Data Mining with Background Knowledge from the Web

Heiko Paulheim, Petar Ristoski,
Evgeny Mitichkin, and Christian Bizer
University of Mannheim
Data and Web Science Group

Abstract

Many data mining problems can be solved better if more background knowledge is added: predictive models can become more accurate, and descriptive models can reveal more interesting findings. However, collecting and integrating background knowledge is tedious manual work. In this paper, we introduce the *RapidMiner Linked Open Data Extension*, which can extend a dataset at hand with additional attributes drawn from the Linked Open Data (LOD) cloud, a large collection of publicly available datasets on various topics. The extension contains operators for linking local data to open data in the LOD cloud, and for augmenting it with additional attributes. In a case study, we show that the prediction error of car fuel consumption can be reduced by 50% by adding additional attributes, e.g., describing the automobile layout and the car body configuration, from Linked Open Data.

1 Introduction

In many data mining problems, adding background knowledge leads to better results. For example, a product's sales figures for different regions can be predicted more accurately if detailed population figures (such as age structure and average purchasing power) are known. Furthermore, more expressive and interesting descriptive models can be mined the more background knowledge is added to the process.

However, collecting and integrating large amounts of background knowledge can be a labor intensive task. Moreover, in most cases, only a small fraction of that background knowledge will be actually used in a predictive or descriptive model, but it is hard to pinpoint the relevant parts in advance. Furthermore, variables involved in unexpected findings are easily overseen, since

assumptions about interrelations in the application domain lead the user when selecting additional attribute, i.e., he or she will be subject to a selection bias. To overcome these shortcomings, open data sources on the web, providing knowledge in a large variety of domains, are a valuable source of background knowledge.

In the recent years, the *Linked Open Data* (LOD) paradigm has been established, which allows for publishing interlinked datasets using machine interpretable semantics. Datasets such as *DBpedia*[1] [5] and *YAGO*[2] [16], which translate structured data in Wikipedia, such as infoboxes and categories, into machine interpretable form, or *Freebase*[3] (which forms the base of Google's Knowledge Graph) are prominent examples of such open datasets. Apart from the scientific community, companies and government agencies have also taken up the idea of linked data to publish their data in a machine-processable way, using the LOD paradigm.

In this paper, we introduce the *RapidMiner Linked Open Data (LOD) Extension*[4], which allows for using such public open datasets as background knowledge in data mining tasks. The extension contains operators which establish links from local databases to those public datasets, and which extend the local databases with attributes from the open datasets. Such extended databases can then be used for various data mining tasks.

The rest of this paper is structured as follows. In section 2, we introduce the basic concepts of Linked Open Data, and throw a glance at the datasets available. In section 3, we give a brief overview of the operators in the RapidMiner LOD Extension, and in section 4, we discuss an example use case. Section 5 gives an overview of related approaches and RapidMiner extensions. We conclude with a short summary and an outlook on planned future developments.

2 Linked Open Data in a Nutshell

The Linked Open Data (LOD) paradigm describes a set of best practices for publishing machine-interpretable data online. A central notion of Linked Open Data are *resources*. Resources form a graph, connected by properties. Each resource is identified by a *Uniform Resource Identifiers (URI)*. For Linked Open Data, HTTP URIs are used, which have to be *dereferencable*, i.e., when issuing a HTTP request to the URI, data about the resource should be returned using a standard such as the *Resource Description Framework (RDF)*[5]. This

[1] `http://dbpedia.org/`
[2] `http://www.mpi-inf.mpg.de/yago-naga/yago/`
[3] `http://www.freebase.com/`
[4] `http://dws.informatik.uni-mannheim.de/en/research/rapidminer-lod-extension/`
[5] `http://www.w3.org/RDF/`

