Predicting Incorrect Mappings: A Data-Driven Approach Applied to DBpedia

Mariano Rico
Ontology Engineering Group,
Universidad Politécnica de Madrid
Madrid, Spain
mariano.rico@fi.upm.es

Nandana Mihindukulasooriya
Ontology Engineering Group,
Universidad Politécnica de Madrid
Madrid, Spain
nmihindu@fi.upm.es

Dimitris Kontokostas
AKSW, Department of Computer Science, Leipzig University
Leipzig, Germany
kontokostas@informatik.uni-leipzig.de

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

Sebastian Hellmann
AKSW, Department of Computer Science, Leipzig University
Leipzig, Germany
hellmann@informatik.uni-leipzig.de

Asunción Gómez-Pérez
Ontology Engineering Group,
Universidad Politécnica de Madrid
Madrid, Spain
asun@fi.upm.es

ABSTRACT

DBpedia releases consist of more than 70 multilingual datasets that cover data extracted from different language-specific Wikipedia instances. The data extracted from those Wikipedia instances are transformed into RDF using mappings created by the DBpedia community. Nevertheless, not all the mappings are correct and consistent across all the distinct language-specific DBpedia datasets. As these incorrect mappings are spread in a large number of mappings, it is not feasible to inspect all such mappings manually to ensure their correctness. Thus, the goal of this work is to propose a data-driven method to detect incorrect mappings automatically by analyzing the information from both instance data as well as ontological axioms. We propose a machine learning based approach to building a predictive model which can detect incorrect mappings. We have evaluated different supervised classification algorithms for this task and our best model achieves 93% accuracy. These results help us to detect incorrect mappings and achieve a high-quality DBpedia.

CCS CONCEPTS

• **Information systems → Resource Description Framework (RDF)**; • **Computing methodologies → Cross-validation**; *Semantic networks*; • **Social and professional topics →** *Quality assurance*;

KEYWORDS

Linked Data, Data Quality, Mappings, DBpedia, Machine Learning

SAC 2018, April 9–13, 2018, Pau, France

ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167164

ACM Reference format:
Mariano Rico, Nandana Mihindukulasooriya, Dimitris Kontokostas, Heiko Paulheim, Sebastian Hellmann, and Asunción Gómez-Pérez. 2018. Predicting Incorrect Mappings: A Data-Driven Approach Applied to DBpedia. In *Proceedings of SAC 2018: Symposium on Applied Computing , Pau, France, April 9–13, 2018 (SAC 2018),* 8 pages.
https://doi.org/10.1145/3167132.3167164

1 INTRODUCTION

A large number of RDF knowledge bases are created by transforming non-RDF data sources into RDF. Such non-RDF formats include relational databases, CSV files, key-value pairs, etc. A key input to this transformation process is a mapping that defines how to transform the non-RDF source data into RDF. Such mapping specifies how to map the source schema into RDF vocabularies, and possibly other data transformations.

DBpedia [2], the main hub of the Linked Open Data Cloud, exposes data from Wikipedia as machine-readable Linked Data. Until 2011, only data from the English language Wikipedia was extracted but, since then, several "language-specific" DBpedia chapters were created for other Wikipedia languages. For example, the Spanish DBpedia[1] (esDBpedia) was created in 2012 and currently there are more than 15 other "language-specific" DBpedia chapters.

The DBpedia data extraction process generates RDF data based on the mappings [15] that map: (1) Wikipedia infobox templates to classes of the DBpedia ontology, and (2) infobox template key-value pairs of each Wikipedia infobox template to ontology properties. In a large knowledge base such as DBpedia there is a considerably large number of mappings; for instance, DBpedia 2016-04 version has more than 5K template mappings (for all languages) and much larger number of infobox template key-value pairs. As these mappings are created by a diverse community of volunteers using crowd-sourcing techniques, it is frequent to have wrong or inconsistent mappings. Notice that each incorrect mapping produces a plethora of incorrect data because there are thousands of infobox instances for a given infobox template. For instance, if the infobox

[1] http://es.dbpedia.org

template for Mountain, which has 16 thousand instances (mountain entries) in the English Wikipedia, has an incorrect mapping for one of its keys (e.g. *dbo:height* instead of *dbo:elevation* for the key *elevation*), all the mountain instances would get an incorrect value for that property.

The wrong mappings can be due to many reasons. The DBpedia ontology is considerably large and evolves in a collaborative manner with significant additions and removals in each version and some mapping contributors do not have knowledge about all the 685 classes and 3500+ properties of the ontology. Further, there are also language issues. Most of the contributors of the language-specific DBpedia chapters are non-native English speakers and they have influences from their native language when selecting mapping terms. For instance, while in the English DBpedia *elevation* of an mountain is mapped to the `dbo:elevation` property most language-specific DBpedia instances map it to *dbo:height* which is not intended for mountains but for humans. Inconsistencies of the mappings (same semantic relation mapped to distinct properties) can also be caused by duplicate properties in large ontologies such as the DBpedia ontology. For instance, the DBpedia ontology has two properties *dbo:formationYear* and *dbo:foundingYear* and both denote the year an organization was established. As both terms are suitable for representing the relation, different DBpedia language-specific chapters use them in mappings in an inconsistent manner. In such cases, even though the generated data is not semantically incorrect, the proper reuse of data is hindered because it is hard to query the data (the users will have to use all the possible alternatives in queries) and hard to integrate data from multiple language-specific datasets.

