Collecting, Integrating, Enriching and Republishing Open City Data as Linked Data*

Stefan Bischof[1,2], Christoph Martin[2], Axel Polleres[2], and Patrik Schneider[2,3]

[1] Siemens AG Österreich, Vienna, Austria
[2] Vienna University of Economics and Business, Vienna, Austria
[3] Vienna University of Technology, Vienna, Austria

Abstract. Access to high quality and recent data is crucial both for decision makers in cities as well as for the public. Likewise, infrastructure providers could offer more tailored solutions to cities based on such data. However, even though there are many data sets containing relevant indicators about cities available as open data, it is cumbersome to integrate and analyze them, since the collection is still a manual process and the sources are not connected to each other upfront. Further, disjoint indicators and cities across the available data sources lead to a large proportion of missing values when integrating these sources. In this paper we present a platform for collecting, integrating, and enriching open data about cities in a reusable and comparable manner: we have integrated various open data sources and present approaches for predicting missing values, where we use standard regression methods in combination with principal component analysis (PCA) to improve quality and amount of predicted values. Since indicators and cities only have partial overlaps across data sets, we particularly focus on predicting indicator values across data sets, where we extend, adapt, and evaluate our prediction model for this particular purpose: as a ”side product” we learn ontology mappings (simple equations and sub-properties) for pairs of indicators from different data sets. Finally, we republish the integrated and predicted values as linked open data.

1 Introduction

Nowadays governments have large collections of data available for decision support. Public administrations use these data collections for backing their decisions and policies, and to compare themselves to other cities, and likewise infrastructure providers like Siemens could offer more tailored solutions to cities based on this data. Having access to high quality and current data is crucial to advance on these goals. Studies like the Green City Index [6] which assess and compare the performance of cities are helpful, in particular for public awareness. However, these documents are outdated soon after publication and reusing or analyzing the evolution of their underlying data is difficult. To improve this situation, we need regularly updated data stores which provide a consolidated, up-to-date view on relevant open data sources for such studies.

Even though there are many relevant data sources which contain quantitative indicators, e.g., population, about cities available as *open data*, it is still cumbersome to

* Compared to an informal, preliminary version of this paper presented at the Know@LOD 2015 workshop, Section 5, 6, and 8 are entirely new, plus more data sources have been integrated.

collect, clean, integrate, and analyze data from these sources: obstacles include different indicator specifications, different languages, formats, and units. Example sources of city data include DBpedia or the Urban Audit data set included in Eurostat; Urban Audit (http://ec.europa.eu/eurostat/web/cities/) for example, provides over 250 indicators on several domains for 258 European cities. Furthermore, several larger cities provide data on their own open data portals, e.g., London, Berlin, or Vienna.[4] Data is published in different formats such as RDF, XML, CSV, XLS, or just as HTML tables. The specifications of the individual data fields – (i) how indicators are defined and (ii) how they have been collected – are often implicit in textual descriptions only and have to be processed manually for understanding.

Moreover, data sources like Urban Audit cover many cities and indicators, but show a large ratio of *missing values* in their data sets. The impact of missing values is even aggravated when combining different data sets, since there is a fair amount of disjoint cities and indicators across those data sets, which makes them hard to integrate. Our assumption though – inspired also by works that suspect the existence of quantitative models behind the working, growth, and scaling of cities [1] – is that most indicators in such a scoped domain have their own structure and dependencies, from which we can build prediction models:[5] we evaluate different standard regression methods to choose the best fitting model to predict missing indicator values. We follow two approaches for computing such predictions. The first approach is based on a selection of "relevant" indicators as predictors for a target indicator. The second approach constructs the principal components (PCs) of the "completed" data sets (missing values are replaced with "neutral" values [21]), which are then used as predictors. We also compare both approaches according to their performance, prediction accuracy, and coverage (the number of possible predictions). We then extend the second approach for cross data set prediction, in case of a large disjointness of indicators and cities.

Contributions and Structure. Our concrete contributions are:

- We analyze and integrate several data sources including DBpedia, Urban Audit, USCCDB, and the UNSD Demographic and Social Statistics;
- We provide a system architecture for an "Open City Data Pipeline" including a crawler, wrappers, ontology-based integration, and data access components;
- We evaluate two prediction approaches for filling-in missing values, combining different standard regression methods and PCs to maximize prediction accuracy;
- We develop an approach for cross data set prediction and discuss its performance;
- We present an approach for learning mappings of indicators between data sets;
- We republish the integrated and predicated values as linked open data (LOD).

Section 2 describes the imported data sources and the challenges arising when processing/integrating their data. Section 3 presents an overview of the Open City Data Pipeline and a lightweight extensible ontology used therein. In Section 4 and 5 we explain the approaches developed for predicting missing values as well as the corresponding evaluation of their performance. Section 6 presents our ontology mapping learning approach. Our

[4] http://data.london.gov.uk/, http://daten.berlin.de/, and http://data.wien.gv.at/

[5] We refer to "predicting" instead of "imputing" values when we mean finding suitable approximation models to predict indicators values for cities and temporal contexts where they are not (yet) available. These predictions may (not) be confirmed, if additional data becomes available.

LOD interface to republish the integrated and predicted data is documented in Section 7. In Section 8 we discuss the use of Semantic Technologies and the lessons learnt from our application. Section 9 concludes with several possible future extensions.

2 Data Sources

The Open City Data Pipelines database contains data ranging from the years 1990 to 2014, but most of the data concerns the years after 2000. Not every indicator is covered over all years, where the highest overlap of indicators is between 2004 and 2011 (see Tables 1 and 2). Most European cities are contained in the Urban Audit data set, but we also include the capital cities and cities with a population over 100 000 from the United Nations Demographic Yearbook (UNYB).

