DBpedia FlexiFusion
The Best of Wikipedia > Wikidata > Your Data

Johannes Frey[1], Marvin Hofer[1], Daniel Obraczka[1], Jens Lehmann[1,2], and Sebastian Hellmann[1]

[1] Leipzig University (AKSW/KILT & database group) & DBpedia Association,
Leipzig, Germany {lastname}@informatik.uni-leipzig.de,
http://aksw.org/Groups/KILT & https://dbs.uni-leipzig.de
[2] Smart Data Analytics (SDA) Group, Bonn, Germany & Fraunhofer IAIS, Dresden,
Germany jens.lehmann@iais.fraunhofer.de

Abstract. The data quality improvement of DBpedia has been in the focus of many publications in the past years with topics covering both knowledge enrichment techniques such as type learning, taxonomy generation, interlinking as well as error detection strategies such as property or value outlier detection, type checking, ontology constraints, or unit-tests, to name just a few. The concrete innovation of the DBpedia FlexiFusion workflow, leveraging the novel DBpedia PreFusion dataset, which we present in this paper, is to massively cut down the engineering workload to apply any of the vast methods available in shorter time and also make it easier to produce customized knowledge graphs or DBpedias. While FlexiFusion is flexible to accommodate other use cases, our main use case in this paper is the generation of richer, language-specific DBpedias for the 20+ DBpedia chapters, which we demonstrate on the Catalan DBpedia. In this paper, we define a set of quality metrics and evaluate them for Wikidata and DBpedia datasets of several language chapters. Moreover, we show that an implementation of FlexiFusion, performed on the proposed PreFusion dataset, increases data size, richness as well as quality in comparison to the source datasets.

Keywords: data fusion, quality assessment, provenance
Stable Databus IRI: https://databus.dbpedia.org/dbpedia/prefusion

1 Introduction

From ancient history until today, being in possession of the right information at the right moment promised great rewards. From the movable types of the Gutenberg press to the long tail of information delivered by the WWW, we can cite ample examples in history where more adequate information delivery had a great effect on society. We certainly do not claim to have discovered such a disruptive technology as the movable types of the Gutenberg Press, which allowed effective production of different kind of books, however, we see our work as a step in the right direction of rapid production of movable knowledge graphs.

The concrete innovation of the DBpedia FlexiFusion approach is to massively cut down engineering workload to produce customized DBpedias. Our main use case here is the generation of richer language-specific DBpedias for the 20+ DBpedia chapters, which we demonstrate on the use case of the Catalan DBpedia[3] (cf. Section 5). Regarding further advances in data engineering, we see various additional uses that can benefit from the flexibility provided. In particular this flexibility concerns:

1. Flexibility of source selection via the DBpedia Databus[4]. In this paper, we load 140 DBpedia language-editions and Wikidata from the Databus. Beyond this, we already experimented with the inclusion of data from the Dutch and German national libraries via existing links and mappings in FlexiFusion.
2. A new format, which stores value options for triples including resolvable rich provenance information.
3. A flexible fusion approach to reduce and resolve available options to materialize new knowledge graphs, that are downward-compatible with the RDF standard. We list a short overview of previous fusion approaches that are applicable in Section 7.

In the next section, we introduce the DBpedia Databus as background, followed by the PreFusion dataset in Section 3. Section 4 describes the details of FlexiFusion. Subsequently, we show two usage scenarios and concrete configurations of FlexiFusion to produce custom fused datasets in Section 5 and evaluate our datasets w.r.t. data coverage and data quality in Section 6. We finish with related work, conclusions and a final discussion.

2 DBpedia Databus - The Digital Factory Platform

The Databus platform is developed via a use-case driven methodology. FlexiFusion is the first use case that has been realized with the Databus and is described here in the context of the Databus. The platform provides two tools to connect consumers and producers: 1. *for consumers*, the website `https://databus.dbpedia.org` and the SPARQL API `https://databus.dbpedia.org/repo/sparql` serve as a user interface to configure data set retrieval and combination in catalogues, 2. *for providers*, the Databus Maven plugin[5] enables systematic upload and release of datasets on the bus.

2.1 FlexiFusion Workflow on the Databus

Data management tasks such as ETL, integration, fusion and quality assurance are hard and repetitive. In the course of developing the new DBpedia strategy "Global and Unified Access to Knowledge Graphs", we have intensively studied and discussed the (Linked Open) data network for the past two years

[3] `http://ca.dbpedia.org`

[4] `https://databus.dbpedia.org/`

[5] `http://dev.dbpedia.org/Databus_Maven_Plugin`

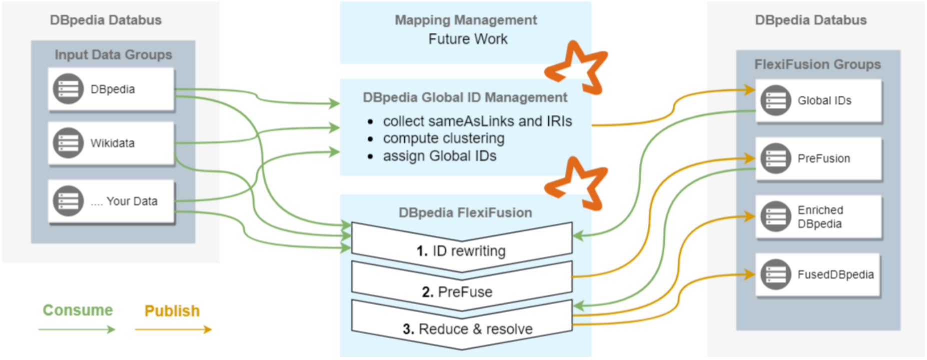

Fig. 1. FlexiFusion on the Databus.