Another problematic type of mappings is mapping more specific relations to generic properties in DBpedia. For instance, for describing the postal code of a city, some mappings use the generic property *dbo:code* instead of the more specific and most appropriate property *dbo:postalCode*. Such mappings also create inconsistencies among data from different language-specific DBpedia datasets and make queries across the DBpedia language-specific datasets harder. Thus, all these inaccuracies and inconsistencies in mappings either lead to incorrect data or causes inconveniences for querying or data integration. However, as these inaccuracies and inconsistencies are scattered over a large number of mappings, it is not feasible to analyze each of them manually by a set of experts. Further, most of such inconsistencies are not unveiled when they are inspected individually but rather uncovered only when they are compared to other similar mappings.

The main objective of our work is to propose a data-driven method to detect the aforementioned mapping deficiencies automatically by analyzing: (1) instance data from distinct language-specific datasets, and (2) the ontological axioms of the DBpedia ontology. More concretely, our goal is to build a classifier using a set of features that can be used to identify such deficiencies. In this way we would be able to automatically classify mappings as Correct or Incorrect.

In our approach, we use the intuitive assumption that when a given resource has the same object value for two distinct properties, there is a high probability of having a mapping inconsistency, *i.e.*, the same relation is mapped to two distinct properties. For example, if the English and Spanish DBpedia have the same subject-object pair <Mount_Everest, 8848> related with two distinct properties `dbo:elevation` and `dbo:height`, it may be possible that both properties refer to the same relation. To identify such occurences, we use different language-specific DBpedia instances (for example, English DBpedia, Spanish DBpedia, Greek DBpedia, Dutch DBpedia) with similar data with equivalence relations (as owl:sameAs links) among them.

Nevertheless, we also take into account that the possibility of two distinct properties having the same value by coincidence (e.g., *birthPlace* and *deathPlace* of a person or *largestCity* and *capital* of a country) could have same subject and same object pairs quite frequently even though those relations are not semantically equivalent. We define a set of data-driven metrics taking all these aspects into account. Further, we also define a set of metrics based on the ontological axioms such as the domain and range of a given property and their hierarchical relationships to extract features that can help to determine if a given mapping is incorrect.

Our hypothesis is that we can develop a classifier capable of identifying incorrect mappings with high precision by using the features that we have defined in this study. In order to evaluate this hypothesis we have created a set of test data by manually annotating DBpedia mappings pairs from multiple languages (English-Spanish, English-Greek, English-Dutch, Spanish-German). The selection of the languages were driven by the availability of human annotators. We have analyzed a selection of supervised learning classification algorithms. In the best case, the proposed classifier (based on the Random Forest algorithm) has an overall accuracy of 93% (mappings classified correctly as 'Correct' or 'Incorrect').

The main contributions of this work are (a) a feasibility study of using data-driven features for classification of incorrect mappings, (b) a set of test data manually annotated by the DBpedia experts that can be reused for other studies, and (c) a predictive model for detecting incorrect mappings in English, Spanish, Greek, and Dutch DBpedia instances.

2 PROBLEM DEFINITION

The main research problem studied in this paper is *"is it possible to automatically detect incorrect mappings by analyzing two knowledge graphs created using two sets of different mappings?"*. We assume that the knowledge graphs have an overlap, *i.e.*, some entities are described in both, and the coreferences (*owl:sameAs* links) can be resolved.

In the DBpedia use case, we can find a large number of datasets that contain similar data, *i.e.*, the language-specific DBpedia knowledge bases and a large portion of the entities present in those knowledge bases are linked by equivalence relationships based on the manually annotated wikilinks contributed by the Wikipedia community. For instance, the entity *dbr:Mount_Everest* is included in more than 15 other DBpedia language-specific datasets, and the coreferences (i.e., the links that refer to the same entity) annotated using the *owl:sameAs* relation [8]. All these entities that describe Mount Everest in different DBpedia datases such as English DBpedia, Spanish DBpedia, Greek DBpedia, and German DBpedia mostly contain similar information about it such as its location,

