Predicting Metacritic Film Reviews Using Linked Open Data and Semantic Technologies

Meyer A. Bossert

Cray Inc., Seattle Washington, USA
bossert@cray.com

Abstract. Aristotle was quoted as saying that "the whole is more than the sum of its parts". Using Linked Open Data, we are finally able to test and quantify Aristotle's theory[1]. By using the flexible data representation of RDF as well as the graph-oriented nature of SPARQL, we attempt to answer the questions: What makes a movie good? And more specifically, what makes a critic think that a movie is good? We take a novel approach to predictive analytics that is implemented entirely in SPARQL rather than using more traditional statistical machine learning platforms (e.g. Rapidminer, etc.).

1 Introduction

The Linked Data Mining Challenge (LDMC)[2] proposes that we use Linked Open Data (LOD) sets in order to predict Metacritic reviews. Data in the Resource Description Framework (RDF) format provides many advantages; Principal among these are the flexible nature of data representation, especially when multiple properties are associated with a given entity, but in an inconsistent way (i.e. not every film's DBpedia entry contains the same properties). Further, RDF is better suited for analyzing relationships between entities, especially when the potential relationships are not known; which allows us to develop a much more robust and flexible mechanism for predicting the general impression that critics have of a given film.

In the interest of brevity, all materials other than the source datasets consisting of the DBpedia 2014 dump[3] and the Freebase 2014 RDF dump[4] have been stored on Github[5] and made accessible to the general public under the MIT license.

[1] This statement is made with a little humor in mind. At no point does this work attempt to seriously tackle Aristotle's work.

[2] http://knowalod2015.informatik.uni-mannheim.de/en/linked-data-mining-challenge/

[3] http://wiki.dbpedia.org/Downloads2014

[4] http://commondatastorage.googleapis.com/freebase-public/rdf/freebase-rdf-latest.gz?

[5] https://github.com/mabossert/LDMC_2015

2 Analytic Approach

In order to predict Metacritic reviews based on the training and testing data provided, we note the steps needed to accomplish our goal. Some pre-processing is required as is some general data cleansing prior to tackling the challenge of generating predictions.

2.1 Pre-processing

In order to link the provided training and testing data to the DBpedia dataset, we develop a simple Perl script to convert each row of data into RDF format (N-Triples). Also, because the provided DBpedia URI's are not directly connected to all the available data, we run SPARQL queries that identify URI's within the provided datasets that contain little information, have a disambiguation link, or a redirect link. When one of those linked URI's matches the release year and also contains significantly more properties than the initially provided one, we swap them out[6].

2.2 Data Cleansing

Using the Cray Urika GD[7] graph appliance, which implements the Apache Jena quad store on top of a global shared memory platform containing 2TB of shared memory, we load the entire DBpedia and Freebase datasets as well as the training and testing datasets. The entire dataset consists of approximately 3.45 billion triples.

We ignore properties and predicates that are obviously irrelevant or have the potential to produce erroneous predictions (i.e. owl:Thing is not germane to the problem).

In order to implement the data cleanup, we perform INSERT and DELETE operations to flag both predicates and objects associated with films as "keep" or "drop"[8]. This step is performed in lieu of a long and complex series of FILTER operations in the actual queries used to generate the film score predictions.

In order to simplify the SPARQL queries, we transpose all of the desired properties associated with film entities found in the Freebase dataset to their equivalent entities found in the DBpedia dataset (e.g. as identified by the owl:sameAs relationship)[9].

2.3 Predictive Algorithm Development

Earlier, we alluded to Aristotle's famous quote "The whole is greater than the sum of its parts"; our theory is that if we know, for each attribute associated with a film, on average how many times that attribute is associated with a "good" or "bad" film, then we can surmise with some degree of certainty that the score as determined by taking

[6] Execution time to disambiguate/correct is approximately 13 seconds

[7] http://www.cray.com/products/analytics/urika-gd

[8] Execution time to insert new triples (e.g. "keep" and "drop" properties) is approximately 5 seconds

[9] Execution time to transpose Freebase properties is approximately 34 seconds

the average of all desired[10] attributes will be a good indicator of the likelihood of a film receiving positive or negative reviews as quantified by the overall Metacritic score. Finally, we acknowledge that there are some properties that should be considered with a higher weight than others. In particular, we make the assumption that films that receive awards of any type are likely to considered "good"[11]. Though most films that receive awards are well received by critics, not all are, thus we account for this observation by weighting awards differently based on if films in the training dataset were rated as "good" or "bad" (e.g. weight of 5)[12][13].