and analysed the struggle of stakeholders to collaborate, hindered by technical and organizational barriers. The efforts for the creation and maintenance of mappings and linksets, error detection & correction, to name just a few, are repeated in individual and use case specific data management processes applied both in research, public bodies and corporate environments. With the DBpedia Databus we envision a hub, where users can register various data artifacts of their data management tasks. In that hub, useful operations like versioning, cleaning, transformation, mapping, linking, merging, can be applied and coordinated on a central communication system - the bus - and then again dispersed in a decentralized network to consumers and applications. On the Databus, data flows from data producers through the platform to consumers while errors or feedback can flow in the opposite direction and propagate to the data source to allow a continuous integration and quality improvement.

Figure 1 shows the FlexiFusion workflow, which is an application of medium complexity built on top of the Databus. Data is likewise consumed (green arrows) and published (orange arrows). The image shows a simplified view, describing FlexiFusion as a pipeline, but in fact it is a distributed network model of individual components, which might be better expressed via formalisms such as Petri Nets[6] that enable analysis of circular dependencies and critical paths. An additional layer of complexity is hidden in the data sources and the sinks on the right and left, as these are in fact data artifacts with versioned snapshots. In the future, any component of FlexiFusion can publish additional feedback information to improve e.g. the ID and Mapping Management based on available options found in the fusion process.

2.2 Modular DBpedia Releases on the Databus

The main motivation to develop the Databus was to switch from *one* very complex, highly interdependent, work-intensive release workflow of DBpedia to sev-

[6] https://en.wikipedia.org/wiki/Petri_net

```
@prefix : <https://downloads.dbpedia.org/repo/lts/mappings/instance-types/2018.12.01/dataid.ttl#> .
@prefix dataid-cv: <http://dataid.dbpedia.org/ns/cv#> . # namespace for content-variants

:Dataset
   a                    dataid:Dataset ;
   dct:title            "DBpedia Ontology instance types"@en ;
   dataid:account        <https://databus.dbpedia.org/dbpedia> ;
   dataid:group          <https://databus.dbpedia.org/dbpedia/mappings> ;
   dataid:artifact       <https://databus.dbpedia.org/dbpedia/mappings/instance-types> ;
   dataid:version        <https://databus.dbpedia.org/dbpedia/mappings/instance-types/2018.12.01> ;
   dct:publisher        <https://webid.dbpedia.org/webid.ttl#this> ;
   dct:license          <http://purl.oclc.org/NET/rdflicense/cc-by3.0> .

:instance-types_transitive_lang=en.ttl.bz2
   a                        dataid:SingleFile ;
   dct:isDistributionOf     :Dataset ;
   dct:title                "DBpedia Ontology instance types"@en ;
   dct:hasVersion           "2018.12.01" ;
   # language and other variants are encoded here
   dataid:contentVariant    "en" , "transitive" ;
   dataid-cv:lang           "en" ;
   dataid-cv:tag            "transitive" ;
   dcat:downloadURL         :instance-types_transitive_lang=en.ttl.bz2 ;
   dcat:mediaType           dataid-mt:ApplicationNTriples .
```

Listing 1: DBpedia DataID snippet of `https://databus.dbpedia.org/dbpedia/mappings/instance-types/2018.12.01`

eral agile, frequent and automated modular releases [4] with short cycles which allows a faster delivery of community contributions (mappings, interlinks, extraction framework fixes and Wikipedia/Wikidata updates) to end users.

Inspired by Maven, datasets are described by `publisher` / `group` / `artifact` / `version`. *Groups* provide a coarse modularization. From a top level view, DBpedia is now separated into 5 different groups, which are produced by separate extraction processes with separated dependencies: *generic* (automatically extracted information from raw infoboxes and other sources), *mappings* (mapping-aided infobox extraction), *text* (article abstracts and textual content), and *wikidata* (Wikidata facts mapped to DBpedia ontology [5]) and the *ontology*. *Artifacts* are the abstract identity of the dataset with a *stable dataset id*, e.g. there is a *geo-coordinates* artifact in generic, mappings and wikidata. Each artifact has *versions*, that usually contain the same set of files for each release. Files within a version are additionally described by content variants (e.g. `lang=en`), mediatype and compression. The overall structure is very flexible as software libraries, but also – once defined – as fixed as software to prevent applications from breaking, if they update on a new dataset version [4]. Further details are described in the user manual[7].

2.3 Data Selection and Retrieval

Once artifacts are established, new versions can be published automatically and the metadata of the published data is machine-comprehensible via the

[7] `http://dev.dbpedia.org/Databus_Upload_User_Manual`

```
PREFIX dataid: <http://dataid.dbpedia.org/ns/core#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
SELECT distinct ?file {
  ?dataid dataid:version ?latest;
          dcat:distribution ?distribution .
  ?distribution dcat:downloadURL ?file;
                dataid:contentVariant "transitive"^^xsd:string .
  { SELECT DISTINCT ( MAX( ?version ) as ?latest ) {
    ?s a dataid:Dataset ;
       dataid:artifact ?artifact;
       dataid:version  ?version .
       FILTER ( ?artifact in (
          <https://databus.dbpedia.org/dbpedia/mappings/instance-types>,
          <https://databus.dbpedia.org/dbpedia/wikidata/instance-types>
        ))
    } GROUP BY ?artifact
}}
```

Listing 2: Example SPARQL query for input dataset selection fetching the download URLs for the latest version of transitive type information from DBpedia and Wikidata instance types artifacts.