Table 1: Data from correct and consistent mappings

DBpedia	Subject	Predicate	Object
English	dbr:Mount_Everest	geo:long	86.925278
		dbo:mountainRange	dbr:Himalayas
Spanish	dbr-es:Monte_Everest (owl:sameAs dbr:Mount_Everest)	geo:long	86.925278
		dbo:mountainRange	dbr-es:Himalayas (owl:sameAs dbr:Himalayas)
Greek	dbr-el: (owl:sameAs dbr:Mount_Everest)	geo:long	86.925278
		dbo:mountainRange	dbr-el: (owl:sameAs dbr:Himalayas)
German	dbr-de:Mount_Everest (owl:sameAs dbr:Mount_Everest)	geo:long	86.925278
		dbo:mountainRange	dbr-de: Mahalangur_Himal (owl:sameAs dbr:Himalayas)

Table 2: Data from an incorrect mapping

DBpedia Dataset	Subject	Predicate	Object
English	dbr:Mount_Everest	dbo:elevation	8848
Spanish	dbr-es:Monte_Everest	**dbo:height**	8848
Greek	dbr-el:	dbo:elevation	8848
German	dbr-de:Mount_Everest	dbo:elevation	8848

elevation, important climbers, etc. Thus, as it can be seen in Table 1 such relations are mapped correctly to the same corresponding property in all DBpedia language-specific datasets.

However, in some cases, due to the errors in the mappings such relations can be mapped to properties not intended for the given relation. For instance, Table 2 shows the property that corresponds to the elevation of the Month Everest in each dataset. As it can be seen from the table, while three of the datasets have mapped that relation to `dbo:elevation` property, the Spanish DBpedia dataset has mapped it to `dbo:height`. In our work, we use a set of features extracted from the two datasets to identify such incorrect mappings.

In DBpedia, this data is generated from Wikipedia by mapping Wikipedia infobox keys to DBpedia ontology properties. DBpedia maintains mappings for each language-specific Wikipedia template and those mappings are improved over time by contributers adding mappings to common keys in those infobox templates. At the moment, these mappings are maintained in a specific wiki[2] and the DBpedia contributors with access to the mapping wiki maintain and improve the mappings on regular basis. However, as those mappings are created by a diverse community, inaccuracies and inconsistencies are introduced during this process. In this work, we categorize mappings firstly into two categories: correct and incorrect mappings.

Our hypothesis is that we can develop a classifier capable of identifying incorrect mappings with high precision by using the features that we have defined in this study. In the next section, we discuss an approach for developing such classifier and how to evaluate it.

[2]See http://mappings.dbpedia.org

3 APPROACH

This section describes the feature engineering for creating the predictive model, as well as the model preparation process.

3.1 Feature descriptions

The features that we used for building the model are of two types: (a) instance-based features, and (b) schema-based features. We will define the features using two RDF graphs, graph G_i where $\langle S_i, P_i, O_i \rangle \in G_i$ and graph G_i where $\langle S_j, P_j, O_j \rangle \in G_j$.

Instance-based features are extracted from *ABox* information in a data-driven manner. The key advantage of such features is that they can be extracted even when no schema information is available. There are 4 direct features, denoted by M1-M4 and 4 derived features, denoted C1-C4. Direct features can be any whole number, derived features always have a normalized range of $[0-1]$.

$M_1(G_i, G_j, P_i, P_j)$: M_1 is defined as the count of S_i resources i.e. $\langle S_i, P_i, O_i \rangle \in G_i$, $\langle S_j, P_j, O_j \rangle \in G_j$, and S_i is same as or equivalent to S_j. This gives the frequency of resources having property P_i in graph G_i while having property P_j in G_j. The rationale behind this feature is to check how probable is that a resource will have property P_j in G_j given that the resource has P_i in G_i. For example, if the elevation of a mountain is mapped to dbo:elevation in English DBpedia, and wrongly mapped to dbo:height in Spanish DBpedia, the instances of mountains having dbo:elevation in English DBpedia and dbo:height in Spanish. Our intuition is if P_j is wrongly mapped in G_j (to denote P_i) this number should be high.

$M_2(G_i, G_j, P_i, P_j)$: M_2 is defined as the number of $\langle S_i, O_i \rangle$ pairs i.e. $\langle S_i, P_i, O_i \rangle \in G_i$, $\langle S_j, P_j, O_j \rangle \in Gj$, S_i is equivalent to S_j, and also O_i is equivalent to O_j. Because M_2 only counts occurrences where the objec is also the same, it is even stronger indication that two properties refer to the same relation (see, Table 2). Our intuition is that when M_2 is high, there is a higher-probability that those properties may refer to the same relation. For example, if we take M_2(*EN,ES,dbo:elevation,dbo:height*), there are 5108 resources with the same object value in two graphs, giving an indication that they might refer to the same relation and one mapping is inconsistent.

$M_3(G_i, G_j, P_i, P_j)$: M_3 is defined as the number of S_i resources i.e. $\langle S_i, P_i, O_i \rangle \in G_i$ and $\langle S_j, P_j, O_j \rangle \in G_j$ where S_i is equivalent to S_j but O_i is different from O_j. This features looks for the opposite of M_2. The rationale is that it is possible to have false positives in M_2, for example, people are born and have died in the same place by coincidence (*dbo:birthPlace*/*dbo:deathPlace*) or actors who are also directors in given film (*dbo:directedBy*/*dbo:actor*). Thus, in this feature we count the number of counter examples. Our intuition is that when we find few matches in M_2 by coincidence, M_3 should be able to find reasonable amount of counter examples.

