Learning Conflict Resolution Strategies for Cross-Language Wikipedia Data Fusion

Volha Bryl
Data and Web Science Group
University of Mannheim
volha@informatik.uni-mannheim.de

Christian Bizer
Data and Web Science Group
University of Mannheim
chris@informatik.uni-mannheim.de

ABSTRACT

In order to efficiently use the ever growing amounts of structured data on the web, methods and tools for quality-aware data integration should be devised. In this paper we propose an approach to automatically learn the conflict resolution strategies, which is a crucial step in large-scale data integration. The approach is implemented as an extension of the Sieve data quality assessment and fusion framework. We apply and evaluate our approach on the use case of fusing data from 10 language editions of DBpedia, a large-scale structured knowledge base extracted from Wikipedia. We also propose a method for extracting rich provenance metadata for each DBpedia fact, which is later used in data fusion.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

Data fusion, conflict resolution, learning, provenance metadata

1. INTRODUCTION

In the recent decades the amount of structured machine readable data available on the Web is growing rapidly. One important example is the Linked Open Data (LOD) initiative [8], which is publishing and interlinking open datasets on the web. The amount of data in LOD datasets is of the order of tens of billions of fact statements, spanning from cross-domain encyclopedic knowledge bases, to biomedical, geospatial, product and service data, media content, etc. Potentially, the LOD resources are extremely useful for applications in business, science and public domains. However, the quality of the Linked Open Data sources varies greatly across domains and single datasets, making the efficient use of data problematic. Quality problems come from data publishing errors, automatic data extraction methods, as well as from data being outdated or inaccurate [1, 9]. Another aspect is the lack of data consistency: same real world entities are described in different datasets using different vocabularies and data formats, and the descriptions often contain conflicting values.

In order for applications to benefit from using multiple heterogeneous web data sources, two key problems have to be addressed: data quality and data integration. The two are tightly interrelated: data integration of multiple heterogeneous data sources aims at improving data quality as an integrated dataset is expected to be more complete and consistent. On the other hand, the choice of conflict resolution strategies depends on the quality of data coming from different sources (e.g. the strategy could be to prefer most trusted or most recent data).

We are interested in *data fusion*, a process of integrating data describing the same real world entity coming from multiple sources into a single consistent representation [3]. The key challenge of data fusion is *resolving conflicts* in data. The related work in the area concerns fusing data within relational databases [3], or specific resolution strategies and their combinations, e.g. trustworthiness and voting [5, 7]. Some recent works go in the direction of large-scale web data fusion, for instance, by automatically selecting sources for data integration [6]. However, to the best of our knowledge, the problem of automatically learning conflict resolution strategies is not well investigated. We find this topic particularly promising, as manually defining a conflict resolution strategy requires domain knowledge and understanding of the data, is time-consuming, and does not guarantee an optimal result.

In this paper we focus on the use case of DBpedia [10][1], a large-scale structured multi-lingual cross-domain knowledge base automatically extracted from Wikipedia. The latest DBpedia 3.9 contains 2.46 billion facts describing 12.6 million unique things, and is a widely used knowledge resource with around 5000 downloads a year. The data is extracted from Wikipedia infoboxes (tables usually found in upper right part of a Wikipedia page), page categories, interlanguage links and many more. Data is extracted from 119 Wikipedia language editions, and is represented as a distinct language edition of the knowledge base. This leads to lots of data conflicts across descriptions of the same real world entity in different languages, inherited from Wikipedia, as well as data conflicts that appear due to extraction errors [1].

[1]DBpedia community project – `http://dbpedia.org`

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2578999.

Our goal is to fuse DBpedia, and consequently, Wikipedia data across languages, thus coming up with even more large-scale and higher quality cross-domain knowledge base.

We build on the previous work, extending the Sieve data quality assessment and fusion framework [11] with the learning capabilities for choosing conflict resolution strategies. In addition, we present an approach for extracting rich provenance metadata per each DBpedia fact, which we later use in the process of data fusion.