DataID/DCAT vocabulary (an example can be seen in Listing 1). The Databus Maven Plugin uses the Maven Lifecycle phases to generate this metadata based on a configuration provided by the publisher via ‘mvn databus:metadata‘ and uploads it to the Databus via ‘mvn deploy‘ at the final stage of the publishing process to the Databus SPARQL endpoint. This endpoint can be queried in order to fetch a custom tailored selection of groups/artifacts/files in specific versions. As the data itself is hosted in the publisher’s webspace, queries retrieve metadata in form of `dcat:downloadURLs` for the files.

FlexiFusion is fed by a fine-grained selection of RDF data files (`?files`) via SPARQL queries (see Listing 2) using stable identifiers of the form `https://databus.dbpedia.org/<publisher>/<group>/<artifact>`. The SPARQL queries are considered as configuration of input data dependencies and can be used to fetch the most recent versions of the dependencies.

3 DBpedia PreFusion Dataset

The *DBpedia* ***PreFusion*** dataset is a new addition to the modular DBpedia releases combining DBpedia data from over 140 Wikipedia language editions and Wikidata. As an intermediate step in the FlexiFusion workflow, a global and unified preFused view is provided on a core selection of DBpedia dumps extracted by the DBpedia extraction framework [7]. The facts are harvested as RDF triples and aggregated using a new serialization format to track statement-level provenance. Unified access to knowledge from different sources is achieved by exploiting previously existing mappings of the DBpedia Ontology as well as merged,

```
{ "@id": "fc4ebb0fed3c3171578c299b3ce21f411202ff2afc93568a54b4db7a75",
  "subject": { "@id": "https://global.dbpedia.org/id/12HpzV" },
  "predicate": { "@id": "http://dbpedia.org/ontology/floorCount" },
  "objects": [ {
    "object": {
      "@value": "4",
      "@type": "http://www.w3.org/2001/XMLSchema#positiveInteger" },
    "source": [ {
      "@id": "d0:lang=fr.ttl.bz2",
      "iHash": "cbdcb" } ]
    }, {
    "object": {
      "@value": "3",
      "@type": "http://www.w3.org/2001/XMLSchema#positiveInteger" },
    "source": [ {
      "@id": "d0:lang=en.ttl.bz2",
      "iHash": "1e7d4"
      }, {
      "@id": "d0:lang=es.ttl.bz2",
      "iHash": "eb41e" } ] } ],
  "@context": "sources=dbpw_context.jsonld" }
```

Listing 3: Example PreFusion JSON(-LD) Object for sp-pair *Eiffel tower* and `dbo:floorCount`. The French Wikipedia version reports 3 floors (above ground) in contrast to 4 in English and Spanish.

normalized entity identifiers (DBpedia Global IDs). The ontology defines a comprehensive class hierarchy and properties, which are modelling common entities described in Wikipedia and Wikidata, and also reuses prominent vocabularies like FOAF and PROV. The dataset offers knowledge about very broad domains (like persons and organizations) but also for very specific domains (e.g. nutrition facts or animal classifications).

The dataset is published under an open CC-BY license on the DBpedia Databus[8] and there is an experimental web service[9] which allows to browse all triples with their provenance for a given entity id (IRI). The DBpedia Association has drafted a roadmap[10] for automating modular releases and also releases of the PreFusion dataset in a sustainable way. Both the browsable interface and the PreFusion dump are preliminary work for the GlobalFactSync project[11] funded by Wikimedia.

PreFusion Format. The PreFusion dataset is stored as JSON-LD using a custom scheme optimized for an efficient representation of entities with overlapping object values and groups multi-value statement-level provenance. Thus, the

[8] `https://databus.dbpedia.org/dbpedia/prefusion`

[9] `https://global.dbpedia.org/`

[10] `https://blog.dbpedia.org/2019/07/04/dbpedia-growth-hack`

[11] `https://meta.wikimedia.org/wiki/Grants:Project/DBpedia/GlobalFactSyncRE`

Table 1. PreFusion dataset factsheet, `dbpedia/prefusion/$artifact/2019.03.01`

artifact	distinct objects	source triples	subjects	sp-pairs	wikipedias	size (bz2)
labels	266,633,208	297,345,045	91,146,077	91,146,077	139+wd	7.2G
instance-types	191,702,603	293,261,187	25,230,546	25,230,546	40+wd	2.1G
mappingbased-objects	150,955,259	263,677,844	45,063,398	98,388,770	40+wd	6.1G
mappingbased-literals	94,111,662	100,049,794	36,500,856	71,427,960	40+wd	4.0G
geo-coordinates	41,313,484	51,178,574	8,517,009	34,099,723	140+wd	1.8G
specific-mappingbased	2,198,020	2,548,485	1,083,961	1,568,804	40	82M

dataset can be loaded both into JSON document/NoSQL stores, in case simple lookups are required – and triple stores – in case joins are required. Each PreFusion document describes a normalised subject-predicate pair ($sp - pair$) to aggregate all different object and literal values from the input sources as shown in Listing 3. The provenance record(s) are referencing the corresponding input file(s) of the object value and iHash value(s) which can be used to determine the original (non-normalized) IRI(s) of the triple(s) by hashing the result of the Global ID Resolution service[12].