$M_4(G_i, P_i, P_j)$: M_4 is defined as the number of S_i resources in graph G_i that contain both property P_i and P_j simultaneously in the same graph. The rationale for this feature is similar to M_3. The intuition is that this is higher when the two properties denote two distinct relation than when they denote the same. For example,

more resources have both *dbo:birthPlace* and *dbo:deathPlace* properties in same resource simultaneously (denoting two distinct relations) compared to *dbo:elevation* and *dbo:height* in same resource.

The derived features are calculated by normalizing M_2, M_3, and M_4 using M_1 *i.e.* $C_1 = M_2/M_1$, $C_2 = M_3/M_1$, and $C_3 = M_4/M_1$.

Schema-based features are extracted from *TBox* information. When we manually analyzed mappings, we found that it's common to have generic properties in some wrong mappings, for instance, using *dbo:code* instead of *dbo:postalCode*. Further, we noticed wrong mappings could also be due to duplicated properties (for the same relation), for example, *dbo:foundingYear* and *dbo:formationYear*. Schema-based features try to capture hints for such cases. There are 11 schema-based features: TB1-TB11.

TB_1 checks if the property P_i is a subproperty of P_j and TB_2 checks vice versa, *i.e.*, P_j is a subproperty of P_i. TB_3 checks if the classes corresponding are the same in both graphs. TB_4 checks if the class in G_i is a subclass of the class in G_j and TB_5 checks vice versa. TB_6 checks if the domains of P_i and P_j are the same. TB_7 *domain*(P_i) is subclass of *domain*(P_j) and TB_8 checks vice versa. Similarly, TB_9 checks if the ranges of P_i and P_j are the same. TB_{10} checks *range*(P_i) is subclass of *range*(P_j) and TB_{11} checks vice versa.

3.2 Model preparation

Because we are using supervised training techniques, it is necessary to collect annotations to train the learning algorithms.

We have asked experts from 4 DBpedia chapters to manually inspect mappings. For selecting the mappings that can possibly contain errors, we have used our previous assumption that when there is a high number of same subject and same object combinations with different properties in two different datasets of DBpedia, this could be because of a wrong mapping. Thus, we selected such mappings from 4 combinations of DBpedia datasets (EN-ES, EN-DE, EN-NL, EN-GR) and asked the language-pair experts to annotate if the mappings were correct or not. The instructions provided to the language-pair experts are available online in the following link[3].

To facilitate the annotation, for each mapping we provided the Wikipedia infobox template name, infobox key, and the property it is mapped to for each of the language. A snippet from the English-Spanish annotation table is shown in Table 3.

For sake of clarity, we will describe the process for the English-Spanish (EN-ES) case. We distinguish between two kinds of objects: IRIs and literals. We start with the literals case, in which 226 annotations have been provided manually by contributors from our institutions, fluent in both languages. Then, we trained a predictive model with these annotations. The data file is publicly available at https://www.openml.org/d/40742, were you also can see a statistic summary for each variable in the dataset. The training set contains 182 mappings annotated as "Correct" and 44 as "Incorrect". Therefore, the simplest classifier (known as ZeroR), which assigns the most popular class value, has an accuracy of 64.29%. This classifier establishes the baseline value that must be enhanced by our model.

We tested several classifiers using a 10-fold cross-validation. Each fold was stratified, that is, keeping class proportions. We used the weka environment (version 3.8.1) [9] and the classifiers use default parameters unless otherwise stated.

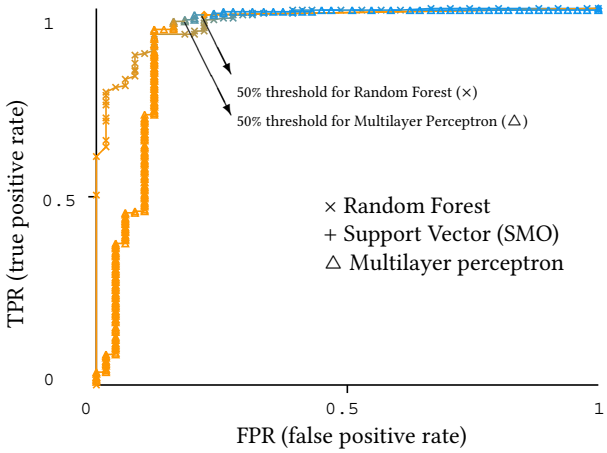

Figure 1: ROC curve for relevant classifiers.