Before integration, locations have varying names in different data sets (e.g., Wien vs. Vienna), a Uniform Resource Identifier (URI) for every city is essential for the integration and enables to link the cities and indicators back to DBpedia and other LOD data sets. We choose to have a one-to-one (functional) mapping of every city from our namespace to the English DBpedia resource, which in our republished data is encoded by sameAs relations. We identify the matching DBpedia URIs for multilingual city names and apply basic *entity recognition*, similar to Paulheim et al. [17], with three steps using the city's names from Urban Audit and UNYB:

- Accessing the DBpedia resource directly and following possible redirects;
- Using the Geonames API (http://api.geonames.org/) to identify the resource;
- For the remaining cities, we manually looked up the URL on DBpedia.

DBpedia. DBpedia, initially released in 2007, is an effort to extract structured data from Wikipedia and publish the data as Linked Data [4]. For cities, DBpedia provides various basic indicators such as demographic and geographic information (e.g., population, latitude/longitude, elevation). The Open City Data Pipeline extracts the URLs, weather data, and the population of a city. While we only integrated a limited subset of indicators from DBpedia for now, we plan to add other indicators like economic and spatial indicators in the future. Since temporal validity of indicators is rarely documented, we assume them to be current as accessed.

Urban Audit (UA). The Urban Audit collection started as an initiative to assess the quality of life in European cities. It is conducted by the national statistical institutes and Eurostat. Currently, data collection takes place every three years (last survey in November 2012) and is published via Eurostat (http://ec.europa.eu/eurostat). All data is provided on a voluntary basis which leads to varying data availability and missing values in the collected data sets. Urban Audit aims to provide an extensive look at the cities under investigation, since it is a policy tool to the European Commission: "The projects' ultimate goal is to contribute towards the improvement of the quality of urban life" [15]. At the city level, Urban Audit contains over 250 indicators divided into the categories Demography, Social Aspects, Economic Aspects, and Civic Involvement. Currently, we extract the data sets including the topics population structure, fertility and mortality, living conditions and education, culture and tourism, labour market, transport, and environment.

Table 1: Urban Audit Data Set

Year(s)	Cities	Indicators	Available Values	Missing Values	Missing Ratio (%)
1990	177	121	2 480	18 937	88.4
2000	477	156	10 347	64 065	85.0
2005	651	167	23 494	85 223	78.4
2010	905	202	90 490	92 320	50.5
2004 - 2012	943	215	531 146	1 293 559	70.9
All (1990 - 2012)	943	215	638 934	4 024 201	86.3

Table 2: United Nations Data Set

Year(s)	Cities	Indicators	Available Values	Missing Values	Missing Ratio (%)
1990	7	3	10	11	52.4
2000	1 391	147	7 492	196 985	96.3
2005	1 048	142	3 654	145 162	97.5
2010	2 008	151	10 681	292 527	96.5
2004 - 2012	2 733	154	44 944	3 322 112	98.7
All (1990 - 2012)	4 319	154	69 772	14 563 000	99.5

United Nations Statistics Division (UNSD). The UNSD offers data on a wide range of topics such as education, environment, health, technology, and tourism. Our main source is the UNSD Demographic and Social Statistics, which is based on the data collected annually (since 1948) by questionnaires to national statistical offices (http://unstats.un.org/unsd/demographic/). The UNSD data marts consist of the following topics: population by age distribution, sex, and housing; occupants of housing units/dwellings by broad types (e.g., size, lighting); occupied housing units by different criteria (e.g., walls, waste). The collected data has over 650 indicators, wherein we kept a set of course-grained indicators and drop the most fine-grained indicator level, e.g., keeping *housing units total* but dropping *housing units 1 room*. We prefer more coarse-grained indicators to avoid large groups of similar indicators which are highly correlated. Fine-grained indicators would only be interesting for LOD publication.

U.S. Census. The *County and City Data Book 2007* (USCCDB) of U.S. Census Bureau [26] offers two data sets concerning U.S. statistics; *Table C-1 to C-6* of [26] cover the topics Area and Population, Crime, Civilian Labor Force for cities larger than 20 000 inhabitants; *Table D-1 to D-6* of [26] cover Population, Education, Income and Poverty for locations with 100 000 inhabitants and more. Initially, we have integrated the data sets from Table C-1 to C-3 , which are the only sources including data points for several years, namely 1990, 2000, and 2005. Contrary to the UN and UA data sets, the USCCDB has a low ratio of missing values ranging from 0% to 5% for a total of 1267 cities. The data set contains 21 indicators, e.g., population, crime, and unemployment rate.

Future Data Sources. At the point of writing, the data sources are strongly focused on European cities and demographic data. Hence, we aim to integrate further national and international data sources. The Carbon Disclosure Project (CDP) is an organization based in the UK aiming at "[...] using the power of measurement and information disclosure to improve the management of environmental risk" (https://www.cdp.net/en-US/Pages/About-Us.aspx). The *CDP cities* project has data collected on more than