PreFusion dataset statistics. The dataset is structured in 6 artifacts[13] shown in Table 1 with similar names to the original structure of the DBpedia and Wikidata extraction dumps. The dataset contains a billion triples and more than 321 million subject-predicate pairs. Mappings are only maintained for 40 Wikipedia languages which explains the lower number of entities for this artifact. We picked 5 essential artifacts with overlapping but also complementary data in the input sources and the `labels` artifact. The latter contains more than 266 million `rdfs:labels` for over 91 million entities covering 139 language (variants). The `instance-types` artifact contains `rdf:type` statements using the DBpedia ontology as foundation but also incorporating other ontology classes (e.g. schema.org, Wikidata, FOAF, etc.). The mapping-based artifacts contain factual knowledge about entities extracted from Wikipedia infoboxes using mappings maintained by the community[14]. The `geo-coordinates` artifact adds a spatial dimension by offering coordinates which have been mapped from the infoboxes but also points which are related to an entity since they have been spotted in the Wikipedia article.

4 FlexiFusion Workflow

4.1 PreFuse: Normalize

ID Management. The Web of Data uses a decentralized approach with `owl:sameAs` relations to interlink different RDF Resources which represent the same thing. However, a lot of effort is required to obtain a global view of this decentralized knowledge in order to perform a holistic data integration. We developed

[12] `http://dev.dbpedia.org/Global_IRI_Resolution_Service`
[13] version: `https://databus.dbpedia.org/dbpedia/prefusion/$artifact/2019.03.01`
[14] `http://mappings.dbpedia.org/index.php/Main_Page`

the DBpedia Global ID Management[15] to create a central curation hub. In a nutshell, it materializes the global view of links formed by several linksets and datasets available in the Web of Data, computes SameAs clusters by deriving connected components, and selects a DBpedia Global ID as a representative for every cluster, which can be used as uniform identifier for all of its equivalent identifiers. Moreover, the ID Management assigns stable Global identifiers for IRIs from a configurable list of data authorities. Current snapshots of the ID Management are accessible as dump or in a resolution service. The ID Management works independent of any link discovery tool. Linking results from any approach can be injected if they are represented as owl:sameAs links.

Mappings. While the ID Management normalizes IRI's of subjects and objects, the normalization of literals and predicates needs to be handled by mappings. For the case of the Wikipedia and Wikidata extraction, the DBpedia ontology is used as global schema. Units (e.g. feet vs. meters) between DBpedia chapters are already normalized and standard RDF datatypes as well as DBpedia Ontology datatypes are used to represent literal values in a normalized form. In order to include other datasets, ontology mappings and value transformations need to be provided. While not developed yet, we can imagine a Mapping Management component which works similar to the ID Management, i.e. connected components over `owl:equivalent(Property|Class)`. In the current workflow, we assume that existing mapping tools are used to provide triple files using normalized predicates and literal values.

4.2 PreFuse: Aggregate

The **PreFuse** operation is fed with the individually tailored combination of *normalized* triple files from Databus artifacts. Every input triple from this collection is extended by a provenance record and then streamed into a sorted queue. A preFused entity e is created by grouping all triples first by same subject and then by their predicate value. We can represent the result of this grouping as a set of predicates for e whereas for each predicate a list of pairs of the form (object, provenance) is stored and then embedded as JSON(-LD) object for its subject-predicate *sp*-pair. The output of the PreFusion operation is the PreFusion dump with indexed preFused sp-pairs – a global and unified view of its input data. Since this view is persisted on the Databus, it can be used as input for a series of different data fusion approaches without expensive re-computation of the global view. While it can be used for analytical queries as is, we think of it as a predigested version of the input data which can be used to derive custom (fused) datasets from it.

4.3 Fuse: Reduce and Resolve

The PreFusion is followed by two consecutive operations: *reduce* and *resolve*. Different combinations of reduce and resolve realisations can be applied on the PreFusion dump to produce various custom-tailored fused datasets.

[15] `http://dev.dbpedia.org/ID_and_Clustering`

The **reduce** operation is a function applied on the subject-predicate pairs to reduce or filter the amount of entities and the amount of information for each entity. Reduce acts as a coarse-grained blocking key that removes irrelevant fusion decisions in resolve. Reduction is based on type, source, predicate or entity, e.g. reducing to `dbo:birtplace` for the five largest language sources or just the 1 millon entities from the Catalan source. Reduce is scalable and can even select just a single subject-predicate pair as in the online service.

The purpose of the **resolve** function is to pick a number of objects from the list of each subject-predicate pair in the reduction and resolve conflicts to improve data quality. Let's consider an example person having multiple contradicting `dbo:birthdate`s. The realisation of a resolve function could e.g. be defined to select only one object value if the predicate is a functional property. The candidate to pick could be chosen by heuristics like source-preference, majority-voting, random choice, etc.

In the current resolve prototype (based on Apache Spark), we implemented 2 conflict resolution strategies: a configurable preference list which specifies the priority of input sources and a majority voting approach. The former picks the value from the source which has the highest preference while the latter picks the option which has the highest number of occurrences.