The best results for accuracy (correctly classified instances) were for Random Forest (93.36%), Multilayer Perceptron (94.25%) and Support Vector classifier (93.36%). Figure 1 shows the ROC curve for these classifiers. In this figure (and also in columns 'ROC Area' in table 5) we can see that Random Forest has the highest ROC AUC (Area Under the Curve) and, therefore, can be considered the best classifier among the ones considered.

This figure also shows the specific point in which these classifiers reach a 50% threshold. These points are the "working points" for the non-penalty classifiers, that is, the points for which the cost/benefit curve reaches its minimum value. If we are interested in penalizing the false positives (i.e., a higher value than the value assigned to false negatives) the working point of the classifier will move to another point in this curve.

Table 4 summarizes some relevant classifier output values for these classifiers. Table 5 shows the detailed accuracy by class as well as the confusion matrix.

4 RESULTS

4.1 Using the predictive model

4.1.1 Prediction on IRIs. If we apply this predictive model to objects being IRIs, the accuracy (correctly classified instances) is 95.00%, very similar to the 93.36% achieved when objects are literals. Notice that this IRIs dataset, or any other IRI datatset, has not been "seen" by the model in its training. The dataset (80 instances, 71 Correct and 9 Incorrect) is publicly available[4].

4.1.2 Prediction on datasets in other languages. It would be great to have a unique model (trained with data annotated in an specific language) capable of predicting incorrect mappings not only in its own language but in other languages. This would be

[3] http://goo.gl/M1go5S

[4] https://www.openml.org/d/40744

Table 3: A snippet from English-Spanish mapping annotation

Template(en)	Infobox_French_commune	Infobox_company	Infobox_mountain	Infobox_album
Attribute(en)	elevation_m	foundation	elevation_m	Artist
Template(es)	Ficha_de_entidad_subnacional	Ficha_de_organización	Ficha_de_montaña	Ficha_de_álbum
Attribute(es)	elevación_media	fundación	Elevación	productor
Prop(en)	dbo:elevation	dbo:foundingYear	dbo:elevation	dbo:artist
Prop(es)	dbo:height	dbo:formationYear	dbo:prominence	dbo:producer
Annotation	Wrong mapping (note that in this is case, `dbo:elevation` was the expected property to be used with mountains etc.)	Wrong mapping (note that sometimes the ontology has similar properties for the same thing but this is still a wrong mapping)	Wrong mapping	Correct (it happens that the producer is the same person as the artist in some albums by coincidence)

Table 4: Summary of classifiers output.

	Random Forest	Multilayer Perceptron	SMO
Correctly Classified Instances	211 (93.36%)	213 (94.25%)	211 (93.36%)
Incorrectly Classified Instances	15 (6.64%)	13 (5.75%)	15 (6.64%)
Kappa statistic	0.7865	0.8117	0.7748
Mean absolute error	0.1101	0.0641	0.0664
Root mean squared error	0.2288	0.2276	0.2576
Relative absolute error	34.8987%	20.3324%	21.0402%
Root relative squared error	57.7554%	57.4747%	65.0442%
Total Number of Instances	226	226	226

some sort of multilingual predictor. Notice that the manual annotation of each dataset is a high-specialized task that requires humans with an excellent knowledge of the two languages involved. A unique model would save a lot of human work.

We have applied this specific predictive model (EN-ES-lit) to the dataset ES-DE-IRI containing 110 annotations. The accuracy (correctly classified instances) is 87.28%, very similar to the 85.71% achieved for EN-ES-IRIs, but not as good as the EN-ES-lit with 93.36%. The dataset is publicly available at https://www.openml.org/d/40743.

However, the English-Dutch literals (EN-NL-lit), with 83 annotations (35 Incorrect, 48 Correct) has an accuracy of only 61.45% If we create a predictive model for this data we get an accuracy of 71.08%.

For Dutch (EN-NL-IRIs), with 28 annotations (19 Incorrect, 9 Correct) has an accuracy very low as well, only 67.86%. A predictive model with this dataset would get an accuracy of 100%. A detailed analysis, changing the aleatory seed, produces accuracy values around 94%. This dependency on the seed indicates that the number of instances is too low.

For EN-GR-lit, with 64 annotations (30 Incorrect, 33 Correct) we get an accuracy of 77.78%. If we create a predictive model for this data we get an accuracy of 73.02%. All this information is condensed in table 8, were we can see that for the 4 language-pairs studied the model for IRIs is always better than the model for literals. Besides, we can see that the EN-ES-lit model is good predicting incorrect mappings for some language-pairs (such as ES-DE-IRI or EN-GR-IRI), but not so good for another language-pairs (such as EN-NL-lit or EN-NL-IRI). Our conclusion is that is not feasible to have a unique predictive model for all the language-pairs.