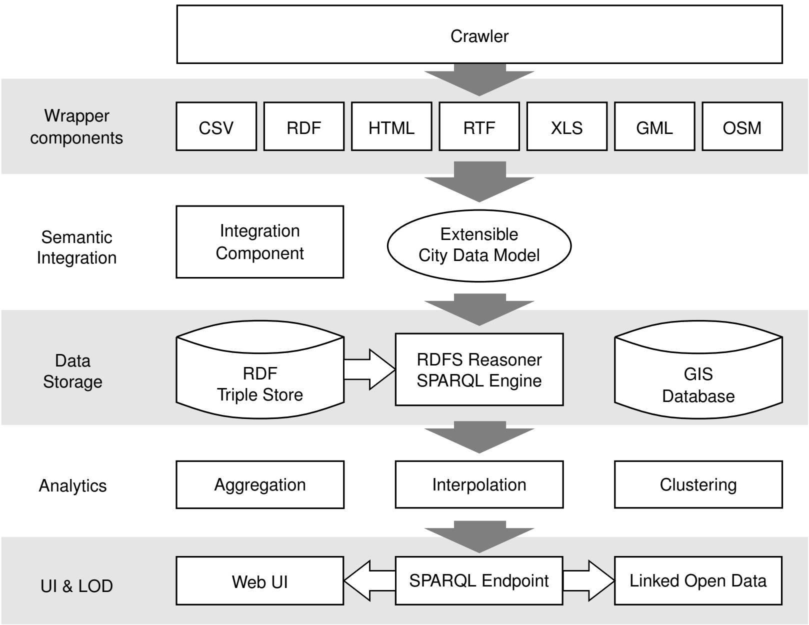


Fig. 1: City Data Pipeline architecture showing components for crawling wrapping, cleaning, integrating, and presenting information

200 cities worldwide. CDP cities offers a reporting platform for city governments using an online questionnaire covering climate-related areas like Emissions, Governance, and Climate risks. Single city open data portals (e.g., New York, Vienna) could be added and integrated. This is surely a large effort by its own, since our crawling and mapping components would have to be extended to deal with heterogeneity of every cities' portal.

3 System Architecture

The Open City Data Pipeline collects data, organizes it into indicators, and shows these indicators to the user. This section introduces the system which is organized in several layers (see Figure 1): *crawler*, *wrapper components*, *semantic integration*, *data storage*, *analytics*, and *external interfaces* (user interface, SPARQL endpoint, and LOD).

Crawler. The Open City Data Pipeline semi-automatically collects data from various registered open data sources periodically dependent on the specific source. The crawler currently collects data from 32 different sources. Due to a high heterogeneity in the source data, adding new data sources is still a manual process, where the source-specific mapping of the data to RDF has to be provided by scripts. However, a more automated mapping process of new sources is an appealing extension for future work.

Wrapper Components. As a first step of data integration, a set of custom wrapper components parses the downloaded data and converts it to source-specific RDF. The set of wrapper components include a CSV wrapper for parsing and cleaning, a wrapper for extracting HTML tables, a wrapper for extracting tables of RTF documents, a wrapper

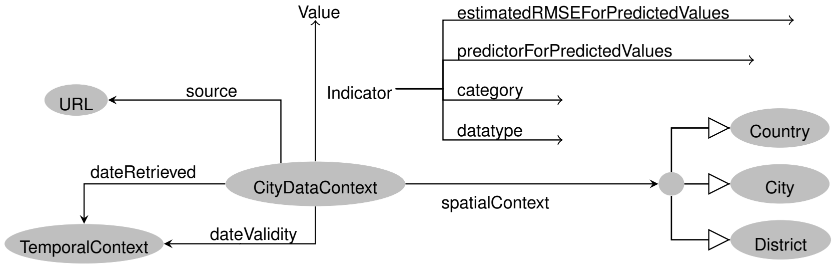


Fig. 2: Excerpt of the City Data Model ontology

for Excel sheets, and a wrapper for cleaning RDF data as well. All of these wrappers are customizable to cater for diverse source-specific issues. These wrappers convert the data to RDF and preprocess the data before integrating the data with the existing triple store. Preprocessing contains data cleansing tasks, i.e., unit conversions, number and data formatting, string encoding, and filtering invalid data (see [20]).

Semantic Integration (Ontology). To access a single indicator such as the population number, which is provided by several data sources, the semantic integration component *unifies the vocabulary* of the different data sources through an ontology (see Figure 2). The semantic integration component is partly implemented in the individual wrappers and partly by an RDFS [5] ontology (extended with capabilities for reasoning over numbers by using equations [2]) called *City Data Model* (see http://citydata.wu.ac.at/ns#). The ontology covers several aspects: spatial context (country, region, city, district), temporal context (validity, date retrieved), provenance (data source), terms of usage (license), and an extensible list of indicators.

Indicator is the super-property of all the indicator properties mapping CityDataContexts to actual values. Each Indicator of the ontology contains, a name, description, a unit of measurement, a data type, and is grouped into one of the following *categories*: (a) Demography, (b) Social Aspects, (c) Economic Aspects, (d) Training and Education, (e) Environment, (f) Travel and Transport, (g) Culture and Recreation, and (h) Geography. To integrate the source-specific indicators the ontology maps data-source-specific RDF properties to City Data Model properties, e.g., it maps dbpedia:population to citydata:population by an RDFS subPropertyOf property. A CityDataContext is an anchor connecting a set of data points to a spatial context, a temporal context, and a data source. When importing an input CSV file containing the indicators as columns and the cities as rows, each row corresponds to (at least) one CityDataContext. The SpatialContext class collects all resources with spatial dimension: country, province, region, city, and district. Furthermore entities of different granularity can be connected by the property locatedIn. The dateValidity property maps a CityDataContext to a point in time where the values are valid. Additionally the property periodValidity can indicate what the validity period is (possible values are biannual, annual, quarterly, monthly, weekly, daily, hourly or irregular). Whereas the dateRetrieved property records the date and time of the data set download. The source property links a CityDataContext to its data source.

Data Storage, Analytics, UI & LOD. To store the processed data we use Jena TDB as a *triple store* for RDF data. Subsequent subsystems can access the RDF data via a SPARQL interface (http://citydata.wu.ac.at/). The SPARQL engine provides RDFS reasoning support by query rewriting (including reasoning over numbers [2]).

The analytics layer includes tools to fill-in missing values by using statistical regression methods. Section 4 describes the missing value prediction in detail. The results are also stored in the RDF triple store and the SPARQL engine provides access to them. Section 7 explains the frontend, user interface, SPARQL endpoint, and publishing data as LOD. Bischof et al. [3] describe the system components in more detail.

