Exploiting Linked Open Data as Background Knowledge in Data Mining

Heiko Paulheim

University of Mannheim, Germany
Research Group Data and Web Science
heiko@informatik.uni-mannheim.de

Abstract. Many data mining problems can be solved better if they are augmented with additional background knowledge. This paper discusses a framework of adding background knowledge from Linked Open Data to a given data mining problem in a fully automatic, unsupervised manner. It introduces the *FeGeLOD* framework and its latest implementation, the *RapidMiner Linked Open Data extension.* We show the use of the approach in different problem domains and discuss current research directions.

Keywords: Data Mining, Background Knowledge, Linked Open Data, RapidMiner

1 Introduction

Data Mining is the process of identifying novel, valid, and interesting patterns in data [4]. Using background knowledge can help discovering those patterns, as well as finding completely new patterns that originate from combining the original data with additional data from different sources.

In the recent years, Linked Open Data [3] has grown to a large collection of open datasets following well-defined standards such as RDF. Using explicit semantics, that data is made interpretable by machines. This paper discusses a framework and use cases of using that large data collection as background knowledge in data mining, showing different use cases.

There are two principle strategies for using Linked Open Data for data mining:

1. Developing specialized mining methods for Linked Open Data. Examples include operators for rule learning algorithms, e.g., DL-Learner [9] or specialized kernel functions for support vector machines [11].
2. Pre-processing Linked Open Data so that it can be accessed with traditional (e.g., propositional) data mining methods (e.g., [8,15]).

This paper introduces a method that follows the second principle, i.e., a preprocessing strategy. The rationale is that such a strategy allows for re-using many existing data mining algorithms and tools, and is thus considered more versatile.

The rest of this paper is structured as follows. Section 2 introduces our theoretical framework, for which the current implementation is discussed in section 3. Section 4 discusses different example applications which use the framework introduced in this paper, i.e., text classification and interpreting statistics. We conclude with a review of current challenges in section 5, and a short summary.

2 Theoretical Framework

To augment a dataset with Linked Open Data, we propose a general pipeline comprising three steps:

1. First, entities in Linked Open Data have to be recognized that correspond entities in the original dataset. For example, in a dataset about cities, DBpedia[1] [10] or Geonames[2] URIs identifying those cities are added to the dataset.
2. Second, data about the entities in question is extracted using the previously identified URIs. This results in generating additional features and adding them the original dataset.
3. Since the previous step may, depending on the dataset and the strategies used, create an abundance of features, employing feature selection is often necessary before further processing the data with data mining algorithms.

Fig. 1 shows the three steps, using the example of a book sales dataset storing sales figures for individual books in book stores located in different cities. In the end, a set of additional features is added to the original dataset. With the help of these features, novel patterns might be discovered, such as certain books selling better in larger cities.

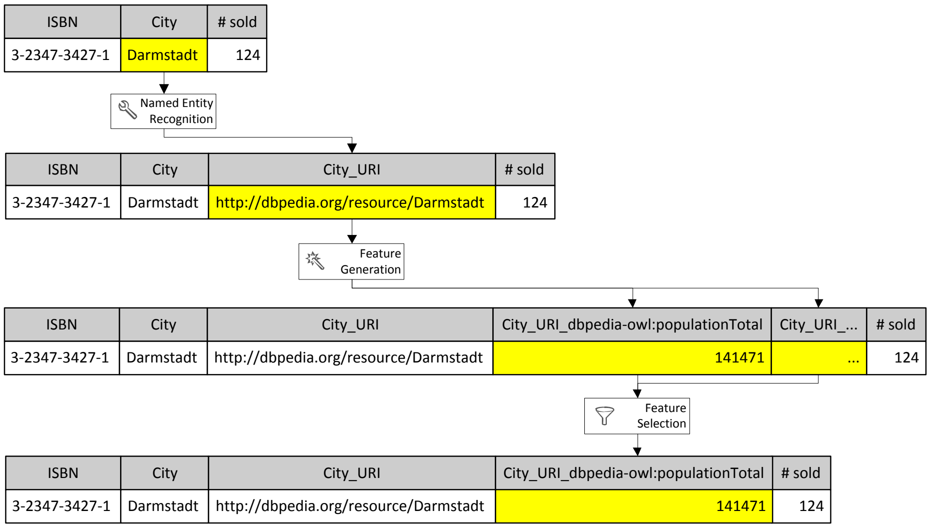
3 The RapidMiner Linked Open Data Extension

We have implemented the essential steps of our theoretical framework as an extension to *RapidMiner*[3], an open data mining platform[4] (except for feature selection, which is already covered by RapidMiner and other extensions).