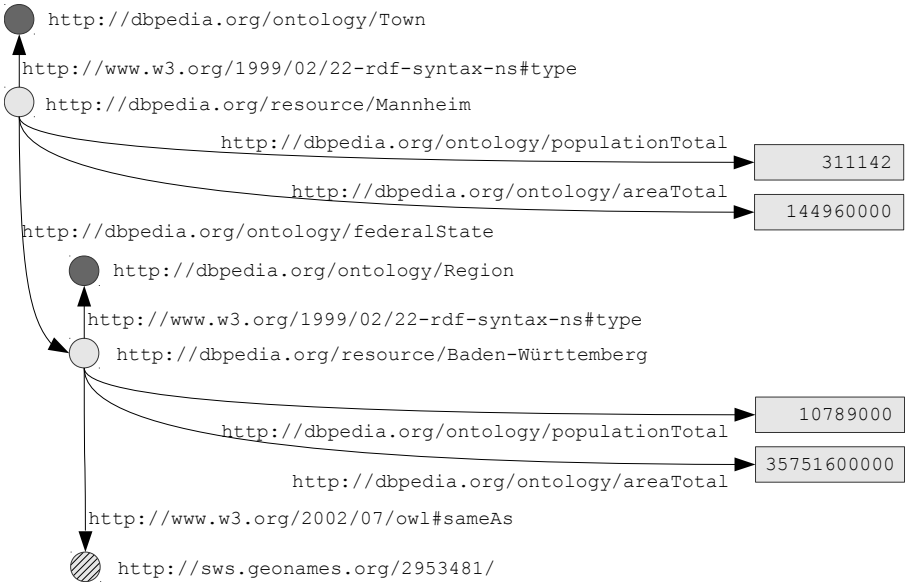


Figure 1: An example excerpt of DBpedia and a link to an external dataset. The light gray circles and rectangles depict resources and values defined in DBpedia, the dark gray circles are classes in the DBpedia ontology, and the hatched circle is a resource in a different dataset, i.e., *GeoNames*.

allows agents to traverse the knowledge graph in Linked Open Data within and across different datasets, and discover information on demand.

In each dataset, resources are described using a *schema* or *ontology*. With that schema, resources are assigned *direct types* (e.g., *Mannheim* is a *Town*), and statements about resources are made with defined *properties* (e.g., stating the population of Mannheim, and that it is located in the federal state of Baden-Württemberg). The languages used to define schemas are RDF Schema[6] and OWL[7], and schemas are usually published according to LOD principles as well, using dereferencable URIs for defining types and properties.

Links between datasets are an essential ingredient to *Linked* Open Data. Most often, those links specify identity relations between resources defined in different datasets, which allows an agent to look up a resource in other datasets to obtain additional information. For example, one dataset may state the population and area of Mannheim, and contain a link to another datset, which states which companies and organizations are located in Mannheim.

[6] `http://www.w3.org/TR/rdf-schema/`
[7] `http://www.w3.org/TR/owl-overview/`

With those links, Linked Open Data constitutes a cloud of interconnected datasets instead of isolated data islands, as depicted in Fig. 2.

Figure 1 depicts an example excerpt of the DBpedia dataset. The population and area of Mannheim and Baden-Württemberg are depicted, and both are connected via a property, as well as assigned types in the DBpedia ontology. Furthermore, a link to an external dataset (Geonames) exists, where additional information about Baden-Württemberg can be found. An agent consuming the data, which knows the URI for Mannheim, can follow those links within and across datasets to gather information about Mannheim. As shown in the figure, all types and predicates are identified by (dereferencable) URIs as well. With the help of those definitions, agents can look up definitions of vocabulary terms, which eases the data consumption.

Besides following links, many datasets also offer sophisticated query interfaces via *SPARQL*[8], a query language for RDF data which is similar to SQL for relational databases.

Figure 2 depicts an overview of Linked Open Data sets and their interlinks. There are currently around 1,000 datasets published as Linked Open Data, comprising various domains, such as government and geographic data (e.g., statistics published by the World Bank or EU agencies such as Eurostat), life sciences (e.g., databases about proteins or drug interactions) as well as cross-domain knowledge bases such as *DBpedia* and *Freebase*.

3 Operators in the Extension

A typical process using the Linked Open Data extension comprises four steps. First, the local dataset at hand (e.g., sales figures) are *linked* to an open dataset. This means that for each instance in the dataset, one or more *links* are generated, which point to *resources* describing the instance, e.g., for a dataset containing information sales regions, resources in LOD datasets are identified which represent those regions, and their URIs are included as additional attributes in the dataset.