4.2 Optimizing the model

We have computed a Principal Component Analysis over the EN-ES-lit dataset. This analysis produces a new set of attributes, linear combination of the initial attributes, ranked by its contribution to the data variance, from more relevant to less relevant. Specifically, keeping a variance of 95%, we can move from a dataset with 23 attributes to a dataset with 13 attributes (12 numeric attributes and 1 nominal attribute). If we compute a new model using again Random Forest we achieve an accuracy of 92.4779%, very close to the 93.3628% achieved with all the attributes. Table 9 shows the effect of the progressive elimination of less relevant attributes on the accuracy of the predictive model. The 'Order' column is the variable ID, from lowest to highest variance. For instance, the first row in this table shows that the effect of removing the first lowest variance PCA attribute, having a 0.55 variance (indeed std. dev.), produces a predictive model with accuracy 93.8053%. The second row shows that the effect of removing the 2 lowest variance PCA attributes, in which the second attribute has a variance (std. dev.) of 0.688, produces a predictive model with accuracy 93.8053%. In this table we can see that the effect of removing the 3 lowest variance PCT attributes is the same: a model accuracy of 93.8053%. However, removing the 4 lowest we get lower accuracy (93.3628%), that maintains this value when removing the 5th and 6th lowest variance PCA attributes. But, when we remove the 8th we get an increment in the accuracy. If we remove more PCA attributes we get lower values. The last row shows the effect of removing all the PCA attributes except the one with the highest variance. Using this unique attribute we get a model with accuracy 80.9735%.

It is remarkable that with a reduction from 23 to 4 attributes, the Random Forest classifier obtains a slightly better accuracy (93.8053%) than the initial model with 23 attributes (accuracy 93.3628%).

If we repeat the PCA keeping a variance slightly lower than before, specifically to 90%, we get now 10 PCA attributes. En table 9 we can see that we can remove the 6 attributes with lowest variance while getting a predictive model with accuracy 93.8053%. Again, we get a model with only 4 attributes that produces a predictive model with accuracy 93.8053%.

Figure 2 shows a 2D projection of the 4D PCA space after the optimization. Although there is no clear separation for the classes, despite the 93% accuracy of the model, we can see a clear concentration (cluster) for Incorrect instances (dashed black ellipse).

Table 5: Detailed accuracy data and confusion matrix for the classifiers.

SMO

Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.750	0.022	0.892	0.750	0.815	0.779	0.864	0.718	Incorrect
	0.978	0.250	0.942	0.978	0.960	0.779	0.864	0.939	Correct
Avg.	0.934	0.206	0.932	0.934	0.931	0.779	0.864	0.896	

Confusion matrix

	Pred. Correct	Pred. Incorrect
True Correct	178	4
True Incorrect	11	33

Multilayer Perceptron

Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.818	0.027	0.878	0.818	0.847	0.812	0.934	0.854	Incorrect
	0.973	0.182	0.957	0.973	0.965	0.812	0.934	0.979	Correct
Avg.	0.942	0.152	0.941	0.942	0.942	0.812	0.934	0.955	

Confusion matrix

	Pred. Correct	Pred. Incorrect
True Correct	177	5
True Incorrect	8	36

Random Forest

Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.818	0.038	0.837	0.818	0.828	0.787	0.966	0.907	Incorrect
	0,962	0,182	0,956	0,962	0,959	0,787	0,966	0,991	Correct
Avg.	0.934	0.154	0.933	0.934	0.933	0.787	0.966	0.975	

Confusion matrix

	Pred. Correct	Pred. Incorrect
True Correct	175	7
True Incorrect	8	36

Table 6: Summary of the results of applying the predictive model (EN-ES literals) to the IRIs dataset.

Correctly Classified Instances	12 (85.7143%)
Incorrectly Classified Instances	2 (14.2857%)
Kappa statistic	0.6889
Mean absolute error	0.2479
Root mean squared error	0.3102
Total Number of Instances	14

This concentration could provide the separation between classes required for such a good classification.

5 RELATED WORK

Zaveri *et al.* [16] present a comprehensive systematic review of data quality assessment methodologies applied to LOD. They have extracted 18 quality dimensions and a total of 110 objective and subjective quality indicators. The work presented in this paper is related to the conciseness dimension of the intrinsic data quality. The survey does not contain any specific metric for measuring the inconsistencies in the mappings.

Dimou *et al.* [5] propose a test-driven approach for assessing the mappings and semi-automatic mapping refinements based on the results of the quality assessment. However, in contrast to the work presented in this paper the quality assessment is performed based on the mapping definitions before the RDF data is produced. The work presented in this paper uses a data-driven approach using

Table 7: Detailed accuracy data and confusion matrix for the predictive model (EN-ES literals) on the IRIs dataset.

Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.800	0.111	0.800	0.800	0.800	0.689	0.978	0.967	Incorrect
	0.889	0.200	0.889	0.889	0.889	0.689	0.978	0.989	Correct
Avg.	0.857	0.168	0.857	0.857	0.857	0.689	0.978	0.981	

Confusion matrix

	Pred. Correct	Pred. Incorrect
True Correct	8	1
True Incorrect	1	4

Table 8: Summary of the prediction accuracy for other language-pairs.