4 Prediction of Missing Values

After integrating the different sources, we discovered a large number of missing values in our data sets. We identified two reasons for that:

- As shown in Table 1 and 2, we can observe a large ratio of missing values due to incomplete data published by the data providers;
- More severely, when we combine the different data sets even more missing values are introduced, since there is a fair amount of disjoint cities and indicators.

Base Methods. Our assumption is that every indicator has its own distribution (e.g., normal, Poisson) and relationship to other indicators. Hence, we aim to evaluate different regression methods and choose the best fitting model to predict the missing values. We measure the prediction accuracy by comparing the *normalized root mean squared error* in % (RMSE%) [29] of every regression method.

In the field of Data Mining [29,10] (DM) various regression methods for prediction were developed. We chose the following three "standard" methods for our evaluation due to their robustness and general performance.

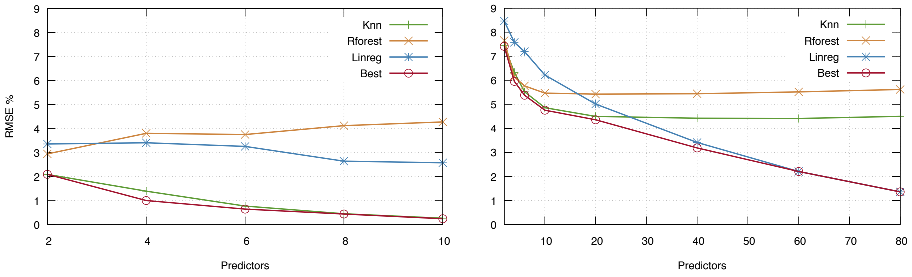
K-Nearest-Neighbour Regression (KNN), models denoted as M_{KNN}, is a wide-spread DM technique based on using a distance function to partition the instance space. As stated in [10], the algorithm is simple, easily understandable and reasonably scalable. KNN can be used in variants for clustering as well as regression.

Multiple Linear Regression (MLR), models denoted as M_{MLR}, has the goal to find a linear relationship between a target and several predictor variables. The linear relationship can be expressed as a regression line through the data points. The most common approach is *ordinary least squares* to measure and minimize the cumulated distances [10].

Random Forest Decision Trees (RFD), models denoted as M_{RFD}, involve the top-down segmentation of the data into multiple smaller regions represented by a tree with decision and leaf nodes. A random forest is generated by a large number of trees, which are built according to a random selection of attributes at each node. We use the algorithm introduced by Breiman [24].

Preprocessing. The preprocessing starts with the extraction of the base data set from our RDF triple store. We use SPARQL queries with the fixed period of *2004–2011* and produce an initial data set as a matrix with tuples of the form $\langle$*City*, *Indicator*, *Year*, *Value*$\rangle$. Based on the initial matrix, we perform the preprocessing as follows:

- Removing boolean and nominal columns, as well as all weather related data and sub-indicators in the U.N. data set, e.g., *occupants of housing units with 2 rooms*;



(a) Approach 1 (Building Complete Subsets)

(b) Approach 2 (PC Regression)

Fig. 3: Prediction results

- Merging the dimensions year and city, resulting in $\langle City\ Year, Indicator, Value\rangle$;
- Transposing the initial matrix by moving the indicators into the columns, resulting in tuples of the form $\langle City\ Year, Indicator_1 Value, \ldots, Indicator_n Value\rangle$;
- Deleting columns and rows which have a missing values ratio larger than 90%.

Our initial data set from UA, UN, and DBpedia contains 3 399 cities with 370 indicators. By merging city and year and transposing the matrix we create 13 482 city/year rows. And after deleting the cities/indicators with a missing values ratio larger than 90%, we have the final matrix of 4 438 rows (city/year) with 207 columns (indicators).

Approach 1 - Building Complete Subsets. In the first approach, we try to build models for a target indicator by directly using the available indicators as predictors. For this, we are using the correlation matrix of the data to find indicators which are suitable predictors. Subsequently, we build a complete subset from our data, i.e., we first perform a projection on our data table, keeping only the predictors and the specific target as columns. More detailed, our approach has the following steps on the initial data set, the matrix A_1 and a fixed number of predictors n (we test this approach on different $n's$):

1. Select the target indicator I_T;
2. Calculate the correlation matrix A_C of A_1 between I_T and the remaining indicators;
3. Create the submatrix A_2 of A_1 with I_T and the n "best" indicators (called the predictors). The predictors are selected according to the highest absolute correlation coefficients in A_C;
4. Create the complete matrix A_3 by deleting all rows in A_2 with missing values;
5. Apply *stratified tenfold cross-validation* (see [29]) on A_3 to obtain ten training- and test sets. Then, train the models M_{KNN}, M_{MLR}, and M_{RFD} using the training sets. Finally, calculate the mean of the ten RMSE% based on the test set for each model and choose the best performing model M_{Best} accordingly;
6. Use the method for M_{Best} to build a new model on A_2 for predicting the missing values of I_T.

The performance of the regression methods were evaluated for two to ten predictors. Two regression methods have their best RMSE% with ten indicators: 0.27% for KNN and 2.57% for MLR. Whereas RFD has the best RMSE% of 4.12% with eight indicators. Figure 3a gives an overview of the results. By picking the best performing regression for every indicator (red line) the median RMSE% can be reduced only slightly. For ten predictors the median RMSE% improves to 0.25% over KNN with 0.27%. Depending on n, we fill-in between 122 056 for ten and 296 069 values for two predictors. For a single city and ten predictors, the number of predicted values range from 7 to 1 770. The limited number of filled-in values is due to the restriction of using the complete matrix for the regression methods.