In a second step, *attributes are extracted* from that dataset. The corresponding operators use the links created in the first step to retrieve information about the instances and include that information in the form of additional attributes.

Since the number of attributes generated can fairly large, depending on the dataset and extraction method used, *feature subset selection* is often performed as a third step. Finally, the actual *data analysis* is carried out on the extended dataset, i.e., building a predictive or descriptive model. That last step is

[8] `http://www.w3.org/TR/sparql11-overview/`

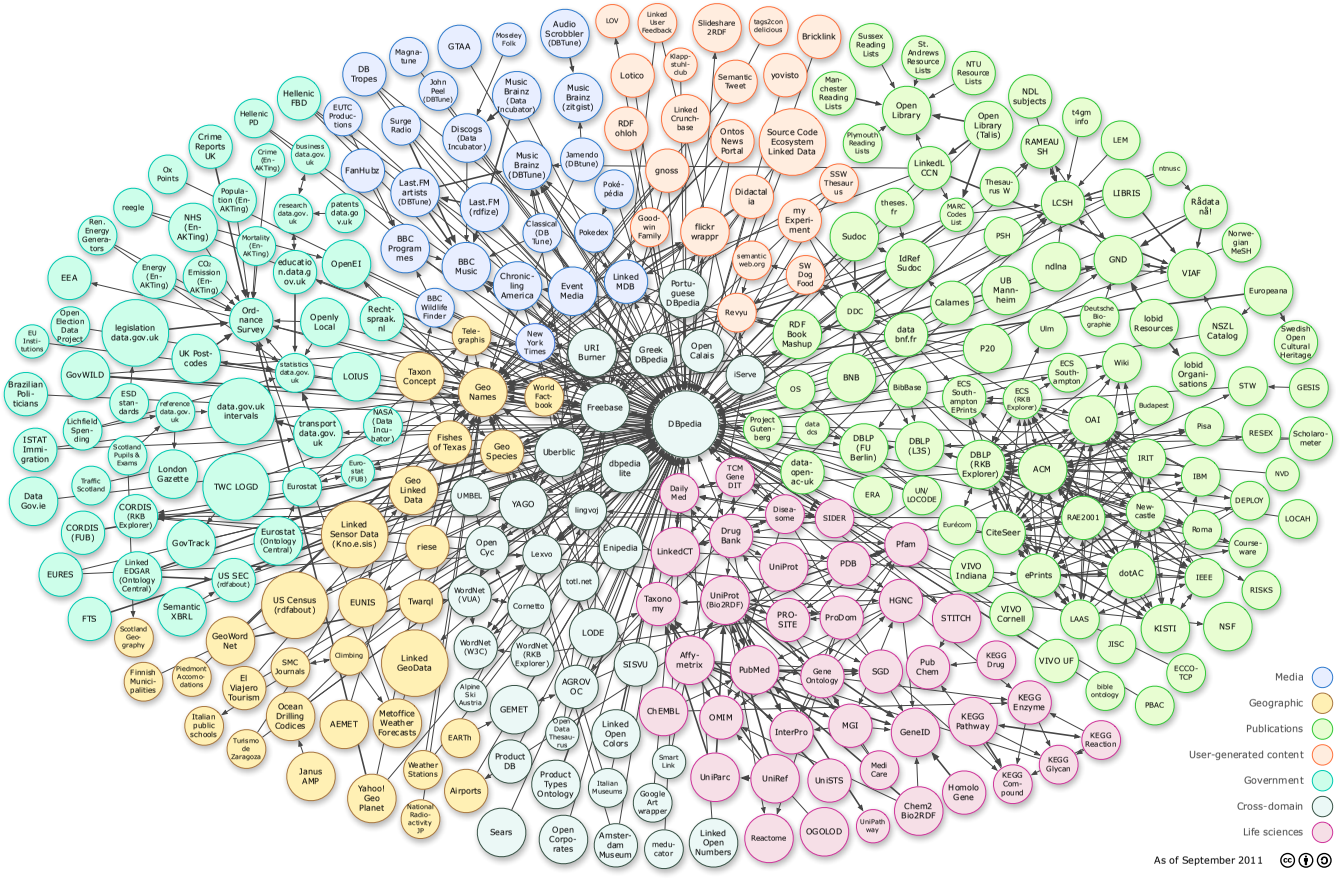


Figure 2: Diagram depicting the LOD cloud, as of September 2011. Each circle in the diagram depicts a dataset.

usually performed with standard RapidMiner operators. In the following, we will introduce the individual operators in more detail.