These strategies are further augmented by a cardinality component that limits the number of selected values. Whenever the resolve function resolves an sp-pair with a p declared functional or max-cardinality=1 in the ontology, it will pick only one value based on the decision of the above conflict resolution strategy. As schema declarations are often missing and to account for misbehaving data, a second approach uses *predicate median out degree (PMOD)* as a heuristic, which is calculated based on the PreFusion Dump for every property. $PMOD(p)$ is defined as the median of the list of all cardinality/out degree values for predicate p from entities with at least one object for p. It triggers, if the $PMOD(p)$ equals one.

5 DBpedia Chapter Use Case

DBpedia is organized in so called chapters. A chapter concentrates and coordinates effort and resources to host and maintain one specific language version of DBpedia. This includes forming a local community to maintain mappings, hosting a SPARQL endpoint and many more. The data quality of the dataset of a specific chapter is influenced by 3 major factors: the richness and freshness of the used Wikipedia version, the language-specific infobox mappings, localization configurations and adaptions in the extraction framework. More importantly, chapter data often contains richer and more accurate information about entities of regional importance which are not covered at all in other prominent chapters or Wikidata. Moreover, every chapter offers some complementary data (e.g. labels of entities in the local language). However, complementing each other via collaboration on the data level has the potential to increase coverage and quality and lessen maintenance workload, thus benefiting primarily chapters, but also

Table 2. Overall coverage and knowledge gain of fusion.

	Wikidata	English	German	French	Dutch	Swedish	**Fusion**
triples	436,808,402	124,994,586	42,630,107	39,438,426	36,924,058	37,942,711	558,597,215
sp-pairs	179,789,022	77,368,237	26,086,747	26,049,036	24,339,480	29,062,921	465,018,956
entities	45,649,373	17,576,432	5,020,972	5,429,710	3,638,110	5,862,430	66,822,365
dist. properties	166	1,412	598	1,052	979	415	2,292
avg. dist. predicates per entity	3.938	4.402	5.196	4.798	6.690	4.957	6.959

Table 3. Typed entity distribution of the four types person, company, location, organization. Each second line counts the entities that exist in at least one other source, but are only typed in this source. Percentage gain is relative to richest source.

Class	Wikidata	English	German	French	Dutch	Swedish	**Fusion**
dbo:Person	4,197,564	1.757,100	627,353	491,304	188,025	62,814	4,612,463 (+9,88%)
only typed in source	2,246,879	350,137	26,896	6,498	4,506	316	
dbo:Company	188,107	70,208	25,208	14,889	4,446	3,291	209,433 (+11,34%)
only typed in source	80,443	4,038	834	548	89	121	
dbo:Location	3,952,788	839,987	406,979	276,096	449,750	1,480,627	5,293,969 (+33,93%)
only typed in source	2,451,306	27,430	25,804	14,979	101,422	33,425	
dbo:Animal	8,307	228,319	145	0	675,337	437	784,808 (+16,21%)
only typed in source	2,963	2,302	1	0	2,029	5	

the main DBpedia as well as Wikidata and Wikipedia. In the scope of the paper we created two scenarios: a *FusedDBpedia* prototype comprising information of several chapters and an enrichment of the Catalan DBpedia.

For *FusedDBpedia* we reduced to 6 sources, i.e. Wikidata, the English (EN), German (DE), French (FR), Dutch (NL) and Swedish (SV) chapter and resolved via: select 1 object value based on language preference (Wikidata, EN, DE, FR, NL, SV) iff $PMOD(p) = 1$; else take all values. For *EnrichedCatalan* we reduced to *sp*-pairs where s is a subject from the Catalan DBpedia data and resolved via: select all values iff $PMOD > 1$ else Catalan value has preference, if exists, otherwise use preference list of FusedDBpedia. For *FusedDBpedia*, as the earlier evaluation scenario, we used an older version of the PreFusion dataset comprised of DBpedia releases from October 2016, whereas for the *EnrichedCatalan* new releases were available and we used the version from March 2019 presented in Section 3. We used the ID Management snapshot from February 2019 which is based on Wikidata Interwiki-Links.

6 Evaluation

6.1 FusedDBpedia Dataset Evaluation

Data Coverage. Table 2 gives an overview on data coverage of the *FusedDBpedia* dataset compared to the 6 source datasets. The fused dataset gained more than 120 million triples and almost 300 million subject-predicate pairs. Entity coverage is improved by 47% with respect to the entity-richest source (Wikidata-DBpedia). Further, the fused data offers on average seven distinct

Table 4. Property coverage, gains, and distribution for two high frequent properties.

Property	Wikidata	English	German	French	Dutch	Swedish	**Fusion**
triples with dbo:birthDate	3,044,381	1,740,614	639,851	623,055	246,102	606	3,096,767
distinct entities	3,031,415	1,216,106	639,281	449,742	175,587	606	3,096,767
only in source	1,376,942	25,272	33,540	4,852	1,330	7	+2,16%
triples with dbo:scientificName	0	0	241,998	0	890,644	1,329,536	1,691,734
distinct entities	0	0	43,974	0	890,567	1,329,535	1,691,734
only in source	0	0	7,171	0	351,990	780,555	+27,24%

Table 5. Overall failed RDFUnit test case comparison between source and result data.