Approach 2 - Principal Component Regression. In the second approach, we omit the direct use of indicators as predictors. Instead, we first perform a Principal Component Analysis (PCA) to reduce the number of dimensions of the data set and use the new compressed dimensions, called *principal components* (PCs) as predictors. As stated in [10], the PCA is a common technique for finding patterns in data of high dimensions. Parts of the evaluation is similar to Approach 1, but we have an additional step where we *impute* all the missing values with *neutral* values for the PCA. The neutral values are created according to the *regularized iterative PCA algorithm* described in [21]. This step is needed to perform the PCA on the entire data set. The following steps are evaluated having an initial data set A_1 as a matrix and a predefined number of predictors n (we test this approach also on different $n's$):

1. Select the target indicator I_T;
2. Impute the missing values in A_1 using the regularized iterative PCA algorithm resulting in matrix A_2 and remove the column with I_T;
3. Perform the PCA on the A_2 resulting in a matrix A_3 of a maximum of 80 PCs;
4. Append the column of I_T to A_3 creating A_4 and calculate the correlation matrix A_C of A_4 between I_T and the PCs;
5. Create the submatrix A_5 of A_4 on the selection of the PCs with the highest absolute correlation coefficients and limit them by n;
6. Create submatrix A_6 of A_5 for validation by deleting rows with miss. values for I_T;
7. Apply stratified tenfold cross-validation on A_6 with the Step 5 from Approach 1, which results in the best performing model M_{Best};
8. Use the method for M_{Best} to build a new model on A_5 (not A_6) for predicting the missing values of I_T.

Figure 3b shows the median RMSE% for KNN, RFD, MLR, and the best method with an increasing number of predictors. For 80 predictors MLR performs best with a median RMSE% of 1.36%, where KNN (resp. RFD) has a median RMSE% of 4.50% (resp. 5.62%). MLR improves steady up to 80 predictors. KNN provides good results for a lower number of predictors, but starts flattening with 20 predictors. An increasing number of k could improve the result though. The red line in Figure 3b shows the median RMSE% with the best regression method chosen. Up to 60 predictors, the overall results improves by selecting the best performing method (for each indicator). The best median RMSE% of 1.36% is reached with 80 predictors. For this, MLR is predominant but still 14 out of 207 indicators are predicted by KNN.

As mentioned, we have two quality measurements to evaluate our approaches. First, it is important to build models which are able to predict many (preferably all) missing values. Second, the prediction accuracy of the models is essential, so that the Open City Data Pipeline can fulfill its purpose of publishing high-quality, accurate data and predictions. Prediction accuracy is higher in Approach 1 than 2 (for 4 to 10 predictors), which we relate to the reduced size of the data set. However in Approach 1, we fill-in at the maximum 296 069 values with 2 predictors (median RMSE% of 2.09%), which is about 66% of Approach 2. Due to the reduced number of predictions, we will apply Approach 2 for publishing the filled-in missing values.

5 Cross Data Set Prediction

Our initial overall matrix has 13 482 city/year rows and 369 columns, which are reduced after deleting all with a missing values ratio of 90% to the matrix of 4 438 rows and 207 columns. Cross Data Set Predictions (CDP) aims to fill the gap of the 162 columns mainly caused by the disjointness of indicators/cities in the data sets (e.g., UN and UA). As seen in Figure 4, there are two areas which are not covered, the first is the UA cities for the UN indicators and the second is the UN cities for the UA indicators. The success of the CDP approach depends on one data set, which has a reasonable amount of overlapping cities with the other data sets. At the time of writing the UN data set seems the most promising covering cities of the whole world.

For CDP, we always select one data set (e.g., UN), called the source data set S, and predict into another data set (e.g., UA), called the target data set T, denoted as $S \to T$. We evaluate again the different base regression methods and choose the best fitting model for prediction. The preprocessing is altered so we only delete columns and rows which are entirely empty. Since Approach 1 needs a complete matrix, we only consider Approach 2 and modify it accordingly. We start in the CDP approach with the initial source and target data sets A_S and A_T. The following steps are evaluated for a different number of predictors n:

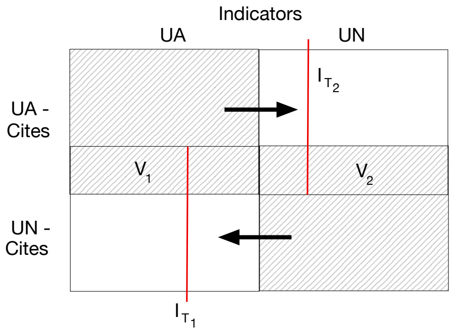
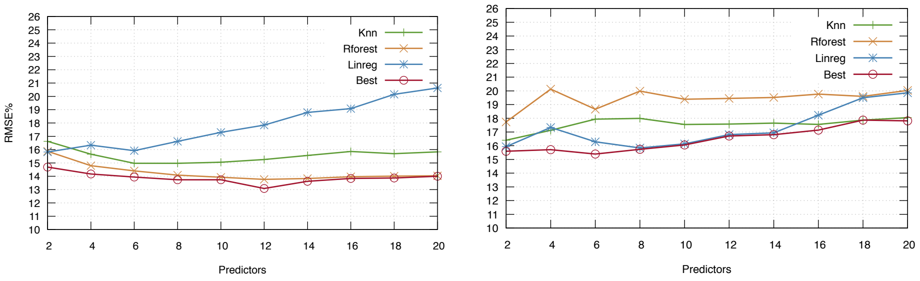


Fig. 4: Predicting I_{T_1} (resp. I_{T_2}) from the UN (resp. UA) data set

1. Select the target indicator I_T from T;
2. Impute the missing values in A_S using the regularized iterative PCA algorithm resulting in matrix A_{S_2};
3. Perform the PCA on the A_{S_2} resulting in a matrix A_{S_3} of a maximum of 40 PCs;
4. Append the column I_T to A_{S_3} creating A_{S_4} and calculate the correlation matrix A_{S_C} between I_T and the PCs;
5. Create the submatrix A_{S_5} of A_{S_4} on the selection of the PCs with the highest absolute correlation coefficients and limit them by n;
6. Create validation submatrix A_{S_6} of A_{S_5} by deleting rows with missing values for I_T;
7. Apply stratified *fivefold* cross-validation on A_{S_6} similar to Step 7 from Approach 2, which results in the best performing model M_{Best};[6]
8. Use the method for M_{Best} to build a model on A_{S_5} to predict missing values of I_T.