3.1 Operators for Linking

The first step of using Linked Open Data together with local data is to establish a links to a dataset in Linked Open Data. A linking operator creates a new column with a URI (see above) for each instance to be linked. There are different linking operators available in the Linked Open Data extension:

- The *pattern-based linker* creates URIs based on a string pattern. If the pattern a dataset uses for constructing its URIs is known, this is the fastest and most accurate way to construct URIs. For example, the *RDF Book Mashup* dataset uses a URI pattern for books which is based on the ISBN.[9]

- The *label-based linker* searches for resources whose label is similar to an attribute in the local dataset, e.g., the product name. It can only be used on datasets providing a SPARQL interface and is slower than the pattern-based linker, but can be applied if the link patterns are not known, or cannot be constructed automatically (e.g., if they use a dataset-internal ID).

- The *Lookup linker* uses a specific search interface[10] for the *DBpedia* dataset. It also finds resources by alternative names (e.g., *NYC* or *NY City* for *New York City*). For DBpedia, it usually provides the best accuracy.

- The *SameAs linker* can be used to follow links from one dataset to another. Since many datasets link to DBpedia, it is typically combined with the Lookup linker, which first establishes links to DBpedia at high accuracy. The resources identified by the Lookup linker can then be exploited to discover further resources in other datasets. In Fig. 1, the Lookup linker would link the local database to DBpedia, the SameAs linker would follow the link to the Geonames dataset, and subsequent operators could then read information from that dataset.

Furthermore, the *Spotlight linker* exists[11], which identifies multiple DBpedia resources in a longer text [6], such as a product description, and is typically used for text mining tasks.

[9] In cases where additional processing is required, such as removing dashes in an ISBN, the operator may be combined with the built-in *Generate Attributes* operator, which can perform such operations.

[10] `http://lookup.dbpedia.org/`

[11] `http://spotlight.dbpedia.org/`

3.2 Operators for Attribute Generation

For creating attributes from Linked Open Data sources, different strategies are implemented in the extension's operators:

- The *Direct Types* generator extracts all types of a linked resource. For datasets such as YAGO (see above), those types are often very informative, for example, products may have concise types such as *Smartphone* or *AndroidDevice*.

- The *Datatype Properties* generator extracts all datatype properties, i.e., numerical and date information (such as the price and release date of products).

- The *Relations* generator creates a binary or a numeric attribute for each property that connects a resource to other resource. For example, if a product has won one or more awards, an *award* attribute would be generated which captures that information.

- The *Qualified Relations* generator also generates binary or numeric attributes for properties, but takes the type of the related resource into account. For example, an attribute stating that the manufacturer of a product is a German company would be created.

- The *Specific Relations* generator creates features for a user-specified relation, such as Wikipedia categories included in DBpedia.

In addition to those automatic generators, it is also possible to control the attribute generation process in a more fine-grained manner by issuing specific SPARQL queries using the *Custom SPARQL* generator.

3.3 Operators for Feature Subset Selection

All standard methods for feature subset selection can be used in conjunction with the RapidMiner Linked Open Data extension, as well as operators from the *Feature Subset Selection extension* [15]. Furthermore, the Linked Open Data extension provides the *Simple Hierarchy Filter*, which exploits the schema information of a Linked Open Data source, and often achieves a better compression of the feature set than standard, non-hierarchical operators, without losing valuable features [14].

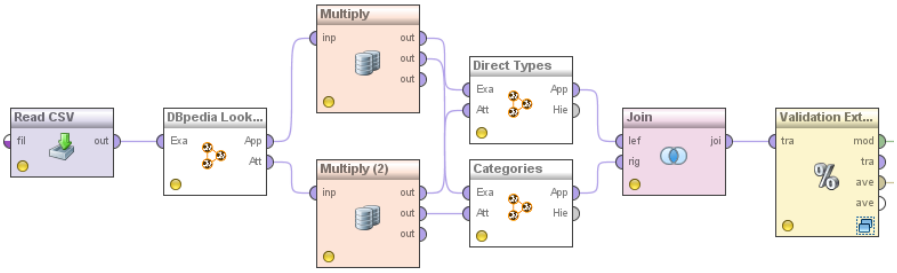


Figure 3: Example process using the Linked Open Data extension for extracting both direct types as well as Wikipedia categories from DBpedia.