		EN-ES		ES-DE		EN-NL		EN-GR	
		lit	IRI	lit	IRI	lit	IRI	lit	IRI
Accuracy	Ad hoc model	93.36%	**95.00%**	N.A.	**96.36%**	**71.08%**	**~94%**	73.02%	**89.71%**
	En-ES-lit model		65.00%	N.A.	87.28%	61.45%	67.86%	**77.78%**	88.24%
Annotations	Total instances	211	80		110	83	28	63	68
	'Correct' instances	175	71		102	35	9	33	44
	'Incorrect' instances	36	9		8	48	19	30	24
Number of mappings		799	4979		4999	1329	4971	328	2785

Table 9: Principal Components Analysis (PCA). Effect of reducing the number of PCA attributes on the accuracy of the predictive model.

Order	Data variance			
	95%		90%	
	Std.dev.	Accuracy	Std. dev.	Accuracy
1	0.55	93.8053%	0.779	93.8053%
2	0.688	93.8053%	0.91	93.3628%
3	0.779	93.8053%	0.938	93.3628%
4	0.91	93.3628%	0.996	93.3628%
5	0.938	93.3628%	1.047	92.9204%
6	0.996	93.3628%	1.074	**93.8053%**
7	1.047	92.9204%	1.207	92.9204%
8	1.074	**93.8053%**	1.277	88.4956%
9	1.207	92.9204%	1.939	80.9735%
10	1.277	88.4956%		
11	1.939	80.9735%		

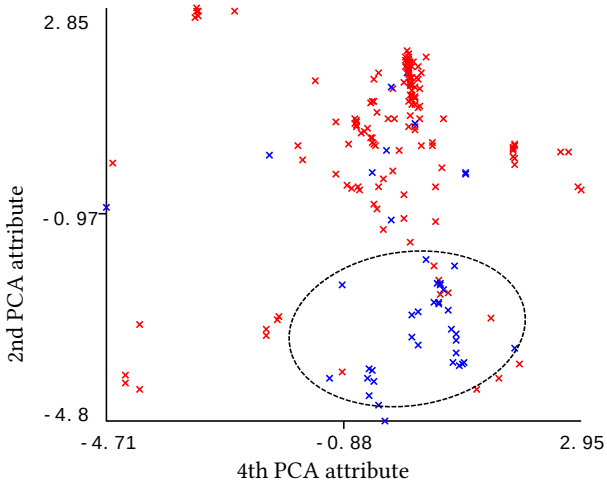

Figure 2: Selected projection to 2D of the 4D optimized PCA space. Blue crosses are Incorrect predictions. The ellipse shows a high concentration of Incorrect predictions that could explain the good classification of the model.

multiple datasets and the problem addressed in this paper can not be addressed only by analyzing the mapping definitions.

Paulheim [13] presents a data-driven approach to discover problems in mappings as well as in the ontology. This approach uses the ontology to check logically inconsistent statements, identify the mapping that was used to generate each inconsistent statement, and group the inconsistencies by mappings. Finally, a score is computed for each mapping depending on the frequency of inconsistent mappings. This is the closest to our work considering the objective. However, the approach presented in this paper is based on annotations by the experts and supervised machine learning algorithms.

Previous work that are not focused on evaluating the mappings but rather on data has used variety of approaches such as statistical

methods [4, 14], outlier detection [3, 6, 12], using external sources of knowledge [7, 11], crowdsourcing [1], and gamification [10]. Such approaches detect the errors in data but do not identify the mappings that caused those errors. Further, most of those work are focused on detecting errors in a single dataset and not on detecting inconsistencies among a set of datasets.

6 CONCLUSIONS

We have analyzed, from a data-driven perspective, the mappings that convert non-structured data to linked data. The method shown in this paper provides 22 numeric features for each mapping in a language-pair basis, and it is applied to several chapters of DBpedia. Additionally we provide non numeric information about each mapping to people fluent in each language-pair, and we ask them to decide if the mapping is correct or incorrect. With the numeric features we trained different supervised learning models, and we selected the random forest as the most appropriate.

The effort required to annotate a given language-pair is remarkable, but annotating a minimal part of all the mappings we achieve a 93% accuracy. A Principal Component Analysis can reduce the number of features to only 4 (linear combinations of the previous 22 features), keeping 93% accuracy, and a 2D projection allow us to see a concentration of 'Incorrect' data in a specific area.

We also explore the possibility of having a unique predictive model for all the language-pairs. However, experiments show that is better to have a model per language-pair. Results also show that models created from mappings having IRIs as objects (IRI models) are better than models created from mappings having literals as objects (lit models). The main reason for this could be the equivalence checking of the objects. When the objects are IRIs, the equivalence checks are more accurate because they are explicit owl:sameAs relations. But when checking literals (e.g., strings, numbers, dates, etc.) small syntactic differences could lead to non-equivalent values, even though they represent the same value.