For the entity linking step, different strategies are currently supported by our implementation:

- Creating links based on custom URI patterns, e.g., appending the string value of an attribute containing city names to a constant, such as `http://dbpedia.org/resource/`.

[1] `http://dbpedia.org`
[2] `http://www.geonames.org`
[3] `http://dws.informatik.uni-mannheim.de/en/research/rapidminer-lod-extension/`
[4] `http://rapid-i.com/content/view/181/`



ISBN	City	# sold
3-2347-3427-1	Darmstadt	124

ISBN	City	City_URI	# sold
3-2347-3427-1	Darmstadt	http://dbpedia.org/resource/Darmstadt	124

ISBN	City	City_URI	City_URI_dbpedia-owl:populationTotal	City_URI_...	# sold
3-2347-3427-1	Darmstadt	http://dbpedia.org/resource/Darmstadt	141471	...	124

ISBN	City	City_URI	City_URI_dbpedia-owl:populationTotal	# sold
3-2347-3427-1	Darmstadt	http://dbpedia.org/resource/Darmstadt	141471	124

Fig. 1. Abstract pipeline for augmenting datasets with additional knowledge from Linked Open Data [15].

- Discovering links by full text search with SPARQL statements.
- Using the DBpedia lookup service[5], optionally with type restrictions, and different disambiguation strategies.
- Using the DBpedia Spotlight service[6] [12] for text processing.

With respect to feature generation, different generators are supported by our implementation:

- Adding datatype properties as features. This generator includes heuristic guessing of appropriate attribute types (e.g., recognizing numerics and dates).
- Adding direct types as boolean features.
- Adding boolean or numeric features for incoming and outgoing relations.
- Adding boolean or numeric features for incoming and outgoing relations plus their type, i.e., using qualified relations.
- Adding features using custom SPARQL queries.

Details on those generators can be found in [15]. Fig. 2 shows an example RapidMiner process using operators of the Linked Open Data extension.

4 Example Applications

Background knowledge from Linked Open Data can be helpful in different tasks. In the following, we show examples from different domains, which have been built using either the RapidMiner Linked Open Data extension or one of its predecessors.

[5] http://lookup.dbpedia.org

[6] http://spotlight.dbpedia.org/

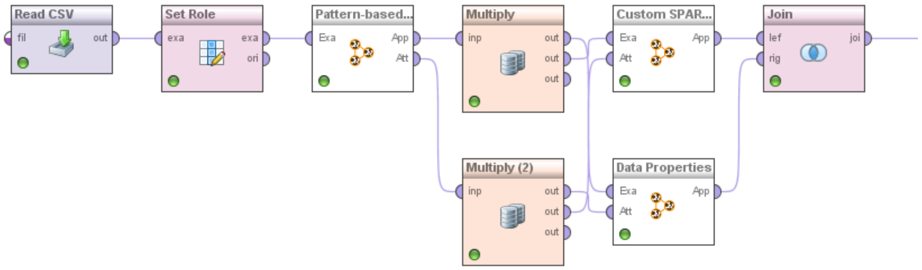


Fig. 2. An example RapidMiner process using the Linked Open Data extension.

4.1 Text Classification

In [5], a dataset of events extracted from Wikipedia has been introduced. The dataset contains events that are harvested from Wikipedia pages for years and months in different languages. Each event is represented by a time, a short text, and links to entities involved in the event (which are DBpedia links for the Wikipedia pages linked from the event text).

In order to further enrich the dataset, we have assigned classes to the events automatically. Since some of the events already have classes (since they are harvested from pages with a topical structure), we had a training and testing set for English language events available.

As features for the classification, we have used direct types and categories of the resources involved in an event. The rationale is that for example, sports events can be identified by athletes and/or stadiums being involved, while politics events can be identified by politicians being involed. Using only such binary features, we were able to achieve an accuracy of 80% for a problem comprising more ten different categories and a training and test set of 1,000 instances.

Furthermore, since we did not use any textual features, but only structured content from DBpedia, we were able to apply a model trained on the English dataset on event datasets in other languages as well. We have shown that the classification accuracy is still the same when applying the model trained on the English dataset to a set of events in German language [5].

A similar use case is the classification of social media texts, such as Tweets, for example for the use in emergency management applications [18]. In [2], we have used different textual features for identifying Tweets that talk about car accidents, being able to achieve a classification accuracy of 90%.

In that experiment, we had gathered training data using Tweets from a particular city. However, when using the trained model on data from another city, the classification accuracy dropped to 85%. The reason is that an overfitting effect occurs, e.g., the names of major streets are used as indicators for identifying Tweets about car accidents.