4 Example Use Case

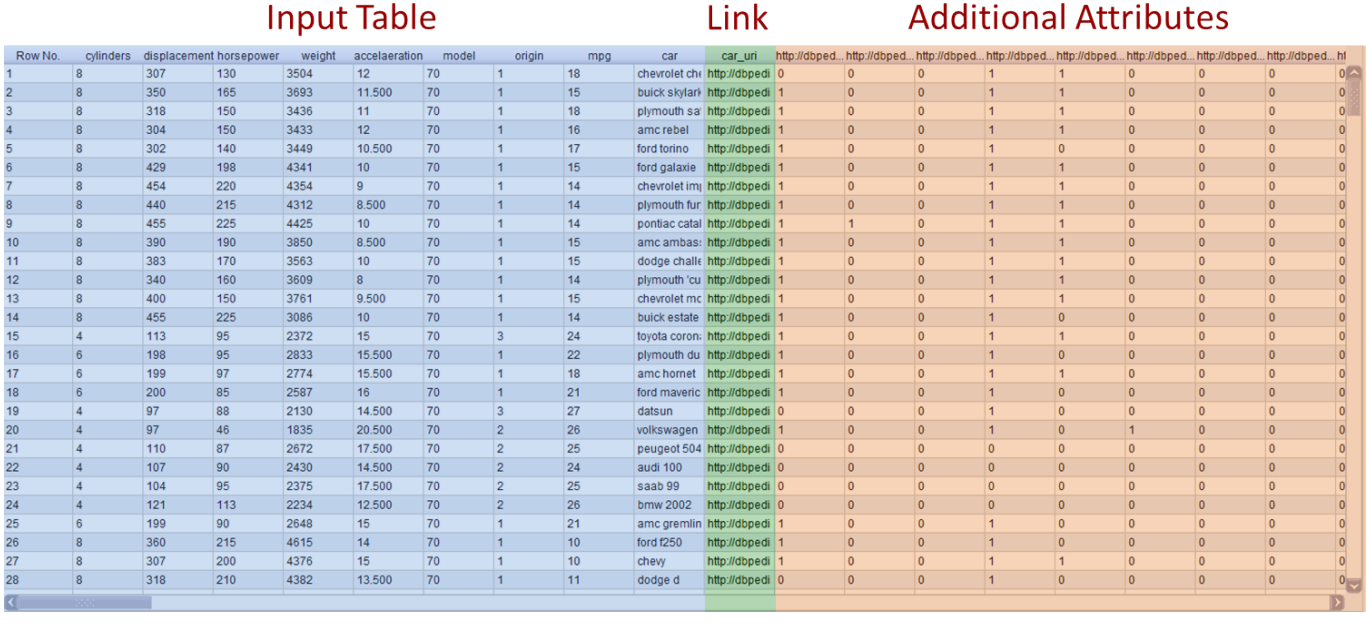
In a running example, we use the *Auto MPG* data set[12], a dataset that captures different characteristics of cars (such as cyclinders, transmission, horsepower), and the target is to predict the fuel consumption in Miles per Gallon (MPG) [12]. The original dataset contains 398 cars, each having a name, seven data attributes, and the MPG target attribute. The goal is to predict the fuel consumption from the characteristics of the car, i.e., a regression model hsa to be learned.

Using our extension, we add different new attributes from the DBpedia dataset, as shown in Fig. 3. First, we link each car to its corresponding DBpedia resource using the DBpedia Lookup linker. After manually correcting typos in the dataset, we were able to link all instances to DBpedia. Then, we extract both direct types as well as Wikipedia categories in two parallel operators, and join the datasets for running the prediction. The direct types operator adds 264 types in total (e.g., *Convertibles*, *Coupes*, or *Honda Vehicles*), for the categories (retrieved using the *Specific Relations* operator), another 199 are added (e.g., *1960s Automobiles*, *Cars from Turkey*, or *Rally Cars*).[13]. Figure 4 depicts the extended dataset.