Concerning challenges, one key problem is data incompleteness. For instance, the attribute 'death_date' is correctly mapped to `dbo:deathDate` but most of the data only contain the year. However, in our approach it will match with the `dbo:deathYear` property of the other dataset because both values are death years. In such cases, the mapping could be falsely identified as incorrect because of the inaccuracies in data. Nevertheless, it is important to note that if the mapping is corrected as suggested it will result in more accurate data as the data contains the death year rather than death date.

Another challenge is that some of the annotations require considerable domain knowledge to decide if a mapping is correct or incorrect. For instance, a template for F1 Racing or musical genres, uses a lot of terms specific to the given domain which annotators are not familiar with. This challenge can be mitigated by providing human annotators with a richer user interface. Currently, the annotators see a set of features on a spreadsheet. Some of the annotators claimed a better interface, with more features or examples based on data mappings.

Our future plans include to measure the number of triples with high probability of being incorrect due to an incorrect mapping, as well as applying these predictive models to assist people in the mapping process. The model allows us to assign a cost matrix, for instance to penalize false negatives over false positives, in order to minimize recall.

ACKNOWLEDGMENTS

This work was partially funded by the Spanish MINECO Ministry (projects RTC-2016-4952-7 and TIN2013-46238-C4-2-R), the BES-2014-068449 grant, and grants from the EU's H2020 Programmes for the ALIGNED project (GA 644055). Also, this work has been partially funded by the project Data 4.0 (TIN2016-78011-C4-4-R), from the Spanish State Investigation Agency of the MINECO and FEDER Funds. Further, we would like to thank Enno Meijers, Gerard Kuys, Roland Cornelissen, Gerald Wildenbeest, and Julia Bosque Gil for their contribution to the mapping annotation tasks.

REFERENCES

[1] Maribel Acosta, Amrapali Zaveri, Elena Simperl, Dimitris Kontokostas, Sören Auer, and Jens Lehmann. 2013. Crowdsourcing linked data quality assessment. In *International Semantic Web Conference*. Springer, 260–276.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. *The Semantic Web* (2007), 722–735.

[3] Jeremy Debattista, Christoph Lange, and Sören Auer. 2016. A Preliminary Investigation Towards Improving Linked Data Quality Using Distance-Based Outlier Detection. In *Joint International Semantic Technology Conference*. Springer, 116–124.

[4] Jeremy Debattista, Santiago Londoño, Christoph Lange, and Sören Auer. 2015. Quality assessment of linked datasets using probabilistic approximation. In *European Semantic Web Conference*. Springer, 221–236.

[5] Anastasia Dimou, Dimitris Kontokostas, Markus Freudenberg, Ruben Verborgh, Jens Lehmann, Erik Mannens, Sebastian Hellmann, and Rik Van de Walle. 2015. Assessing and refining mappingsto rdf to improve dataset quality. In *International Semantic Web Conference*. Springer, 133–149.

[6] Daniel Fleischhacker, Heiko Paulheim, Volha Bryl, Johanna Völker, and Christian Bizer. 2014. Detecting errors in numerical linked data using cross-checked outlier detection. In *International Semantic Web Conference*. Springer, 357–372.

[7] Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. 2015. DeFactoTemporal and multilingual Deep Fact Validation. *Web Semantics: Science, Services and Agents on the World Wide Web* 35 (2015), 85–101.

[8] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. 2013. Evaluating entity linking with Wikipedia. *Artificial intelligence* 194 (2013), 130–150.

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

[10] Markus Ketterl, Lars Knipping, Nadine Ludwig, Robert Mertens, Jörg Waitelonis, Nadine Ludwig, Magnus Knuth, and Harald Sack. 2011. Whoknows? evaluating linked data heuristics with a quiz that cleans up dbpedia. *Interactive Technology and Smart Education* 8, 4 (2011), 236–248.

[11] Pablo N Mendes, Hannes Mühleisen, and Christian Bizer. 2012. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*. ACM, 116–123.

[12] Heiko Paulheim. 2014. Identifying Wrong Links between Datasets by Multi-dimensional Outlier Detection.. In *WoDOOM*. 27–38.

[13] Heiko Paulheim. 2017. Data-driven Joint Debugging of the DBpedia Mappings and Ontology. In *European Semantic Web Conference*. 1–15.

[14] Heiko Paulheim and Christian Bizer. 2014. Improving the quality of linked data using statistical distributions. *International Journal on Semantic Web and Information Systems (IJSWIS)* 10, 2 (2014), 63–86.

[15] Mariano Rico, Nandana Mihindukulasooriya, and Asunción Gómez-Pérez. 2016. Data-Driven RDF Property Semantic-Equivalence Detection Using NLP Techniques. In *EKAW Proceedings, LNCS 10024*. Springer International Publishing, 797–804. https://doi.org/10.1007/978-3-319-49004-5_51

[16] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. 2016. Quality assessment for linked data: A survey. *Semantic Web* 7, 1 (2016), 63–93.