Not-So-Linked Solution to the Linked Data Mining Challenge 2016

Jedrzej Potoniec

Institute of Computing Science, Poznan University of Technology
ul. Piotrowo 2, 60-965 Poznan, Poland
Jedrzej.Potoniec@cs.put.poznan.pl

Abstract. We present a solution for the *Linked Data Mining Challenge 2016*, that achieved 92.5% accuracy according to the submission system. The solution uses a hand-crafted dataset, that was created by scraping various websites for reviews. We use logistic regression to learn a classification model and we publish all our results to *GitHub*.

1 Introduction

As indicated in the challenge website, Linked Data Mining is a novel and challenging research area, mainly due to large amount, variety and heterogeneity of the data. Unfortunately, there are also very basic, almost technical, problems with the data: they do not comply with the standards, there is a lot of mistakes introduced during extraction and transformation from an original format to the Linked Data, websites publishing the data are frequently down [1]. Because of that, we decided to take another path in our solution. We did the extraction by ourselves, thus receiving dataset well-suited for the challenge, as described in Section 2. We performed normalization and applied a very popular logistic regression method to train a classification model, as described in Section 3.

Throughout the rest of the paper, we use a prefix `dbr:` for `http://dbpedia.org/resource/` and `dbp:` for `http://dbpedia.org/property/`. Web scraping scripts, created dataset, machine learning process and model are available on *GitHub*: `https://github.com/jpotoniec/LDMC2016`.

2 Datasets

2.1 Training and test data

We observed some irregularities and unexpected things in the datasets provided by the challenge. For the album *In Some Way, Shape, or Form* by *Four Year Strong* the data pointed to the resource `dbr:In_Some_Way,_Shape_or_Form`. Unfortunately, in the *DBpedia* [2] there are two corresponding resources, the other one being `dbr:In_Some_Way,_Shape,_or_Form` (note the second comma). The *DBpedia* website does some sort of redirection, so when visiting with a web browser both URIs point to `http://dbpedia.org/page/In_Some_Way,_Shape,`

`_or_Form`. Conversely, the SPARQL endpoint[1] treats both URIs separately, the first one occurring in 18 triples and the second one in 100 triples.

For an artist *St. Vincent* there are two albums in the datasets: *Strange Mercy* in the training data and *St. Vincent* in the testing data. Unfortunately, both have the same URI `dbr:Strange_Mercy`. We think there may be a few similar issues, as there are eight URIs that occur more than once in the datasets.

2.2 Linked datasets

In the beginning, we planned to extract features from *DBpedia* using *Fr-ONT-Qu* [4] from *RMonto* [6], a plugin to *RapidMiner* [5]. Unfortunately, the most promising feature discovered this way was a binary information if an album has a review score from *Pitchfork*[2] or not. After investigating, we discovered that during the extraction from *Wikipedia* to *DBpedia* a relation between a reviewing website and a review score has been lost. Consider triples for the *Strange Mercy* album[3]: there are 11 triples with a property `dbp:rev` and a few triples with properties like `dbp:rev10score`, but one has absolutely no way to connect scores to the reviewing websites. The very same problem happens with properties `dbp:title` (12 values) and `dbp:length` (11 values): it is impossible to decide on a length for a concrete track. Due to the lack of space, we present a detailed analysis in the suplementary material available in *GitHub*.

We thought about using *Yago* [7], but it seemed to lack review information. We also tried to use *DBTune*[4], as suggested by the challenge website, but it rendered out to be a dead end. For example, *MusicBrainz data*, the most interesting dataset there, is a *Service Temporarily Unavailable* for a very long time now.

2.3 Non-linked datasets

Instead of trying to use existing Linked Data, we decided find data to solve the challenge, and then make it available to the community. As the datasets for the challenge are quite small (1592 different URIs), we did some web scrapping with *Python* and *Scrapy*[5] to obtain reviews of considered albums.

We scraped *Wikipedia* to obtain reviews collected from various websites. It rendered out to be a tedious process, as these reviews have various formats, frequently with some additional information (like a date or an URL to a review), or spelling mistakes. We performed normalization to a range $[0, 1]$, by dividing in case of reviews on scales from 0 to 10 or 100 and by assigning arbitrarily numeric values to descriptive reviews (like *favorable*). We also did some heuristic to normalize reviewing websites, e.g. *BBC* and *BBC Music*. We strictly avoided using *Metacritic* reviews available in *Wikipedia*: these reviews use `MC` field in `Album ratings` template, while we used only fields starting with `rev` [10].