[6] Cross-validation is reduced from ten- to fivefold, so the test set for validation is large enough.



(a) UN → UA

(b) UA → UN

Fig. 5: Cross data set prediction results

Note that the validation of Step 7 is performed on the set V_1 or V_2 of cities overlapping S and T. We ignore a target indicator if the set is empty, since we can not determine the quality of our prediction. The amount of overlapping cities with values ranging for T as UA (resp. T as UN) from 16 (resp. 11) to 1194 (resp. 1429) with an average of 445 (resp. 88) cities. We performed the CDP from UN → UA and the results are shown in Figure 5a. RFD performs best for with a median RMSE% of 13.76% for 12 predictors. The median RMSE% of M_{Best} is 13.08% with 12 predictors and always very close to the RFD results. With more than 12 predictors, the result does not improve anymore. The population related indicators are predicted best (e.g., *Population male* has a RMSE% of 4.86%), weather related indicators are worst (e.g., *Total hours of sunshine per day* has a RMSE% of 1176.36%). The reason lies within in the UN source data set, where three indicators *population*, *population female*, and *population male* are predominant. For the UA → UN predictions, shown in Figure 5b, the results are best with a median RMSE% of 15.40% with 6 predictors. More predictors do not improve the result, whereas MLR performs best overall. The most frequent indicators, again *population*, are reasonable predicted, whereas most of the other UN indicators can not be proper validated due to the low number of overlapping cities (avg. of 88). If we apply the threshold RMSE% of 7% for publishing the predicted values, we are able to predict for UN → UA: 34 out of 210 indicators; and for UA → UN: 2 out of 142 indicators. The results for UN → UA are satisfying, since we are able to predict 34 UA indicators for UN cities, the picture is dimmer for UA → UN, where only a few indicators are below our threshold.

6 Learning Ontology Mappings from Indicator Values

So far we used regression methods to predict missing values over the whole integrated data set as well as across different source data sets. But regression can also be a means to learn ontology axioms to express *dependencies between pairs of indicators* from different data sources. By exploiting these dependencies a reasoner can give more complete answers without materializing a potentially large number of new values beforehand.

The models expressing these dependencies should be intuitive, i.e., comprehensible by a domain expert, and should allow derivation of new values. We focus on pairs of indicators to cover several cases: (1) the same indicator, (2) the same indicator with different units (for example area in km^2 in mile^2), (3) somewhat normalized indicators. Since we are interested in simple models and numerical data, we model the dependencies by *linear equations* containing two indicators from two different sources. Furthermore some data sources (UA) already publish the equations used to compute some of their indicators such as population density. Because of high dependencies of indicators within a data set we only consider pairs of indicators from different data sets.

As a special case we consider pairs of *equivalent indicators*, e.g., many data sets have an indicator for population. We could model this dependency as simple equation $p_1 = p_2$ but ontology languages already provide axioms to express the equivalence of two properties which in turn any standard Semantic Web reasoner can use to get more complete data. OWL 2 provides the EquivalentDataProperties axiom, while RDFS allows modeling equivalent properties by a pair of symmetric subPropertyOf axioms.

We use linear regression to compute the dependencies. In general linear regression estimates the *intercept* a and the *slope* b for a linear equation $y = a + bx$ where x is the *independent variable* (predictor indicator) and y the *dependent variable* (response indicator). Thus it tries to fit a line to the data with as little error as possible. A popular error measure is *least-squares* which is known to suffer heavily from outliers [29] which is also the case for our data set. Thus we perform a *robust regression*, which is computationally more expensive but handles both horizontal and vertical outliers better. We use the R function rlm of the MASS library which implements robust regression by iterated re-weighted least squares and Huber weights [27]. Applying robust regression to all pairs of indicators from UN and Urban Audit results theoretically in 214×148 linear models. Many pairs of indicators have no *complete observations*, i.e., cities with values for both indicators, for which regression can not be applied.

For the ontology we want to keep only those linear models for which the pair of indicators has a strong dependency. We first filter out all dependencies which have less than 100 complete observations. Next we compute a correlation matrix of all indicator pairs to quantify the indicator dependency. Since the standard Pearson correlation assumes a normal distribution, which the indicators not necessarily follow, we use the non-parametric *Kendall rank correlation coefficient* τ implemented in the R function cor [19], instead. We filter out all models with a correlation less than 0.7.

For finding equivalent properties we perform a second linear regression without an intercept, i.e., forcing the linear model through the origin. As before we filter out linear models with low correlation or insufficient complete observations. If the slope of this second linear model is 1 ± 0.01, then we consider the indicator pair as equivalent.

When performing this approach on the UN and UA data sets we get 98 linear equations, 4 of which indicate equivalent indicator pairs published in our ontology. Neither OWL nor RDFS provide a means to express linear equations except property equivalences (represented as sketched above). Thus, for the remaining linearly dependent indicator pairs we use the notation as in previous work [2] to express the respective mappings in our ontology. Additionally the ontology contains the number of complete observations and the correlation for each new axiom as annotations. Detecting more complex relationships between a *set* of indicators from one datasource and a single indicator from a second dataset (which would also be expressible as equations using the notation of [2]) is on our agenda.