For the prediction of the MPG attribute, we used Linear Regression, as well as the M5Rules algorithm [2]. The results are depicted in table 1. It can be observed that the best results can be achieved when combining both the attributes derived from direct types and categories. In general, M5Rules outperforms Linear Regression on the dataset, and the additional attributes lead to a relative error only half as large as on the original data. It is further

[12] `http://archive.ics.uci.edu/ml/datasets/Auto+MPG`

[13] The datasets and RapidMiner processes can be found online at `http://dws.informatik.uni-mannheim.de/en/research/rapidminer-lod-extension/rapid-miner-lod-extension-example-predicting-the-fuel-consumption-of-cars`



Row No.	cylinders	displacement	horsepower	weight	accelaeration	model	origin	mpg	car	car_uri	http://dbped...	http://dbped...	http://dbped...	http://dbped...	http://dbped...	http://dbped...	http://dbped...	http://dbped...	ht
1	8	307	130	3504	12	70	1	18	chevrolet che	http://dbpedi	0	0	0	1	1	0	0	0	0
2	8	350	165	3693	11.500	70	1	15	buick skylark	http://dbpedi	1	0	0	1	1	0	0	0	0
3	8	318	150	3436	11	70	1	18	plymouth sa	http://dbpedi	1	0	0	1	1	0	0	0	0
4	8	304	150	3433	12	70	1	16	amc rebel	http://dbpedi	1	0	0	1	1	0	0	0	0
5	8	302	140	3449	10.500	70	1	17	ford torino	http://dbpedi	1	0	0	1	0	0	0	0	0
6	8	429	198	4341	10	70	1	15	ford galaxie	http://dbpedi	1	0	0	1	1	0	0	0	0
7	8	454	220	4354	9	70	1	14	chevrolet imp	http://dbpedi	1	0	0	1	1	0	0	0	0
8	8	440	215	4312	8.500	70	1	14	plymouth fur	http://dbpedi	1	0	0	1	1	0	0	0	0
9	8	455	225	4425	10	70	1	14	pontiac catal	http://dbpedi	1	1	0	1	1	0	0	0	0
10	8	390	190	3850	8.500	70	1	15	amc ambass	http://dbpedi	1	0	0	1	1	0	0	0	0
11	8	383	170	3563	10	70	1	15	dodge challe	http://dbpedi	1	0	0	1	1	0	0	0	0
12	8	340	160	3609	8	70	1	14	plymouth 'cu	http://dbpedi	1	0	0	1	1	0	0	0	0
13	8	400	150	3761	9.500	70	1	15	chevrolet mc	http://dbpedi	1	0	0	1	1	0	0	0	0
14	8	455	225	3086	10	70	1	14	buick estate	http://dbpedi	1	0	0	1	0	0	0	0	0
15	4	113	95	2372	15	70	3	24	toyota coron	http://dbpedi	1	0	0	1	1	0	0	0	0
16	6	198	95	2833	15.500	70	1	22	plymouth du	http://dbpedi	1	0	0	1	0	0	0	0	0
17	6	199	97	2774	15.500	70	1	18	amc hornet	http://dbpedi	1	0	0	1	1	0	0	0	0
18	6	200	85	2587	16	70	1	21	ford maveric	http://dbpedi	1	0	0	1	0	0	0	0	0
19	4	97	88	2130	14.500	70	3	27	datsun	http://dbpedi	0	0	0	1	0	0	0	0	0
20	4	97	46	1835	20.500	70	2	26	volkswagen	http://dbpedi	1	0	0	1	0	1	0	0	0
21	4	110	87	2672	17.500	70	2	25	peugeot 504	http://dbpedi	0	0	0	0	0	0	0	0	0
22	4	107	90	2430	14.500	70	2	24	audi 100	http://dbpedi	0	0	0	0	0	0	0	0	0
23	4	104	95	2375	17.500	70	2	25	saab 99	http://dbpedi	0	0	0	0	0	0	0	0	0
24	4	121	113	2234	12.500	70	2	26	bmw 2002	http://dbpedi	0	0	0	0	0	0	0	0	0
25	6	199	90	2648	15	70	1	21	amc gremlin	http://dbpedi	1	0	0	1	0	0	0	0	0
26	8	360	215	4615	14	70	1	10	ford f250	http://dbpedi	1	0	0	1	0	0	0	0	0
27	8	307	200	4376	15	70	1	10	chevy	http://dbpedi	1	0	0	1	1	0	0	0	0
28	8	318	210	4382	13.500	70	1	11	dodge d	http://dbpedi	0	0	0	1	0	0	0	0	0