	Wikidata	English	German	French	Dutch	Swedish	Fusion
applicable tests (prevalence>0)	531	5,002	1,992	3,560	3,332	1,486	8,060
overall failed tests	325	1,055	418	722	647	432	1,755
overall success rate	38.79%	78.91%	79.02%	79.72%	80.58%	70.93%	78.23%
smaller fail rate in source	86	288	163	221	285	115	-
equal fail rate in source	5	84	8	74	32	8	-
greater fail rate in source	214	643	229	406	306	297	-
not failed in fused data	20	40	18	21	24	12	-
tendency of data quality improvement	yes	yes	yes	yes	yes	yes	-

properties per entity compared to around five averaged over all sources with an increased vocabulary usage of 62%.

Table 3 shows the distribution of four high frequent entity types. Note that Wikidata refers to the Wikidata-DBpedia extraction [5], which uses several effective methods to discover and clean proper type statements from Wikidata. The fusion achieved an entity-type gain from ≈10-33% for these types. Furthermore, we observed that one or two datasets significantly contribute to the entity gain, but they vary depending upon the class. Nevertheless, we can see that every (besides French chapter for Animals) dataset contributes knowledge, which is especially indicated by the "only typed in source" values. The value shows how many type statements are uniquely provided by one source and can directly enrich at least one other source, e.g. 2,000 *dbo:Animal* types from Dutch to other sources.

In Table 4, the data coverage and knowledge gain for two frequent properties is shown. We observed violations of the cardinality constraint for the functional *dbo:birthDate* property in every source dataset. This issue is solved in the fused data based on the correct decision of the PMOD-based resolution function to pick only one value. With regard to *dbo:scientificName* the Swedish and Dutch datasets provide a high knowledge gain. In accordance with the findings from Table 3 and 4 this supports the hypothesis that smaller and therefore less developed and/or chapters with fewer infobox mappings can still contribute valuable information for specific domains. It substantiates the basic idea of FlexiFusion to include all information in the PreFusion dataset and postpone the decision which data to filter and which values to select to the specific use case.

Data Quality. Evaluating the correctness of a dataset is a challenging task. To the best of our knowledge no gold standard exists which could be used to automatically evaluate the entire dataset. Therefore, we decided to use RDFUnit

Table 6. Enriched Catalan Statistics

	Original	Enriched	Boost		Original	Enriched	Boost
overall triples	4,631,162	31,200,104	6.74	edge to non-Ca IRI	248,685	5,725,446	23.02
distinct entities	981,795	981,795	1.00	edge to Global IDs	-	858,551	-
properties distinct	111	2,275	20.50	Global ID targets	-	254,515	-
sp-pairs	200,094	4,125,355	20.62	ext. non-Ca targets	22,464	2,210,614	98.41
avg pred. outdegree	0.20	4.20	20.62	ext. non-DBp targets	22,464	1,358,754	60.49
avg indegree	0.23	2.58	11.20	ext. DBpedia targets	0	597,045	-

[6] as a meter for data quality. RDFUnit performs an automatic generation of test instances based on schema knowledge (ontology) and the used vocabulary. Additionally it contains several manually defined plausibility tests. It supports reporting for the number of failed test instances and prevalence values (how often could the test be applied). The evaluation is based on a simple assumption: a lower number of failures in the data implicates better data quality. Since the number of triples and entities significantly varies between the datasets, we defined the *fail rate* as quality metric. It is determined by normalizing the number of failed test instances by the prevalence value for the test case. In total 12,250 generated and 14 manual tests were used, resulting in 1,833 distinct failed test cases. For the sake of brevity, we have summarized the RDF unit reports in Table 5.

To summarize the data quality analysis, the test reports are compared by overall failed tests and also by the fail rate for each test. We classify each test result for the sources based on its fail rate in comparison to the fused fail rate into four categories: *smaller* (fail rate smaller in source than in fusion), *equal*, *greater* and *not failed* (fail rate in fusion = 0) compared to the fusion results. *Smaller* is an indicator that data quality decreased throughout the fusion, while all remaining classes are interpreted as improvement.

About 65,8% of the generated test cases returned a prevalence value greater zero on the fused data. This is the highest prevalence number compared to all sources which in turn reflects data coverage improvements. It was not surprising that the number of failed test cases is the highest, too. However, we did not expect that the success rate would be better than average and close to the best value. The rates on top of Table 5 do not take the number of errors per test (i.e. how badly a test failed) into account. In contrast, the bottom-part classification-based comparison pays tribute to the fail rate of every individual test. Based on this, the fused data shows a tendency of quality improvement compared to each individual source dataset.