[1] `http://dbpedia.org/sparql`
[2] `http://pitchfork.com/`
[3] `http://dbpedia.org/page/Strange_Mercy`
[4] `http://dbtune.org/`
[5] `http://scrapy.org/`

We collected number of reviewers and an average score from *Amazon*[6] by scraping the website using titles and artists provided in the challenge datasets. We also used *Discogs*[7] API and provided titles and artists to gather how many people own an album and how many people want it. Finally, we used datadumps provided by *MusicBrainz*[8] [8] and identifiers from Wikidata [9] available in the datadumps and in *DBpedia*, to obtain number of users owning an album and it's average score. The whole dataset consists of 94 numerical attributes.

3 Machine learning process and model

To build a classification model, we used a typical machine learning setup for classification. We performed a Z-transformation on all attributes, that is for every attribute we computed an average value μ and a standard deviation σ, and then replaced every value v of the attribute with $\frac{v-\mu}{\sigma}$. This way all attributes have an average 0 and a standard deviation 1. Further, we replaced missing values with 0. Finally, we used logistic regression [3] to train the model.

To estimate performance of our solution we applied 10-folds cross-validation, which estimated accuracy to be $91.7 \pm 2.17\%$. This value is consistent with 92.5% on the test set reported by the challenge submission system. The whole process have been implemented using *RapidMiner 5* and is available in *GitHub*.

An important part of logistic regression is to assign coefficients to attributes of an input dataset. Values of these coefficients provide an insight which attributes are most important for the model. In our case, the absolute value of the highest coefficient is 2.859 and the lowest 0.022. As all the attributes are on the same scale, this clearly shows that some of them are more important than the others. There were six attributes having coefficients above 1, we present them in the Table 1. Five of these attributes were review scores web scrapped from *Wikipedia*, only the attribute from *Discogs* came from other source. These attributes clearly indicate that *Metacritic* reviews are quite consistent with other sources of reviews. The attribute with the highest coefficient is an review value from *Pitchfork*, what is consistent with the most important attribute from *Linked Data*, as mentioned in Section 2.2. The attribute from *Discogs* indicates how many people own an album and is probably a tendency of people to buy and brag about albums that have good reviews. The attribute with the lowest weight -0.442 is a number of reviews of an album on *Amazon*. As *Amazon* is a shop, it probably shows a tendency of people to complain about bad things and to not appreciate good things.

4 Conclusions

Apparently, we are not there yet with the Semantic Web. In theory, most of the features we used were already available in the Linked Data. In practice,

[6] `http://www.amazon.com/`
[7] `https://www.discogs.com/`
[8] `https://musicbrainz.org/doc/MusicBrainz_Database/Download`

Table 1. The attributes having coefficients in logistic regression model above 1. These coefficients were all positive, what means that the higher they are, the higher probability of a given album being a good one.

attribute	coefficient
review score from *Pitchfork* `pitchfork.com`	2.859
review score from *AllMusic* `www.allmusic.com`	2.437
review score from *Stylus* `www.stylusmagazine.com`	1.926
number of people owning an album according to *Discogs* `www.discogs.com`	1.465
review score from *Entertainment Weekly* `www.ew.com`	1.274
review score from *The Guardian* `www.theguardian.com`	1.096

they were not. The issues with the Linked Data discussed in the paper clearly suggests that even a very simple and crude solutions doing web scrapping can easily outperform solutions based on the Linked Data.

The presented solution consists of 641 lines of Python code and can classify correctly 296 out of 320 test albums, which we find to be quite a good result given a small amount of time invested and irregularities in the provided datasets. It is also worth to note, that the baseline solution was able to classify correctly 222 test albums (69.375%), so our solution offers quite an improvement.

References

1. Beek, W., Rietveld, L., Bazoobandi, H.R., Wielemaker, J., Schlobach, S.: LOD laundromat: a uniform way of publishing other peoples dirty data. In: The Semantic Web–ISWC 2014, pp. 213–228. Springer (2014)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: science, services and agents on the world wide web 7(3), 154–165 (2009)
3. Cox, D.R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society. Series B (Methodological) pp. 215–242 (1958)
4. Ławrynowicz, A., Potoniec, J.: Pattern based feature construction in semantic data mining. Int. J. on Sem. Web and Information Systems (IJSWIS) 10(1), 27–65 (2014)
5. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Proc. of the 12th ACM SIGKDD int. conf. on Knowledge discovery and data mining. pp. 935–940 (2006)
6. Potoniec, J., Ławrynowicz, A.: RMonto: Ontological extension to RapidMiner. In: Poster and Demo Session of the 10th Int. Semantic Web Conf. (2011)
7. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proc. of the 16th int. conf. on World Wide Web. pp. 697–706. ACM (2007)
8. Swartz, A.: Musicbrainz: A semantic web service. Intelligent Systems, IEEE 17(1), 76–77 (2002)
9. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM 57(10), 78–85 (2014)
10. Wikipedia: Template:album ratings, `https://en.wikipedia.org/w/index.php?title=Template:Album_ratings&oldid=670493671`, [Online; accessed 2016-03-07]