Figure 4: Extension of the Auto MPG dataset with a link to DBpedia and additional attributes.

Attribute set	Lin. Regression		M5Rules	
	RMSE	RE	RMSE	RE
Original	3.359	0.118	2.859	0.088
Original + direct types	3.334	0.117	2.835	0.091
Original + categories	4.474	0.144	2.926	0.090
Original + direct types + categories	2.551	0.088	**1.574**	**0.042**

Table 1: Regression results for the Auto MPG dataset, reporting both the Root Mean Squared Error (RMSE) as well as the Relative Error (RE) for Linear Regression and M5Rules.

remarkable that only the combination of attributes leads to that significant improvement, while direct types and categories alone have only little influence on the results, and may even lead to worse results.

Table 2 depicts the ten attributes that have the strongest correlation with the target attribute (i.e., MPG). It can be observed that for this dataset, some of the original attributes have a strong correlation (although the original attribute *acceleration* is not among the top 10). On the other hand, there are quite a few new attributes that are helpful for the prediction.

The new attributes also provide insights that are not possible from the original dataset alone. For example, UK cars obviously have a lower consumption than others (while the *origin* attribute contained in the original dataset only differentiates between America, Europe, and Asia). Front-wheel-drive cars have a lower consumption than rear-wheel-drive ones (the corresponding category being positively negatively with MPG at a level of 0.411), mostly due to the fact that they are lighter. Furthermore, a correlation with the car's design can be observed (e.g., hatchbacks having a lower consumption than station wagons).

At first glance, a slightly surprising finding is that according to the attributes, rally cars have a low consumption. However, when looking at the corresponding category in Wikipedia[14], it contains a lot of ordinary cars, in particular many smaller cars, like the *Ford Ka*, the *Peugeot 205*, or even the *Mini*, which have a lower fuel consumption than larger cars, such as SUVs. Furthermore, rally cars are often optimized for low weight, which leads to a lower consumption.

5 Related Work

The use of Linked Open Data in data mining has been proposed before, and implementations as RapidMiner extensions as well as proprietary toolkits exist.

[14] `http://en.wikipedia.org/wiki/Category:Rally_cars`

Attribute	Source	Correlation
weight	original	-0.854
displacement	original	-0.824
cylinders	original	-0.789
horsepower	original	-0.776
origin	original	0.606
model	original	0.544
Road vehicles manufactured in the UK	categories	0.462
Rally cars	categories	0.461
Front-wheel-drive vehicles	categories	0.459
Hatchbacks	categories	0.458

Table 2: Top ten attributes most strongly correlated with the target (MPG) in the overall dataset. The top ten attributes are either contained in the original dataset, or created from Wikipedia categories. None of the attributes generated from direct types are among the top ten.

The direct predecessor of the RapidMiner LOD extension is the *FeGeLOD* toolkit [10], a data preprocessing toolkit based on the *Weka* platform [1], which contains basic versions of the operators offered by the LOD extension.

Different means to mine data in Linked Open Data sets have been proposed, e.g., an extension for RapidMiner [7], as well as standalone systems like *LiDDM* [3]. In those systems, data can be imported from public SPARQL endpoints using custom queries, but no means to join that data with local data are given.

Similarly, the *RapidMiner SemWeb Extension* [4] is an extension for importing RDF from local files into RapidMiner, using custom SPARQL queries. As discussed above, RDF is a general graph format, which leads to the problem of set-valued features when transforming the data into the relational form used in RapidMiner. To cope with that issue, the extension provides different operators to transform the set-valued data into a lower-dimensional projection, which can be processed by standard RapidMiner operators.