6.2 EnrichedCatalan Dataset Evaluation

We defined two binary criteria to study which kind of data is available in the PreFusion dataset for the Catalan enrichment scenario. The *Sole Source Criterion* (SSC) is true for a source s given an sp-pair p iff all values from s in p are only originated in s. The *Alternative Choices Available Criterion* (ACC) is true iff at least one different value from a source other than s is available in p. The

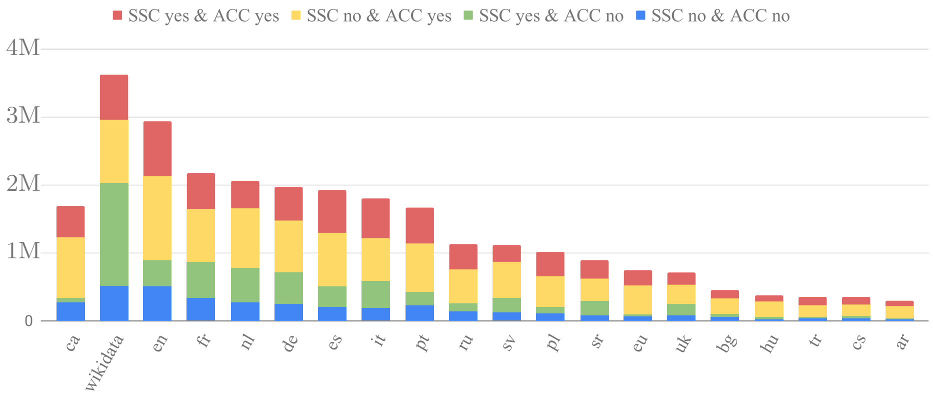

Fig. 2. Information classification in PreFusion reduced for Catalan entities.

distribution is shown in Figure 2 for 20 sources which contribute the highest number of sp-pairs for Catalan. The combination of SSC and ACC allows to distinguish 4 different categories. The sources are unanimous for `no/no`(blue) and agree on these values, interpretable as more accurate or consensual information. New unique information is contributed by s in case of `yes/no`(light green) and if selected makes the resulting dataset more rich, albeit with unclear accuracy, i.e. just more data. Both `no/yes`(yellow) and `yes/yes`(red) have mixed value in need of more elaborate inspection and resolution, whereas `yes/yes`(red) is more polarized and can be interpreted as either complementary beneficial or an erroneous outlier.

Moreover, we present a few statistics on how the PreFusion dataset was used to boost the Catalan chapter in Table 6. The first part of the table shows the overall boost. In the second part we focus on edges between objects only and show an improvement of both intralinking (indegree) of Catalan entities by factor 11 but also interlinking to resources of external datasets by almost factor 100.

7 Related Work

ID Management. An important step in the data integration pipeline is identifying entities that refer to the same real-world thing. In the Semantic Web this is known as *Link Discovery* or *Entity Clustering*, where the latter usually describes the interlinking of entities from multiple sources. A significant amount of research has already been dedicated to this field and an overview can be found in [10]. Dedicated clustering strategies for integrating multiple sources have been developed by Nentwig et al. [9] utlizing existing `owl:SameAs` links to build initial clusters. Saeedi et al. [12] compared different clustering schemes with respect to the suitability and scalability for the entity clustering task. In [11] a scalable approach is presented to integrate new data into existing clusters. To avoid comparing new entities with all members of existing clusters, each cluster creates a cluster representative, that is fused from all the properties of the cluster members. The DBpedia Global ID Management that is used in this approach can be seen as a conservative clustering technique, that makes implicit `owl:SameAs` links that exist in the Web of Data explicit and assigns a global cluster ID.

Fusion architectures. HumMer [1] is a framework for fusing heterogeneous relational data in three steps: schema mapping (based on (DUMAS) [2]), duplicate detection, and conflict resolution. In addition to the DUMAS algorithm, pairwise similarity measurements are used to detect duplicated entities which are then extended by a uniform *objectID*. The conflict resolution is based on user defined aggregation functions in SQL (e.g. *choose source*, *first or last*, *vote*, *group*, *concatenate*, *most recent value*).

Sieve [8] is a project that aims to fuse Linked Data based on data quality assessments. It is implemented in the JAVA-based Linked Data Integration Framework (LDIF) [13] offering modules for data access, schema mappings, identity resolution, and data output management. Sieve uses scoring functions to rank various content- but also context-based (e.g. release date of the triple) quality indicators to calculate quality metrics. The fusion process relies on a configuration defining one fusion function for each property of a class. A fusion function is able to use the defined quality metrics to select the best value(s).

In comparison to our approach, Sieve - albeit more fine-grained and selective – requires a higher complexity and more configuration effort for every ontology used in the source datasets to tailor the fusion. With respect to the DBpedia ontology this configuration would not be pragmatic due to the large number of different classes and properties.

8 Conclusion

The presented FlexiFusion approach creates the PreFusion dataset as part of future canonical DBpedia releases, which is able to hold all information from the input sources plus additional provenance links on dataset-level and entity-level and enables the comparison of values accross datasets.

Based on this PreFusion dump, we have tailored two use-case specific datasets, a fused DBpedia and an enriched version of the Catalan DBpedia based on a datatype-agnostic resolve function, which consists of the computed predicate median out degree and the implementation of the chosen preference or majority value selection.

The first part of the evaluation has shown that the FusedDBpedia has larger coverage, while still containing a higher information density and is overall more consistent regarding RDFUnit test (interpreted as quality improvement). The second part shows that we boost the Catalan source by a factor 7 in size and 10-100 fold in other metrics. The two criteria Sole Source (SSC) and Alternative Choice (ACC) give a high-level insight over all sources about which data is in sync, which data is uniquely gained by new sources and where to expect the most conflicts and quality issues during fusion, thus easing the decision on what to integrate.