7 Publishing as Linked Data

Linked Open Data. The resources (cities) and properties in the City Data namespace (http://citydata.wu.ac.at/) are published according to the Linked Data principles. The ontology (as described in Section 3), contains all City Data property and class descriptions. Each city is assigned a dereferencable URI, e.g., http://citydata.wu.ac.at/resource/Ljubljana for the capital of Slovenia. Depending on the HTTP `Accept` header the server will return either an HTML, RDF/XML, or Turtle representation after a HTTP 303 redirect. The city resources are linked to the LOD cloud via owl:sameAs to the corresponding DBpedia resources.

Predictions. The prediction workflow is based on the current data in the triple store. The *preprocessing* is written in Python and *prediction* and *evaluation* is developed in R [19] using its "standard" packages. As mentioned before, we only publish the predicted values from Approach 2. After the best regression method is selected for a particular indicator, we use this method to fill-in all the missing values and publish them as a *new* indicator with a *prefix* in the CityDataContext. We also add the the source and the year for the prediction. The threshold for publishing is a RMSE% of 7% with 80 predictors. This leads to 6 indicators (e.g. *price of a* m^3 *of domestic water in EUR*) being dropped. We then introduce two new properties describing for each indicator the quality of the prediction by the median RMSE% and the regression method used. In future work, we aim to publish the data using the PROV Data Model [8].

Interface. A simple Java powered web interface allows users to select exactly which subset of the data should be shown. The interface provides programmatic access via HTTP GET to allow external tools such as data visualization frameworks, to query the database. The web application communicates with the Jena triple store via SPARQL 1.1. Users can select one or more of the 450 *indicators* sorted by categories like *Demography*, *Geography*, *Social Aspects*, or *Environment*. The list also shows how many data points are available per indicator and for how many cities data points are available for this indicator. Next the user can select one or several of more than 5 260 *cities* for which we collected data. For a few cities we even have information on the individual districts available. In these cases the user can select one or several of the districts. Optionally the user can specify a *temporal context*, for which year the database should be queried. This feature allows to compare several cities with each other at a certain point of time instead of listing data of all available times.

8 Lessons Learnt and Related Work

We emphasize that our work is not a "Semantics in-use" paper in the classical sense of applying Semantic Web technologies to solve a use case, but rather a demonstration that a portfolio of statistical methods *in combination* with semantic technologies for data integration helps to collect, enrich and serve domain-specific data in a reusable way for *further applications* of the LOD cloud to be developed on top. While there are practical use cases within Siemens, such as studies like the Green City Index [6] which can benefit from an up-to-date data repository for city data, we are looking forward to diverse other applications on top of our collection by others. Also, we have demonstrated that building a domain-specific Open Data pipeline is feasible and enabled by Semantic Web technologies. We envision that such an approach may be worthwhile for other domains as well as a multiplicator to leverage usage of Open Data: for instance similar data pipelines could be built for business intelligence, investment use cases for company data, or finance data. For publishing the prediction as LOD, we set a threshold RMSE% of 7%, which could be adjusted according to the domain of use.

Lessons Learnt. In the wrapper component, integrating cities and indicators for a new data set (often CSV tables) is still a slow manual process and needs custom scripting. The entity recognition for cities and the ontology learning techniques from Section 6 provide a first automation step, where indicators of new data sets can be mapped to existing indicators. This approach is similar to instance based mapping learning techniques also used in ontology matching (cf. [7]). In the analytics and query component, we have had to deal with sparse data sets with many missing values, which is a drawback for analyzing and reusing the data. By applying the PCA-based Approach 2, using a basket of *standard* DM techniques without customization, we reach a good quality for predictions (overall RMSE% of 1.36%) and are able to fill large gaps of the missing values. However, Approach 2 does not tackle the gap of disjoint cities/indicators, which is addressed by extending it to the CDP approach, where we predict from one single data set into another. We applied CDP for predicting UA → UN and UN → UA and discovered reasonable results for the first but unsatisfying for the second direction. The cause for the unsatisfying results can be found in the UN data set with sufficient values for only three population-related indicators. For the CDP approach to succeed, we need one *base* data set which covers a wider range of cities/indicators; this is not the case yet.

Related Work. *QuerioCity* [13] is a platform to integrate static and continuous data with Semantic Web tools. While it uses partly similar technologies, it works as a single city platform and not as a data collection of many cities and concentrates on data integration. We focus on predicting missing values, and publishing the outcomes as Linked Data. The EU project *CitySDK* (http://www.citysdk.eu/) provides unifying APIs, including a Linked Data API for mobility and geo data usable across cities. These reusable APIs enable developers to create portable applications and ease service provisioning for city administrators. If enough cities adopt CitySDK, its APIs can become a valuable data source for the Open City Data Pipeline as well. Regarding the methods, works of Paulheim et al. [16,17,18] are closely related, however they focus on unsupervised DM approaches of unspecified features from Linked Data instead of filling-in missing values for specific attributes. The work by Nickel et al. [14] focuses on relational learning, i.e., rather learning object relations than predicting numeric attribute values. The work in [11] also integrates statistical Linked Data, however it is mainly concerned with query rewriting and less with missing values. The Open City Data Pipeline uses techniques from ETL frameworks (cf. [25]) and DM tools (e.g., [9]) which are *general* technologies and build our base techniques. The main difference to a plain ETL and DM approach concerns (a) the ontology-based integration with query capabilities and continuous integration in the LOD cloud, (b) the ontology-learning capabilities, and (c) using the axioms of the ontology to validate the prediction results by the data type and ranges.