Our evaluation results show that automatically learning conflict resolution strategies leads to higher accuracy of the integrated data, and thus facilitates building large-scale high quality knowledge bases.

The rest of the paper is structured as follows. We begin with introducing the Sieve data quality assessment and fusion framework in Section 2, which is the starting point of our work. Then in Section 3 our approach to learning conflict resolution strategies is presented. Section 4 concerns fusing multi-lingual DBpedia data, where Section 4.1 details the approach we developed for extracting provenance metadata, Section 4.2 introduces the gold standard we use for learning, and Section 4.3 presents and discusses the experimental results.

```
dbp:Amsterdam dbpedia-owl:population "820654" en:Amsterdam:population
dbp:Amsterdam dbpedia-owl:population "758198" ca:Amsterdam:population
dbp:Amsterdam dbpedia-owl:population "820654" it:Amsterdam:population
en:Amsterdam:population dpb-meta:lastedit "2013-01-13T14:52:13Z"
ca:Amsterdam:population dpb-meta:lastedit "2009-06-14T10:36:30Z"
it:Amsterdam:population dpb-meta:lastedit "2013-03-24T17:04:16Z"
```

Figure 1: Data Fusion with Sieve: input data.

```
<QualityAssessment>
  <AssessmentMetric id="sieve:recency">
    <ScoringFunction class="TimeCloseness">
      <Param name="timeSpan" value="500"/>
      <Input path="?GRAPH/dpb-meta:lastedit"/>
    </ScoringFunction>
  </AssessmentMetric>
</QualityAssessment>
<Fusion>
  <Class name="dbpedia-owl:PopulatedPlace">
    <Property name="dbpedia-owl:population">
      <FusionFunction class="KeepFirst" metric="sieve:recency"/>
    </Property>
  </Class>
</Fusion>
```

Figure 2: Data Fusion with Sieve: specification.

2. DATA FUSION WITH SIEVE

In this section we summarize the functionality of Sieve – Linked Data Quality Assessment and Fusion framework[2] [11], which provides the basis of our work and experiments. Sieve allows its users to manually define conflict resolution strategies and quality assessment metrics to be used for each data property, using an XML-based specification language. For instance, one can specify the following conflict resolution strategy for cities: take the most recent population value, the most frequent value for founding year, and the average value for area.

Sieve takes as input two or more RDF[3] data sources, along with the data provenance information. Sieve assumes that schema and object identifiers have been normalized, namely, if two descriptions refer to the same real world object then they have the same identifier (URI), and if two properties refer to the same real world attribute then there should be two values for the same property URI for a given subject URI. Each property value in the input is expressed by a quad *(subject,property,object,graph)* where the *graph* is a named graph, which is used to attach provenance information to a fact or a set of facts. For an example see Figure 1, where the input data for the population of Amsterdam coming from three different DBpedia editions along with the last edit date information are given. Note that the 4th quad component, the provenance graph for *lastedit* property is omitted due to space reasons.

The example of the specification in Figure 2 illustrates how quality assessment metrics and fusion functions are defined: recency assessment metric uses the *last update date* of a fact, which is then transformed by *TimeCloseness* scoring function into a numeric value normalizing it by a range parameter. In the data fusion configuration, a conflict resolution function for each data property is defined: the fusion function for the population property of a city or town is configured to use *KeepFirst* fusion function applied to *recency* quality assessment metric.

The output of the data fusion module is a set of quads, each representing a fused value of a subject-property pair, with the fourth component of the quad identifying the named graph from which a value has been taken. Sieve implements a library of basic quality assessment scores (e.g. for measuring recency or trust) and fusion functions (voting, maximum, average, functions based on quality scores). The framework is extensible, that is, users can define their own scoring and fusion functions.

3. LEARNING CONFLICT RESOLUTION STRATEGIES

As we have already mentioned, one of the drawbacks of Sieve is that the conflict resolution strategies should be defined manually, that is, a user has to specify which fusion function based on which quality assessment score to use for each data property. Such an approach requires good understanding of the input data and does not guarantee an optimal result. To overcome this, we have developed the Sieve Fusion Policy Learner, an extension of Sieve that learns optimal fusion functions using a ground truth dataset.