2.4 Algorithm tuning

The optimal breakpoint is a score of 55 percent or higher and the weight for award-related properties is 5[14]. Finally, we add additional weight to properties that were particularly polarized with respect to the ratio of good to bad films associated with the properties in question. The following table shows the breakdown of weighting[15].

Percent good vs. bad	Weight
>= 75%	0.2
<= 35%	-0.7

The final calculation[16], per property is expressed in the following formula. Let G be the final weighted score for each property, let g and b be the number of instances where a film was rated as "good" or "bad", respectively, let m be the multiple and t be the additional weight for polarized properties. The final score for each film is the average of its constituent weighted scores.

$$G = (g \div (g + b)) \times (m + t)$$

These experimental results are confirmed by an overall accuracy of 92.25%[17] (further detail shown in table below). So, with our lighthearted poke at Aristotle, we can now say that the delta between the sum of the parts and the whole is roughly 7.75%.

[10] As determined in the data cleansing step identified earlier with "keep" and "drop" values.

[11] The average percentage of films rated as "good" was 86.17%.

[12] A full list of the properties can be found at https://github.com/mabossert/LDMC_2015

[13] The weight is multiplied by the actual percentage, potentially producing a score above 100%, which serves to emphasize the higher likelihood that awarded films will receive positive reviews

[14] All non-award related properties have a weight of 1 (e.g. the multiple)

[15] All weighting factors were produced by iterating over possible values against the training dataset (i.e. brute-force), finally selecting the combination that produced the best accuracy for the training dataset

[16] Execution time to predict film review class (e.g. good or bad) is approximately 14 seconds

[17] The un-tuned algorithm generated an accuracy of approximately 84%

PRECISION	RECALL	ACCURACY
90.82%	94%	92.25%

3 Interesting observations

Our assumption had been that, in general, films featured at a film festival would be disproportionately well reviewed by critics, however, our experiments showed that there was little correlation between film festivals and good critical reviews despite the fact that the average percentage of good vs. bad films that had properties associated with film festivals was 80.34% for the training dataset.

Initially, we restricted the types of properties that were considered to just a few obvious choices (e.g. actors, directors, etc.), which resulted in relatively poor accuracy. Once most property types were included, accuracy was drastically increased. We hypothesize that it is actually the rich variety of properties for any given film that makes this approach produce more accurate results.

Documentaries deserve a special mention as well. Regardless of the film, those that were identified as documentaries received overwhelmingly high praise from critics[18].

Finally, we observe that it is slightly easier[19] to predict the review class of good movies than bad ones. We hypothesize that the reason for the imbalance is that good movies tend to have a wide variety of information entered into DBpedia and Freebase while bad movies tend to have less effort put into their documentation[20].

4 Future work

In future iterations of this predictive algorithm, we will attempt to overcome the problems of sparse data by incorporating community detection algorithms as part of the pre-processing.

We will also incorporate more datasets (e.g. IMDB, Rotten Tomatoes, etc.) as well as Natural Language Processing (NLP) of actual critic review text (e.g. entity extraction and sentiment analysis) in order to derive deeper understanding of each film.

5 Conclusion

In our experiments, we find that we can predict Metacritic reviews of any given film with great accuracy so long as it has data found or subsequently inserted in LOD. The aspect of this work that was most interesting to us was that we could break from traditional statistical machine learning approaches to get to an acceptable result. Using nothing more than SPARQL queries, any person can implement this same technique; democratizing data analysis and access is, of course, one of the main principles behind LOD projects. Further, we note that the same algorithm, with minor adjustments, can be applied to many other types of entities and could have significant impacts in areas like social media and news predictive analytics, for instance.

[18] The average percentage of good vs. bad documentaries is 97.5% for the training dataset.

[19] As noted by observing 12 false negative and 19 false positive predictions

[20] Based on the training dataset, “good” films have an average of 218.81 properties, while “bad” films have an average of 169.54 properties