9 Discussion and Future Work

As a critical judgement of the DBpedia FlexiFusion approach, we have to admit that while the approach as a workflow is quite advanced and has been evaluated for the Chapter Use Case, the research on best practices of how to configure and evaluate FlexiFusion is still in its early phase. Nevertheless, we decided to publish the dataset resource in its current configuration (140 Dbpedia language editions plus Wikidata) as we already see great benefits for further research and applications by the community. Our next steps, will be the automated publication of enriched DBpedia language versions and delivery to Chapters as well as the loading of an enriched English version into the main DBpedia endpoint[16]. For now, we provided one evaluation of Wikidata + 5 large DBpedia language editions, which enabled us to draw the above-described conclusions (cf. Section 8), which show the successful application of the Chapter Use Case. However, our evaluation has the following limitations, which create ample opportunities for further research:

- While our work eases the workload to deploy and evaluate fusion approaches (e.g as mentioned [3]), we only implemented three simple methods for the resolve function (median-based, majority, preference), leaving the field wide open for other researchers to experiment with more sophisticated measures.
- We used basic metrics and SHACL shapes in our evaluation. During our development of FlexiFusion, we also saw potential to adjust the fusion algorithm to directly employ the SHACL shapes for selection and resolution, i.e. choosing the option that produces fewest constraint violation. Using SHACL for fusion selection and evaluation at the same time, however, is a weak methodology.
- In our evaluation, we used uniform, rule-based classifiers such as majority or preference, which we expect to be outperformed by deep learning approaches that have shown to produce better fitting results. The main limitation here is the lack of training and test data. The only solace our approach can offer here is that in case a gold standard exists, we can load it alongside the other data into the FlexiFusion format to ease the implementation of further evaluation. Another potential approach is to link, map and load professionally curated data e.g. by libraries to serve as a silver standard.

Moreover we used a simple, but effective method to bootstrap the ID Management to solve the chicken and egg problem: it is hard to automatically derive mappings between two sources without any links for a first clustering, but also hard to compute entity similarities for link discovery without partial mappings. We can imagine to extend the FlexiFusion workflow with feedback loops from the fusion step to the linking and mapping steps. If the fused entity of the cluster has a significant lower similarity to one of its members this is an indicator for an incorrect link. The link in question could be deleted or marked as low confidence link in order to improve the fusion in the next iteration. Similar strategies could be applied to detect mapping errors. Using an automatic quality driven

[16] `http://dbpedia.org/sparql`

approach linksets and mappings could be refined on every iteration based on the quality reports during the fusion (e.g. high conflict rate for one property).

We also see the potential for a Mapping Management based on analogous concepts to the ID Management. Fed by various (binary) mappings it could form a global mapping view to be able to derive mapping rules to translate classes/properties of a dataset using ontology A into the ones of ontology B, potentially also without the need for a direct mapping from A to B. This could be another step into the direction to reuse mappings, establish synergies and share efforts to cut down engineering costs to create, improve and maintain mappings in a collaborative way.

Acknowledgements. We thank Prof. Rahm for his valuable input during the holistic data integration discussions. We thank Jens Grivolla for providing the Catalan use case and all DBpedia chapters and the community. The work is in preparation to the start of the WMF-funded GlobalFactSync project[17].

References

1. Bilke, A., Bleiholder, J., Naumann, F., Böhm, C., Draba, K., Weis, M.: Automatic data fusion with hummer. In: Proceedings of the 31st international conference on Very large data bases. pp. 1251–1254. VLDB Endowment (2005)
2. Bilke, A., Naumann, F.: Schema matching using duplicates. In: Data Engineering, ICDE. pp. 69–80. IEEE (2005)
3. Bleiholder, J., Naumann, F.: Conflict handling strategies in an integrated information system. In: IJCAI Workshop on Information on the Web (IIWeb) (2006)
4. Feeny, K., Davies, J., Welch, J., Hellmann, S., Dirschl, C., Koller, A.: Engineering Agile Big-Data Systems, vol. 1. River Publishers (10 2018)
5. Ismayilov, A., Kontokostas, D., Auer, S., Lehmann, J., Hellmann, S.: Wikidata through the eyes of dbpedia. Semantic Web 9(4), 493–503 (2018)
6. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A.: Test-driven evaluation of linked data quality. In: WWW. pp. 747–758 (2014), http://svn.aksw.org/papers/2014/WWW_Databugger/public.pdf
7. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. SWJ 6(2), 167–195 (2015)
8. Mendes, P.N., Mühleisen, H., Bizer, C.: Sieve: Linked data quality assessment and fusion. In: EDBT/ICDT. pp. 116–123. ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2320765.2320803
9. Nentwig, M., Groß, A., Rahm, E.: Holistic entity clustering for linked data. In: IEEE, ICDMW. IEEE Computer Society (2016)
10. Nentwig, M., Hartung, M., Ngomo, A.C.N., Rahm, E.: A survey of current link discovery frameworks. Semantic Web 8, 419–436 (2017)
11. Nentwig, M., Rahm, E.: Incremental clustering on linked data. 2018 IEEE ICDMW pp. 531–538 (2018)
12. Saeedi, A., Peukert, E., Rahm, E.: Comparative evaluation of distributed clustering schemes for multi-source entity resolution. In: ADBIS (2017)
13. Schultz, A., Matteini, A., Isele, R., Bizer, C., Becker, C.: Ldif-linked data integration framework. In: COLD, Volume 782. pp. 125–130. CEUR-WS. org (2011)

[17] https://meta.wikimedia.org/wiki/Grants:Project/DBpedia/GlobalFactSyncRE