The Fusion Policy Learner takes as input the ground truth dataset (or gold standard, a dataset which provides the correct data values for each property of an entity), and an XML-based specification, in which the user specifies possible conflict resolution strategies for each data property. See Figure 3 for an example, where the list of possible fusion functions for the *population* property is specified.

The learning algorithm then selects the fusion function that minimizes the error with respect to the gold standard. The algorithm first detects, based on the gold standard dataset, whether the values to fuse are numeric or nomi-

[2] `http://sieve.wbsg.de`

[3] RDF (Resource Description Framework) allows representing data in the form of subject-predicate-object expressions – `http://www.w3.org/RDF`

```
<Fusion name="Multilingual DBpedia data fusion">
  <Class name="dbpedia-owl:PopulatedPlace">
    <Property name="dbpedia-owl:population">
      <FusionFunction class="Average"/>
      <FusionFunction class="Maximum"/>
      <FusionFunction class="Voting"/>
      <FusionFunction class="KeepFirst" metric="sieve:recency"/>
      <FusionFunction class="KeepFirst" metric="sieve:english"/>
      <FusionFunction class="KeepFirst" metric="sieve:authactivity"/>
      <FusionFunction class="KeepFirst" metric="sieve:propactivity"/>
      <FusionFunction class="KeepLowest" metric="sieve:authage"/>
    </Property>
  </Class>
</Fusion>
```

Figure 3: Fusion Policy Learner: specifying a list of possible fusion functions for learning.

nal (e.g. strings or URIs)[4]. Then, for numeric values one of the following strategies is applied:

1. the fusion function that minimizes the mean absolute error with respect to the gold standard is selected, or

2. given a maximum error threshold (e.g. 0.05, which corresponds to 5%), the fusion function that maximizes the number of values that deviate from a respective gold standard value no more than by a threshold, is selected.

More formally, let us consider k fusion functions applied to values of a property p of m entities (e.g. to population of m cities). Assume we have a set of gold standard values $g_j, j = \overline{1,m}$ and the corresponding data values n_{ij}, where $i = \overline{1,k}$ identifies a fusion function and j identifies the entity. Also assume that the maximum error threshold is defined to be ϵ. Then, the mean absolute error for the i-th fusion function $abs(i)$ and the optimal fusion functions for the 1st and the 2nd strategies i_1^* and i_2^* are defined as follows:

$$i_1^* = \arg\min_{i=\overline{1,k}} abs(i),\ abs(i) = \frac{\sum_{j=1}^{m} \frac{|n_{ij} - g_j|}{g_j}}{m},$$

$$i_2^* = \arg\max_{i=\overline{1,k}} \sum_{j=1}^{m} I_\epsilon \left(\frac{|n_{ij} - g_j|}{g_j} \right),\ I_\epsilon(x) = \begin{cases} 1 \text{ if } x \leq \epsilon \\ 0 \text{ otherwise.} \end{cases}$$

In case of nominal values, the fusion function that produces the maximum number of exact matches with the gold standard values is selected. The strategy to use and the maximum error threshold are parameters defined by the user in the input specification.

In the following section we present the application of the above approach to the multi-lingual DBpedia use case.

4. FUSING DBPEDIA DATA

Our goal is to fuse DBpedia data across languages, thus coming up with even larger-scale and higher quality cross-domain knowledge base. Applications built on top of DBpedia would benefit from such an integrated dataset, as both coverage and data quality will be improved. Some of these applications already try to combine data from several DBpedia language editions: for example, QAKiS question answering system [4] uses data from English, German and French DBpedia, but leaves open the problem of data conflict resolution.

To evaluate the Fusion Policy Learner, we have experimented with data about populated places (cities and towns) from top 10 DBpedia language editions[5], which resulted in 610,017 entities.