To avoid the overfitting, we used features from DBpedia, first preprocessing the Tweets with DBpedia spotlight, and then adding additional types and categories for the identified concepts, just like in the event classification experiment above. Using those abstract concepts (e.g., `dbpedia-owl:Road` instead of the name of a particular road) remedies the overfitting effect and keeps the accuracy up on the same level when applying a model learned on data from one city to data from a different one.

4.2 Explaining Statistics

In most data mining scenarios, there is already some data available. However, there are also cases where the amount of data is scarce. One typical example are statistics, where usually only one or a few target variables are produced, e.g., conducting a survey on the quality of living in different cities, or gathering data on drug abuse in different countries.

In most cases, people working with these statistics, such as journalists, are interested in finding reasons for the effects reported in those statistics. In [13], we have introduced the prototype tool *Explain-a-LOD*, which uses the pipeline discussed above for enriching statistics files. For example, for a statistical dataset on cities, Linked Open Data sources can provide relevant background knowledge such as population, climate, major companies, etc.

Having enriched the statistics at hand with background information, we analyze the generated features for correlation with the target variables, as well as perform rule learning to find more complex patterns. Fig. 3 depicts a screenshot of the tool. Further details and examples can be found in [13].

5 Challenges

The examples above have shown how Linked open Data provides additional value in different data mining problems. However, there are also some challenges to be addressed.

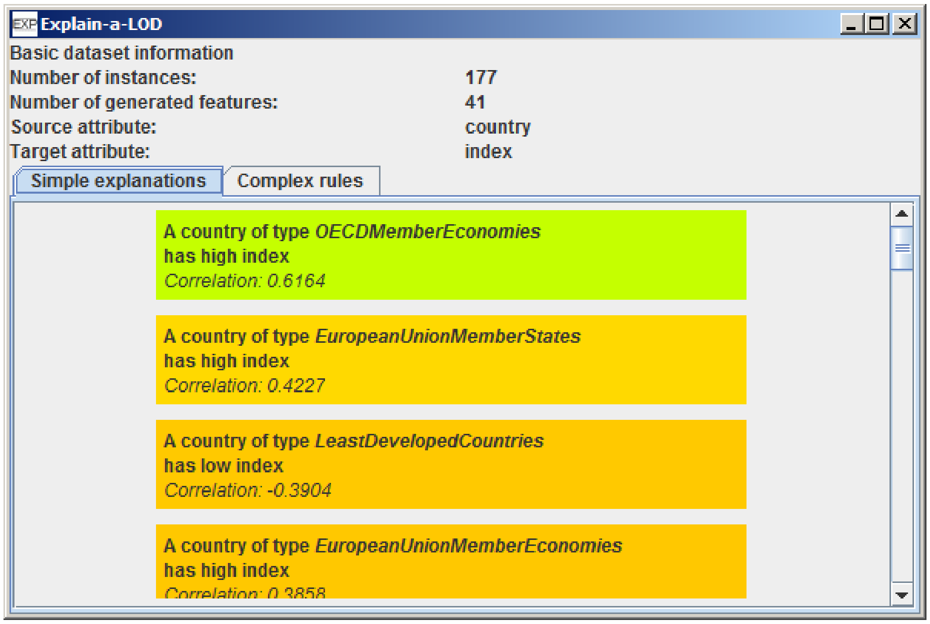
5.1 Dealing with the Variety of Linked Open Data

Despite Linked Open Data being built on well-defined standards, there are different ways to provide Linked Open Data. In its current version, the RapidMiner Linked Open Data extension exploits SPARQL endpoints. However, there are datasets which do not have SPARQL endpoints, but which could provide interesting background knowledge in many cases, e.g., Freebase[7] or OpenCyc[8]. For such datasets, different implementation of the generation algorithms are required.

Furthermore, there are non-standard SPARQL constructs, such as COUNT or the asterisk operator for computing the transitive closure [20] which are supported by some, but not all endpoints. Such constructs may help computing

[7] `http://www.freebase.com/`

[8] `http://www.cyc.com/platform/opencyc`



Basic dataset information

Number of instances:	177
Number of generated features:	41
Source attribute:	country
Target attribute:	index

A country of type *OECDMemberEconomies*
has high index
Correlation: 0.6164

A country of type *EuropeanUnionMemberStates*
has high index
Correlation: 0.4227

A country of type *LeastDevelopedCountries*
has low index
Correlation: -0.3904

A country of type *EuropeanUnionMemberEconomies*
has high index
Correlation: 0.3858

Fig. 3. Screenshot of the tool Explain-a-LOD for finding explaining patterns for statistics [13].

certain features in a more performant way. However, it is difficult to determine the features supported by a SPARQL endpoint automatically, in particular since vocabularies such as VoID do not provide means to describe technical details of SPARQL endpoints [1].

