e.g., the number of extra albums for the producer of album `<AlbumURI>`:

```
SELECT count(?s) WHERE {<AlbumURI> dbo:producer ?o1. ?s  dbo:producer ?o1. ?s a dbo:album>}
```

[8] We use URIs as `http://www.metacritic.com/music/AlbumName/ArtistName/critic-reviews`

[9] URI namespaces are shortened according to prefixes in `http://prefix.cc/`

Table 1. Domain-specific questions, hypotheses, and predicates with their accuracy

#	Question	Hypothesis	Predicate	SVM Acc.
t_1	What are the topics (dct:subjects) for the album? (baseline)	Some albums belong to successful subjects, and vice versa.	`dct:subject`	58.05%
t_2	Who is the artist of the album?	Some artists are more famous than others	`dbo:artist`	48.91%
t_3	Is the artist a band, single artist, etc.?	Bands are more successful than single artists	`rdf:type` of `dbo:artist`	61.95%
t_4	What genres the album belongs to?	Some genres are more popular than others	`dbo:genre`	66.33%
t_5	What are the language(s) in the album?	Albums in English are more likely to be popular	`dbo:language`	47.27%
t_6	Who recorded this album?	Some labels are more popular and record more albums.	`dbo:recordLabel`	49.06%
t_7	Are long albums more popular?	Long albums tend to be more popular	`dbo:runtime`	46.48%
t_8	Who is the director of the album?	Certain directors/artists are more successful	`dbp:director`	47.19%
t_9	What is the region of the album?	Albums created in certain regions are more likely to be successful	`dbp:region`	51.72%
t_{10}	What studio created the album?	Some studios create high quality works, some do not.	`dbp:studio`	47.19%
t_{11}	What is the total length of the album?	Shorter albums are likely to be worse.	`dbp:totalLength`	54.69%
t_{12}	Who are the songwriters of the album?	The songwriters in the album affects the popularity of the album	`dbp:writer`	47.19%
t_{13}	Who are the reviewers of the album?	Some reviewers are likely to review only good or bad albums.	`dbp:rev`	71.41%
t_{14}	What are the topics (dct:subjects) for the artist?	Particular artists are likely to be categorized under certain subjects.	`dct:subject` of `dbo:artist`	68.59%
t_{15}	How many awards does an artist have?	Albums of award winning artists are likely to be more successful	# awards of `dbo:artist`	47.19%
t_{16}	How many other albums a producer of this album have?	Some producers are more successful and produce more albums than others.	# albums by `dbo:producer`	54.53%
t_{17}	Are textual reviews useful for the classification?	A Bag of Words approach can help to separate the classes	BoW	85.00%

During our manual analysis, we noticed that some properties (e.g., `dbp:extra`, `dbp:source`, `dbp:collapsed`, `dbp:extraColumn`, `dbp:type`) have a strong correlation with the class 'good' over 'bad', and vice versa. These properties are also collected and added to the LD feature set. Moreover, some properties are directly related to Metacritic scores (`dbp:mc` is the actual Metacritic score), and other (critic) scores, like `dbp:revNscore` whose values range from 1 to 15. To keep our models unbiased, we decided to exclude them from our extraction.

Besides regular DBpedia properties, we also selected features from textual reviews. For each review, we use Bag-of-Words with lower-case and non-alphanumeric normalizations and stopwords removal. For this, NLTK library[10] was used for stemming and lemmatization of words longer than 2 characters. In [1], the authors also show that aggregation features provide better results when discretized, e.g., based on their numeric range. For instance, the award feature of an artist could be marked as 'high' if the number of awards is more than one; and 'low' otherwise. For other numeric (property) values, we have identified the average values and use them to discretize the values as 'high' (above average) and 'low' (below average). Few average examples are runtime is 2800 sec., number of albums per producer is 40, total length is 2900 sec.

Classification. We trained seven different models listed in Table 2 using k-fold cross-validation ($k = 10$). Each model was trained with five different sets of features, and evaluated using accuracy, $Acc = \frac{tp+tn}{tp+fp+fn+tn}$. The hyperparameters for each model were determined manually via incremental tests, and results extracted from the training set. For example, for SVM we tested a linear kernel with $C \in [0.001 - 0.1]$ and found 0.025 as best performing value.

[10] `http://www.nltk.org/`

3 Experimental Results and Analysis

For our experiments we used the sckit-learn library[11] that supports the training of the proposed seven classifiers using different combinations of our features. Table 2 shows the accuracy values for the best validation values for all seven models with each set of features. We report our best cross-validation accuracy 87.81% on the training set, whilst the challenge system reports 90% for our submission on the testing set. This might be seen as an indication that our models did not overfit on the training data, and they are able to generalise to unseen data. We attribute this mainly to our decision to leave out predicates that are directly or indirectly related to scores for the music albums. We would also like to highlight the use of textual features to increase the true positives and false negatives. Considering solely LD features reached up to 76.64%, while considering solely TEXT features reached up to 85%, both using the SVM model. This fact shows that for a classification problem like this, DBpedia still does not provide enough meta-information for the entities, and other sources must be taken into account. Also we tested our hypotheses with the best performing model and extract accuracy for each one in Table 1.

Table 2. Comparative Analysis of Feature Sets and Classifiers

Feature Set	Linear SVM	KNN	RBF SVM	Dec. Tree	Rand. Forest	AdaBoost	Naïve Bayes
LD	76.64%	60.47%	48.05%	72.66%	53.91%	75.00%	76.41%
LDA	54.53%	52.58%	54.69%	54.45%	48.91%	54.53%	52.89%
LD+LDA	76.72%	60.23%	48.05%	72.66%	52.34%	75.00%	76.41%
TEXT	85.00%	50.00%	47.27%	67.27%	52.81%	78.91%	68.44%
LD+LDA+TEXT	**87.81%**	52.81%	47.27%	72.03%	52.58%	82.50%	77.19%

4 Conclusion

In this paper, we addressed the problem of classification by using features from Linked Data and text reviews. We experimented with several properties related to music albums, however, we noticed that by also considering textual features we could reach higher accuracies. We enriched our knowledge base with textual critics and use them as Bag of Words. We selected our model using 10-fold cross-validation: our best model also showed good predictive accuracy on the test set as reported by the challenge system. This is an indication that our manual analysis and feature selection was a useful pre-processing step. For reproducibility, all source files, crawler code and reviews, enriched knowledge base in RDF, and intermediate files are published as an open-source repository[12].

Acknowledgement. This research is partly supported by The Scientific and Technological Research Council of Turkey (Ref.No: B.14.2. TBT.0.06.01-21514107-020-155998)

References

1. Aldarra, S., Muñoz, E.: A Linked Data-Based Decision Tree Classifier to Review Movies. In: Proc. of the 4th International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data at ESWC 2015. CEUR Workshop Proceedings, vol. 1365. Portoroz, Slovenia (May 2015)
2. Lehmann, J., Isele, Robert and Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S.: DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web 6(2), 167–195 (2015)
3. Ristoski, P., Paulheim, H.: Semantic web in data mining and knowledge discovery: A comprehensive survey. Web Semantics: Science, Services and Agents on the World Wide Web (2016)

[11] `http://scikit-learn.org/`

[12] `https://github.com/semihyumusak/KNOW2016`