Conflict resolution often relies on *data provenance* information, that is, information on where the data comes from, who the author is, when the data was created, updated, etc. Developing methods for extracting and representing provenance metadata is an important research topic per se, especially promising in the case of collaboratively created resources (like Wikipedia, and hence, DBpedia), in which edit logs and information about each single author are often available.

In DBpedia, the provenance metadata is not available as part of the datasets, however, every DBpedia entity is linked to the corresponding Wikipedia page, for which the detailed edit history is available. In the first experiments on fusing DBpedia data with Sieve [11] a simple strategy with respect to provenance metadata was followed: for each source page the last page update timestamp from the Wikipedia dumps was extracted and included in the provenance graph. However, this is only an approximation of the data recency as the last update may concern page layout or free text content of the page, and thus does not say anything about the infobox values – crucial for DBpedia – which may be edited long time ago with respect to the last page update, and thus outdated. In the following we explain how we extracted the provenance information for each fact (DBpedia triple) separately in order to use it in the process of data fusion.

4.1 Provenance metadata extraction

As a source of provenance metadata we used *Wikipedia revision history dumps* made available for download by Wikimedia foundation[6]. The revision dumps file names contain *pages-meta-history* substring; the dumps are sliced into chunks and compressed as the size of the full revision history of a popular Wikipedia language edition is of the order of terabytes. For instance, the size of English Wikipedia revision history is greater than 6 terabytes, the German revision dumps expand to more than 2 terabytes, etc. Due to the challenges the size of the dumps presents, we have run the metadata extraction process for the subset of DBpedia entities that describes populated places. Wikipedia revision history is stored in the XML format, and contains a sequence of revisions for each page, with timestamp and author (user name or IP in case the author is not a registered Wikipedia user) attached to each revision, and the full text of each revision.

The key idea of our provenance metadata extraction approach is to go through all revisions of a page, from the latest one to the earliest, and compare infoboxes between neighbouring revisions in order to detect changes relevant

[4]Detecting more complex data types, such as dates, currencies, double versus integer numbers, etc. is among the future work directions.

[5]10 languages used in our experiments are English, French, German, Italian, Dutch, Spanish, Russian, Polish, Portuguese and Catalan. "Top 10" refer to the 10 largest DBpedia editions.

[6]See e.g. `http://dumps.wikimedia.org/enwiki/latest` for the latest English Wikipedia dumps.

```
ru:Mannheim:population  author  EmausBot
                        autheditcnt 1,136,639
                        propeditcnt 3
                        authregdate 2009-12-18T02:08:09Z
                        lastedit    2011-12-22T00:50:21Z
                        authbot true
nl:Mannheim:population  author  Joopwiki
                        autheditcnt 106,899
                        propeditcnt 1
                        authregdate 2007-04-05T08:54:19Z
                        lastedit    2007-12-09T16:41:06Z
                        authsysop   true
```

Figure 4: Provenance metadata: example.

for DBpedia data. We have developed a tool that parses the Wikipedia revision history dumps combining a standard Java SAX parser[7] with the DBpedia Extraction Framework[8], an open source tool to extract DBpedia data from Wikipedia. As a results, for each DBpedia fact we extract not only the last edit timestamp and the name or IP address of the Wikipedia author that made it (*lastedit* and *author* in Figure 4), but the same metadata about all edits of a property instance, namely, *edit traces* for each DBpedia property of each entity. From the edit traces, we extract the total number of edits of a property (*propeditcnt* in Figure 4).

The source code of the provenance metadata extraction tool alone with some examples of the output is available online[9]. More examples of the extracted metadata are included in the example distributed with the Fusion Policy Learner source code[10].

After having extracted the name or IP address of the author of each DBpedia triple, we obtain metadata regarding Wikipedia authors using the Mediawiki API[11]. Specifically, we extract the following meta properties:

- the total number of edits done by each author (*autheditcnt* in Figure 4),
- author's registration date (*authregdate*),
- whether the author was blocked and when: this information appeared to be irrelevant, as we found no blocked authors among those whose contributions on populated places ended up in DBpedia,
- groups the author belongs to (e.g. *authsysop* or *authbot*)[12].