Besides technical variations, there are also different ways to represent data. For example, the current generator versions rely on data values being direct properties of the entities at hand. However, there are different cases, as this example from CORDIS[9] shows:

```
EU18931 a Funding .
EU18931 has-grant-value [
  has-amount 1300000 .
  has-unit-of-measure EUR .
]
```

A similar challenge exists for data using the data cube vocabulary [21], or time indexed data, which can exist in various fashions [17].

5.2 Discovering Datasets and Exploiting Links

While our approach is unsupervised in that the user does not need to know many details about the datasets to use, this does not hold for selecting the

[9] http://cordis.rkbexplorer.com/

datasets themselves. Since there are no universal means to find datasets that hold information on certain entities, this is difficult to circumvent, nevertheless, it would be a desirable feature of a fully automatic approach.

A possible way to deal with this problem could be to start with one dataset, e.g., DBpedia, for which entity recognition can be performed in high quality. From those datasets, links to other datasets (directly via `owl:sameAs`, or using link repositories such as *sameas.org*[10]) can be followed to successively add further datasets. In that case, SPARQL endpoints should be added automatically and dynamically, which, given only a URI, has been shown to be a difficult issue [16].

5.3 Improving Entity Linking

Entity linking is the first step in our pipeline. It should therefore work with high quality, since errors made at this step are carried over to later steps (e.g., extracting features from entities that do not correspond the entity meant in the original dataset, such as population figures for a different city).

While linking is not difficult for entities such as countries or major cities, it can become more difficult for other classes, such as universities or animals, as discussed in [15]. In particular when not including additional knowledge from the user (which would contradict the paradigm of an unsupervised approach), there are cases that are barely distinguishable, e.g., a set of hurricanes and a set of persons (both of which have person names as attributes). In particular in the absence of additional attributes and meaningful column headers, it is hardly possible to reliably find correct links.

5.4 Exploiting Semantics

Most datasets in Linked Open Data come with at least light-weight explicit semantics, i.e., they use a vocabulary that contains semantic information in the form of ontology statements. These can provide valuable information to an approach for automatically enriching a dataset.

Consider the case where direct types of entities are added as boolean features. Using the schema information, the features form a hierarchy, e.g., *African Island* $\subseteq$ *Island*. This hierarchy can be exploited, e.g., for improving feature selection, similar to the approach described in [7], which helps discovering only meaningful patterns and avoiding overfitting: if both *African Island* and *Island* have the same characteristics (e.g., they are highly correlated, or they have same information gain w.r.t. a target variable), we can prune the more specific variable without losing classification accuracy.

5.5 Combining Feature Creation and Selection

The approach discussed above proposes a pipeline of three strictly sequential steps. In particular, it first generates the whole set of possible features, which are then filtered in a subsequent step.

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

This kind of approach has certain limitations with respect to the amount of features that can be generated. In particular, we have not included a generator which generates features for all individuals linked to a resource for reasons of scalability. However, in a later step, it would turn out that most of those features are not useful, and they would be removed again in the next step. An extreme case is the generator for qualified relations, which, when combined with a deep class hierarchy such as YAGO [6], creates features such as $\exists location^{-1}.ArtSchoolsInParis$, which is *true* for only one instance (i.e., $Paris$), and false for all other cities.

An even more complex strategy for feature generation we have not pursued so far is the construction of new features from those already generated, e.g., the number of cinemas per inhabitants, where both number of cinemas and population are features from the Linked Open Data set.

To remedy those problems and arrive at more scalable approaches, it would be necessary to develop algorithms that combine feature selection and creation in a joint process. A straight forward approach could be to create features for a sample of the data first, determine the relevant features, and then create only those features for the rest of the dataset. Exploiting semantics, as discussed above, could lead to more sophisticated approaches.

5.6 Dealing with Dataset Coverage and Biases

Linked Open Data may be incomplete by design, i.e., following the open world assumption. In many cases, we ignore the open world assumption for creating data mining features. For example, a generator adding types to an instance adds the value `false` if the type is not present – which is not in line with the semantics of RDF using the open world assumption [19]. Similarly, adding a feature like *number of companies in the city*, which counts the corresponding entities in the Linked Open Dataset, neglects the open world assumption.

Some of these problems may be remedied by trying to detect and fill in missing information in a preprocessing step. For example, in [14], we have introduced an approach which has been shown to complete missing type information in DBpedia at very high quality.