Linked Open Data may also be loaded with the *RMOnto* [11] extension, which is similar to the SemWeb extension, but comes with a set of tailored relational learning algorithms and kernel functions. Together, these form a powerful package of operators, but it is difficult to combine them with built-in RapidMiner operators, as well as operators from other extensions.

All of those approaches miss a functionality to link local data to remote Linked Open Data, as in the use case discussed in section 4. Furthermore, they often require expert knowledge on Semantic Web technology, e.g., for formulating custom SPARQL queries, which is not necessary for our extension.

6 Conclusion and Future Work

In this paper, we have introduced the RapidMiner Linked Open Data extension. It provides a set of operators for augmenting existing datasets with additional attributes from open data sources, which often leads to better predictive and descriptive models. In this paper, we have shown the example of predicting the fuel consumption of cars, showing that the prediction error can be reduced by 50%. Other past usages of the extension include, e.g., the analysis of high unemployment regions in the EU [13], or the analysis of corruption statistics [8].

There are different directions of research that are currently pursued in order to improve the extension. Besides developing new algorithms for the functionality already included (e.g., for linking and feature selection), there are also some new functionalities currently being investigated.

First, we are exploring mechanisms that traverse links automatically, combining information *different* datasets. While this is straight forward for some datasets, the large representional variety in the LOD cloud makes the development of *general* operators a challenging task [9].

By traversing such links and combining data from different places, duplicate attributes may be created, e.g., the population of a city may be specified in different datasets. To improve the overall quality of the extended dataset, *schema matching* and *data fusion* capabilities are currently developed to be integrated into the extension.

Finally, the current implementation is rather strict in its three phases, i.e., it generates attributes first and filters them later. Depending on the generation strategy used, this can lead to a large number of features being generated only to be discarded in the subsequent step. To avoid that overhead and improve the performance, we are working on mechanisms that decide on the utility of attribute already during creation and are capable of stopping the generation of attributes earlier if they seem to be useless.

In summary, adding attributes from open data sources is an interesting option for increasing the quality of both predictive and descriptive models. The RapidMiner Linked Open Data extension provides operators that allow for adding such attributes in an automatic, unsupervised manner.

Acknowledgements

The work presented in this paper has been partly funded by the German Research Foundation (DFG) under grant number PA 2373/1-1 (Mine@LOD).

References

[1] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[2] Geoffrey Holmes, Mark Hall, and Eibe Prank. *Generating rule sets from model trees*. Springer, 1999.

[3] 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.

[4] Mansoor Ahmed Khan, Gunnar Aastrand Grimnes, and Andreas Dengel. Two pre-processing operators for improved learning from semanticweb data. In *First RapidMiner Community Meeting And Conference (RCOMM 2010)*, volume 20, 2010.

[5] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sren Auer, and Christian Bizer. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 2013.

[6] 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*, 2011.

[7] Andreas Nolle, German Nemirovski, A Sicilia, and J Pleguezuelos. An approach for accessing linked open data for data mining purposes. In *Proceedings of RapidMiner Community Meeting and Conference (RCOMM 2013), Porto*, 2013.

[8] Heiko Paulheim. Generating possible interpretations for statistics from linked open data. In *9th Extended Semantic Web Conference (ESWC)*, 2012.

[9] Heiko Paulheim. Exploiting linked open data as background knowledge in data mining. In *Workshop on Data Mining on Linked Open Data*, 2013.

[10] 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.

[11] Jedrzej Potoniec and Agnieszka Lawrynowicz. Rmonto: ontological extension to rapidminer. In *10th International Semantic Web Conference*, 2011.

[12] J. Ross Quinlan. Combining instance-based and model-based learning. In *ICML*, page 236, 1993.

[13] Petar Ristoski and Heiko Paulheim. Analyzing statistics with background knowledge from linked open data. In *Workshop on Semantic Statistics*, 2013.

[14] Petar Ristoski and Heiko Paulheim. Feature selection in hierarchical feature spaces. In *Discovery Science*, 2014.

[15] Benjamin Schowe. Feature selection for high-dimensional data with rapidminer. In *Proceedings of the 2nd RapidMiner Community Meeting And Conference (RCOMM 2011), Aachen*, 2011.

[16] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 697–706. ACM, 2007.