For the populated places data subset for the 10 DBpedia language editions, information about 164,000 authors was extracted, more than half of them are anonymous users with no information other than IP address provided. However, the number of changes made by anonymous users that ended up in the considered subset of DBpedia is only around 2.3%, so the absence of the metadata for these authors does not have a significant impact on the data fusion experiments presented in Section 4.3.

Our work is similar to the one recently conducted at Google [2], in which temporally anchored infobox attribute data for English Wikipedia were extracted from the revision history, and used for vandalism detection. We could not reuse the results of [2] as (a) the extraction was done only for English, while we need more languages, (b) in their data infobox facts are not linked to DBpedia.

4.2 Selecting a gold standard

Constructing a gold standard to evaluate the data fusion results for a multi-lingual DBpedia is a non-trivial problem. The main difficulty follows from DBpedia cross-domain and encyclopedic nature: it is difficult to find a trusted data source which is as complete as DBpedia, and is *not based on Wikipedia*. Therefore, the viable approach is to perform a domain-specific evaluation, which we have done in our use case by limiting the scope to populated places, that is, the entities of the type *dbpedia-owl:PopulatedPlace* (which is one of the top DBpedia ontology classes in terms of number of instances). The populated places use case is still challenging as it contains numeric fast-changing attributes such as population, and so the datasets found on the web are likely to be outdated. In addition, the datasets containing information about large number of cities and towns across different countries are rarely available: certain high-quality datasets are limited to a specific country or region, as for instance, the US census[13] or the Eurostat[14] datasets.

We have selected GeoNames geographical database[15] to be a gold standard for our use case. The main reasons for this choice are the following:

- high coverage in a use case domain,
- high quality: the data is based on official sources, e.g. National Statistical Offices,
- high quality links between GeoNames and DBpedia,
- data openness and availability in a number of formats, including RDF.

For obtaining the links between GeoNames and Wikipedia (and hence, DBpedia) we used the *alternateNames* dataset available for download from the GeoNames web page. For countries, we had to map capitals to the corresponding DBpedia entities, as for the values of *capital* property strings but not GeoNames IDs were provided. This was done by first looking in DBpedia for the entities of type *PopulatedPlace* with names exactly matching the string values of *capital* property, and then manually fixing the cases for which mappings were not discovered.

While the GeoNames coverage in terms of instances is high, the number of properties is significantly smaller than in DBpedia, in which around 100 distinct properties are used with more than 5,000 instances of the *PopulatedPlace* class. Based on the data formats and the density of the values, we have selected the following GeoNames properties that can be mapped to the properties of the DBpedia *PopulatedPlace* class: *capital* (for countries), *country* (for cities and

[7] http://docs.oracle.com/javase/7/docs/api/javax/xml/parsers/SAXParser.html
[8] https://github.com/dbpedia/extraction-framework/wiki
[9] https://github.com/VolhaBryl/DBpedia-provenance
[10] https://github.com/wbsg/ldif/tree/master/ldif/examples/dbpedia-multilang
[11] http://www.mediawiki.org/wiki/API
[12] We do not use group meta information in the fusion experiments presented in this paper leaving it for the future work.
[13] http://www.rdfabout.com/demo/census
[14] http://eurostat.linked-statistics.org
[15] http://www.geonames.org

Table 1: DBpedia populated places: numbers of entities per language. Total number of entities in 10 editions is 1,893,445 (sum of row 2), while total number of unique entities is 610,017.

en	nl	es	ru	it	pl	fr	pt	de	ca
460,175	224,080	185,658	183,364	178,293	165,892	149,482	137,892	130,672	77,937

Table 2: DBpedia populated places: number of entities described exactly in 1, 2, 3, ... languages. 279,234 entities are present in English edition. 177,243 entities are described in 3 or more languages.