A more subtle problem is the presence of biases in datasets. In the statistics use case discussed above, we have frequently observed explanations such as *The quality of living is high in cities where many music recordings were produced*, in particular when using DBpedia as a source of background knowledge. This is mainly an effect of a skewed distribution of data in DBpedia (and Wikipedia), which contains more information on popular culture in the western world, so the bottom line of this explanation is that cities in the western world have a high quality of living. However, such biases are difficult to detect automatically.

6 Conclusion

In this paper, we have introduced both a theoretical framework as well as an implementation for using Linked Open Data as background knowledge in data mining. The implementation uses the data mining toolkit RapidMiner, which allows for combining the Linked Open Data specific operators with various other operators for data processing and mining.

We have shown different use cases where background knowledge from Linked Open Data can improve the results and provide new insights that could not have been gained from the mere data without background knowledge.

While Linked Open Data provides a lot of opportunities, there are also a number of challenges to address, ranging from coping with the large variety of data representations within Linked Open Data to dealing with complex features in a scalable way. With this paper, we have sketched a research agenda by enumerating some of the most prevalent challenges. We are confident that this research agenda will lead to a set of approaches and tools that allow for novel data mining tools which have access to a large amount of knowledge and provide deeper insights into data.

Acknowledgements

In the past two years, several people have contributed to the conception and/or implementation of our framework and the RapidMiner Linked Open Data extension. The author would like to thank Johannes Fürnkranz, Raad Bahmani, Alexander Gabriel, and Simon Holthausen for their valuable contributions. The current team developing the extension includes Christian Bizer, Petar Ristoski, and Evgeny Mitichkin.

The author would further like to thank all the people developing the example applications discussed in this paper, including Daniel Hienert and Dennis Wegener at GESIS, and Axel Schulz at SAP Research.

References

1. Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets. In *Linked Data on the Web (LDOW2009)*, 2009.
2. Petar Ristoski Axel Schulz and Heiko Paulheim. I see a car crash: Real-time detection of small scale incidents in microblogs. In *ESWC 2013 Satellite Events: Revised Selected Papers*, 2013.
3. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
4. Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.
5. Daniel Hienert, Daniel Wegener, and Heiko Paulheim. Automatic classification and relationship extraction for multi-lingual and multi-granular events from Wikipedia. In *Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012)*, volume 902 of *CEUR-WS*, pages 1–10, 2012.
6. Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, (194):28–61, 2013.

7. Yoonjae Jeong and Sung-Hyon Myaeng. Feature selection using a semantic hierarchy for event recognition and type classification. In *The 6th International Joint Conference on Natural Language Processing*, 2013.
8. Venkata Narasimha Pavan Kappara, Ryutaro Ichise, and O.P. Vyas. Liddm: A data mining system for linked data. In *Workshop on Linked Data on the Web (LDOW2011)*, 2011.
9. Jens Lehmann. Dl-learner: Learning concepts in description logics. *Journal of Machine Learning Research*, 10:2639–2642, 2009.
10. Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2013.
11. Uta Lösch, Stephan Bloehdorn, and Achim Rettinger. Graph kernels for rdf data. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, pages 134–148, 2012.
12. Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.
13. Heiko Paulheim. Generating possible interpretations for statistics from linked open data. In *9th Extended Semantic Web Conference (ESWC)*, 2012.
14. Heiko Paulheim and Christian Bizer. Type inference on noisy rdf data. In *12th International Semantic Web Conference (ISWC)*, 2013.
15. Heiko Paulheim and Johannes Fürnkranz. Unsupervised Generation of Data Mining Features from Linked Open Data. In *International Conference on Web Intelligence, Mining, and Semantics (WIMS'12)*, 2012.
16. Heiko Paulheim and Sven Hertling. Discoverability of sparql endpoints in linked open data. In *Proceedings of the ISWC 2013 Posters & Demonstrations Track*, 2013.
17. Anisa Rula, Matteo Palmonari, Andreas Harth, Steffen Stadtmüller, and Andrea Maurino. On the diversity and availability of temporal information in linked open data. In *The Semantic Web – ISWC 2012*, pages 492–507. Springer, 2012.
18. Axel Schulz, Heiko Paulheim, and Florian Probst. Crisis information management in the web 3.0 age. In *9th International ISCRAM Conference*, 2012.
19. W3C. Resource Description Framework (RDF): Concepts and Abstract Syntax. `http://www.w3.org/TR/rdf-concepts/`, 2004.
20. W3C. SPARQL New Features and Rationale. `http://www.w3.org/TR/sparql-features/`, 2009.
21. W3C. RDF. `http://www.w3.org/TR/vocab-data-cube/`, 2013.