9 Conclusions and Future Work

In this paper we have presented the *Open City Data Pipeline*, an extensible platform for collecting, integrating, and predicting open city data from several data providers including DBpedia and Urban Audit. We have developed several components including a data crawler, wrappers, an ontology-based integration platform, and a missing value prediction module. Having sparse data sets, the prediction of missing values is a crucial component. For this, we have developed two approaches, one based on predicting a target indicator directly from other indicators, and one based on predictors from components calculated by Principal Components Analysis (PCA). We applied for both approaches three basic regression methods and selected the best performing one. They were compared regarding the number of filled-in values and prediction accuracy, concluding that the PCA-based approach will be used for future work. Filled-in missing values are then published as LOD for further use. In case of a large disjointness regarding indicators/cities, we extended the second approach to Cross Data Set Predictions (CDP).

Our future work includes extensions of the presented data sets, methods, and the system itself. Regarding the data sets, we already mention several sources, e.g., the Carbon Disclosure Project, which are needed to cover a wider range of cities worldwide. As to the methods, CDP has to be evaluated with more data sets to further evaluate the performance of CDP and find the threshold of indicators/cities with sufficient *overlapping* values. We also aim to extend our basket of base methods with other well established regression methods. Promising candidates are Support Vector Machines [22], Neural Networks, and Bayesian Generalized Linear Model [28]. Moreover, we plan to publish more details on the best regression method per indicator as part of our ontology: so far, we only indicate the method and estimated RMSE%, whereas further details such as used parameters and regression models would be needed to reproduce and optimize our predictions. Ontologies such as [12] could serve as a starting point here. We also plan to connect our plattform to the Linked Geo Data Knowledge Base [23] including OpenStreetMap (OSM) data: based on such data, new indicators could be directly calculated, e.g., the size of public green space by aggregating all the parks. Furthermore, we are in the process of improving the user interface to make the application easier to use. For this we investigate several libraries for more advanced information visualization.

Acknowledgements. This work was supported by the Vienna Science and Technology Fund (WWTF) project ICT12-15 and the EU project CityPulse FP7-609035.

References

1. Bettencourt, L.M.A., Lobo, J., Helbing, D., Kühnert, C., West, G.B.: Growth, innovation, scaling, and the pace of life in cities. Proc. of the National Academy of Sciences of the United States of America 104(17), 7301–7306 (2007)
2. Bischof, S., Polleres, A.: RDFS with Attribute Equations via SPARQL Rewriting. In: Proc. of ESWC 2013, pp. 335–350. Springer (2013)
3. Bischof, S., Polleres, A., Sperl, S.: City Data Pipeline. In: Proc. of the I-SEMANTICS 2013 Posters & Demonstrations Track. pp. 45–49 (2013)
4. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the web of data. J. Web Sem. 7(3), 154–165 (2009)

5. Brickley, D., Guha, R., (eds.): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, W3C (2004)
6. Economist Intelligence Unit (ed.): The Green City Index. Siemens AG (2012)
7. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, 2nd edn. (2013)
8. Gil, Y., Miles, S.: PROV Model Primer. W3C Note, W3C (2013)
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
10. Han, J.: Data Mining: Concepts and Techniques, 3rd Edition. Morgan Kaufmann Publishers Inc. (2012)
11. Kämpgen, B., O'Riain, S., Harth, A.: Interacting with statistical linked data via OLAP operations. In: Proc. of ESWC 2012 Satellite Events. pp. 87–101. Springer (2012)
12. Keet, C.M., Ławrynowicz, A., d'Amato, C., Kalousis, A., Nguyen, P., Palma, R., Stevens, R., Hilario, M.: The data mining OPtimization ontology. Web Semantics: Science, Services and Agents on the World Wide Web 32(0), 43 – 53 (2015)
13. Lopez, V., Kotoulas, S., Sbodio, M., Stephenson, M., Gkoulalas-Divanis, A., Aonghusa, P.M.: QuerioCity: A linked data platform for urban information management. In: Proc. of ISWC 2012, pp. 148–163. Springer (2012)
14. Nickel, M., Tresp, V., Kriegel, H.: Factorizing YAGO: scalable machine learning for linked data. In: Proc. of WWW 2012. pp. 271–280 (2012)
15. Office for Official Publications of the European Communities: Urban Audit. Methodological Handbook (2004)
16. Paulheim, H.: Generating possible interpretations for statistics from linked open data. In: Proc. of ESWC 2012. pp. 560–574. Springer (2012)
17. Paulheim, H., Fürnkranz, J.: Unsupervised generation of data mining features from linked open data. In: Proc. of WIMS 2012. p. 31. ACM (2012)
18. Paulheim, H., Ristoski, P., Mitichkin, E., Bizer, C.: Data mining with background knowledge from the web. In: Proc. of the 5th RapidMiner World (2014)
19. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing (2009)
20. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Eng. Bull. 23(4), 3–13 (2000)
21. Roweis, S.T.: EM algorithms for PCA and SPCA. In: Advances in Neural Information Processing Systems 10 (NIPS 1997). pp. 626–632 (1997)
22. Sanchez, V.: Advanced support vector machines and kernel methods. Neurocomputing 55(1–2), 5 – 20 (2003)
23. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: A core for a web of spatial open data. Semantic Web 3(4), 333–354 (2012)
24. Statistics, L.B., Breiman, L.: Random forests. In: Machine Learning. pp. 5–32 (2001)
25. Thomsen, C., Pedersen, T.: A survey of open source tools for business intelligence. In: Data Warehousing and Knowledge Discovery, pp. 74–84. Springer (2005)
26. U.S. Census Bureau: County and City Data Book 2007. https://www.census.gov/compendia/databooks/ (2007)
27. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. Springer, 4th edn. (2002)
28. West, M., Harrison, P.J., Migon, H.S.: Dynamic generalized linear models and bayesian forecasting. Journal of the American Statistical Association 80(389), 73–83 (1985)
29. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition. Morgan Kaufmann Publishers Inc. (2011)