1	2	3	4	5	6	7	8	9	10
349,495	83,279	31,971	31,032	21,691	17,250	22,421	30,427	18,431	4,020

towns), *population*, *area* and *geo coordinates* (*longitude* and *latitude*).

4.3 Evaluation results

We have experimented with data about populated places (cities and towns) in top 10 DBpedia language editions, which resulted in 610,017 entities with 30% of them described in 3 or more languages. Table 1 provides statistics on the number of entities found in each of the 10 language editions. The total number of entities is 1,893,445, which is 3 times more than the number of distinct entities corresponding to a populated place (each of which is described in possibly many languages). The latter and the numbers presented in Table 2 show that the intersection between languages is high enough to make the use case interesting for the task of data fusion.

In the process of learning an optimal fusion policy we have experimented with the following configurations (see also the list of fusion functions in Figure 3 for the specification):

- *Average* – take the average value, used only for numeric properties;
- *Maximum* – take the maximum value, used only for numeric properties;
- *MostFrequent*, or *Voting* – take the most frequent value;
- *MostRecent* – take the most recent value with respect to the last edit timestamp;
- *English* – prefer values from the English DBpedia edition;
- *MostActive author* – prefer values from the author with the highest edit count;
- *MostActive property* – prefer values from the edition in which the property was edited the most;
- *MostExperienced author* – prefer values from the author who registered the earliest.

In order to experiment with different datasets, we have divided the data into subsets according to the country or population range as defined by the gold standard. For instance, the cities and towns with the population greater than 1,000 according to GeoNames formed *cities1000* dataset containing 68,926 unique entities; the *German* subset of *cities1000* contains 7,648 entities; the size of the world *country* dataset is 246; the size of the dataset of cities which according to GeoNames have at least half a million inhabitants is 684. In Table 3 we present the results for several data subsets. In the 3rd column the size of the dataset with respect to the considered property is reported; it can be lower than the overall dataset size (e.g. 243 instead of 246 countries) because in the gold standard some property values may be missing. We compare the mean absolute error of the optimal fusion function to the error we get when preferring data from the English DBpedia edition. The English Wikipedia/DBpedia edition is the largest and the most developed one, so preferring the English values might seem an obvious choice when manually defining the fusion strategy. Moreover, as currently many applications that use DBpedia rely only on the English edition, this is the strategy they, in principle, apply even if they do not explicitly perform data fusion.

As can be seen in Table 3, in some cases the error on the English DBpedia is already low (e.g. for German cities), however, the optimal strategy still gives an improvement. The high error for the country property of Brazilian cities (31% on English DBpedia) is due to the fact that this property is sometimes (wrongly) used by Wikipedia authors to express more granular information (e.g. *Northeast Region of Brazil*) and because the value is often not *Brazil* but *States of Brazil*, which is actually not the article about the country but rather a list of its states. The latter findings indicate that the presented data fusion approach is also useful for detecting (possibly problematic) patterns in the input data. The "most active user" strategy learned for the Brazilian case, actually, refers to preferring the changes done by the *Rei-bot* user of the Portuguese Wikipedia. The prevailing strategies learned by the Fusion Policy Learner are selecting maximum or the most frequent value.

As we already mentioned, the problem of automatically learning the fusion strategies is not well addressed in the literature, while our evaluation results show the advantage of such an approach even when using the simple learning strategy our module implements. Our next step is to experiment with more elaborate learning methods: to move from error minimization to more advanced machine learning techniques such as decision trees or genetic programming. All these techniques work well when enough labelled data (gold standard values) is available. Otherwise, active learning can be a solution: in this approach a human expert is involved in the loop and is proposed data items to be labelled as correct or incorrect, with the goal of the active learning algorithm to minimize the number of items to be labelled.

5. CONCLUSIONS

Given the number and the diversity of datasets available on the web today, integration of these data is crucial for their efficient use in applications. Data quality and consistency are among the most critical yet unsolved problems of the

Table 3: Results of automatically learning the optimal fusion strategy. Selection method for population: minimize mean absolute error.

Property	Dataset	Size	Fusion policy	Error, %	Error, %, en.dbp
populationTotal	cities1000-Germany	7,330	MostFrequent	0.3029	0.6796
populationTotal	cities1000-Netherlands	493	Maximum	2.1933	3.5714
populationTotal	countries	243	Maximum	2.1646	6.3485
populationTotal	cities1000-Brazil	1,247	MostActive property	2.2727	2.6913
country	cities1000-Italy	1,078	MostFrequent	0	1.206
country	cities1000-Brazil	1,119	MostActive author	9.8302	30.9205
country	cities1000-Germany	7,638	MostFrequent	0.0131	0.6415

web of data. Therefore, there is a need for methods and tools for the quality-aware data integration.

The focus of this work is on data fusion, which is a final step of the data integration process [3], aimed at fusing heterogeneous descriptions of the same real world entity coming from different sources. The key challenge of data fusion is resolving conflicts in data. In this paper we have addressed the problem of *automatically learning conflict resolution strategies*, a problem not well addressed in the literature but crucial for the large-scale data integration.

Our starting point is the previous work on Sieve, the data quality assessment and fusion framework for Linked Data, which we extend with learning capabilities. We have developed the Fusion Policy Learner module, which allows automatically learning an optimal combination of fusion functions for a set of data properties. The Sieve project is open source, and hence, the code of the Fusion Policy Learner is available online.

Our use case is data fusion accross multiple language editions of Wikipedia. In particular, we work with DBpedia, which is the Wikipedia's "structured twin". We experimented with the data about populated places (cities and towns) from 10 different language editions of DBpedia, and fused it using GeoNames data as a gold standard.

To widen the range of possible fusion strategies, we have used rich provenance metadata per each DBpedia fact detailing when and by whom each fact was created. Such provenance metadata was not available along with DBpedia datasets, and so we have developed a tool for extracting it from the Wikipedia revision history dumps made available for download by Wikimedia foundation.

The evaluation results show that automatically learning the conflict resolution strategies leads to accuracy improvements of the integrated dataset, even when using the simple learning strategy our module implements.

The future work directions concern exploring other learning techniques, as well as further work on the multiligual DBpedia case study and other LOD datasets towards cross-domain data fusion.

6. ACKNOWLEDGMENTS

The work presented in the paper is supported by the EU FP7 project *LOD2 – Creating Knowledge out of Interlinked Data*[16] (Ref. No. 257943).

7. REFERENCES

[1] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann. Crowdsourcing Linked Data quality assessment. In *12th International Semantic Web Conference (ISWC)*, 2013.

[2] E. Alfonseca, G. Garrido, J.-Y. Delort, and A. Penas. WHAD: Wikipedia historical attributes data – Historical structured data extraction and vandalism detection from the Wikipedia edit history. *Language Resources and Evaluation*, 47(4):1163–1190, 2013.

[3] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys*, 41(1):1:1–1:41, 2009.

[4] E. Cabrio, J. Cojan, F. Gandon, and A. Hallili. Querying multilingual DBpedia with QAKiS. In *Extended Semantic Web Conference (ESWC)*, volume Demo paper., 2013.

[5] X. L. Dong, L. Berti-Equille, and D. Srivastava. Data fusion: Resolving conflicts from multiple sources. In *Web-Age Information Management (WAIM)*, pages 64–76, 2013.

[6] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *PVLDB*, 6(2):37–48, 2012.

[7] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *3rd ACM International Conference on Web Search and Data Mining (WSDM)*, pages 131–140, 2010.

[8] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool Publishers, 2011.

[9] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An empirical survey of Linked Data conformance. *Journal of Web Semantics*, 14:14–44, 2012.

[10] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal*, 2014.

[11] P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: linked data quality assessment and fusion. In *EDBT/ICDT Workshops*, pages 116–123, 2012.

[16] `http://lod2.eu`