

LODE: Linking Digital Humanities Content to the Web of Data

Jakob Huber
Data and Web Science Group
University of Mannheim

Timo Szytler
Data and Web Science Group
University of Mannheim

Jan Noessner
Data and Web Science Group
University of Mannheim

Jaimie Murdock
School of Informatics and
Computing & Program in
Cognitive Science
Indiana University

Colin Allen
Department of History and
Philosophy of Science &
Program in Cognitive Science
Indiana University

Mathias Niepert
Department of Computer
Science and Engineering
University of Washington

ABSTRACT

Numerous digital humanities projects maintain their data collections in the form of text, images, and metadata. While data may be stored in many formats, from plain text to XML to relational databases, the use of the resource description framework (RDF) as a standardized representation has gained considerable traction during the last five years. Almost every digital humanities meeting has at least one session concerned with the topic of digital humanities, RDF, and linked data.

While most existing work in linked data has focused on improving algorithms for entity matching, the aim of the LINKEDHUMANITIES project is to build digital humanities tools that work “out of the box,” enabling their use by humanities scholars, computer scientists, librarians, and information scientists alike.

With this paper, we report on the Linked Open Data Enhancer (LODE) framework developed as part of the LINKEDHUMANITIES project. With LODE we support non-technical users to enrich a local RDF repository with high-quality data from the Linked Open Data cloud. LODE links and enhances the local RDF repository without compromising the quality of the data. In particular, LODE supports the user in the enhancement and linking process by providing intuitive user-interfaces and by suggesting high-quality linking candidates using tailored matching algorithms. We hope that the LODE framework will be useful to digital humanities scholars complementing other digital humanities tools.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.3.7 [Information Storage and Retrieval]: Digital Libraries—*Systems issues*

General Terms

Algorithms, Human Factors, Design

1. INTRODUCTION

A preeminent scholarly problem is how to comprehend the explosion of high-quality scholarship available in digital formats on the Internet. Humanities scholars, like all academics, are increasingly reliant on the World Wide Web for access to scholarly materials and they are rapidly transferring traditional journals and rare archives to digital formats, further exacerbating the problems of information overload.

Every year, digital humanities projects present their work at the International Conference for Digital Scholarship in the Humanities¹ (DH) and the number of collections is growing steadily. The recent introduction of the new track “digital humanities” at the Joint Conference on Digital Libraries (JCDL) underlines the importance of this research field.

Available search engines have failed to solve the problem of meaningful access, and most users, including students and scholars, lack the necessary skills to construct effective search queries. (For an overview of issues relating to the novelty of search querying, see [51].)

In light of these challenges, some digital humanities projects have begun to build and maintain collections using machine-readable and structured representations such as XML and RDF. In recent years, the Linked Data initiative² has gained considerable traction. Its goals are to create large and interconnected collections of open and structured data repositories. Arguably, the most prominent examples is DBPEDIA [9]³ – a data repository that contains structured information extracted from Wikipedia. In the last few years,

¹ <http://dh2013.unl.edu/>

² <http://linkeddata.org>

³ <http://dbpedia.org/About>

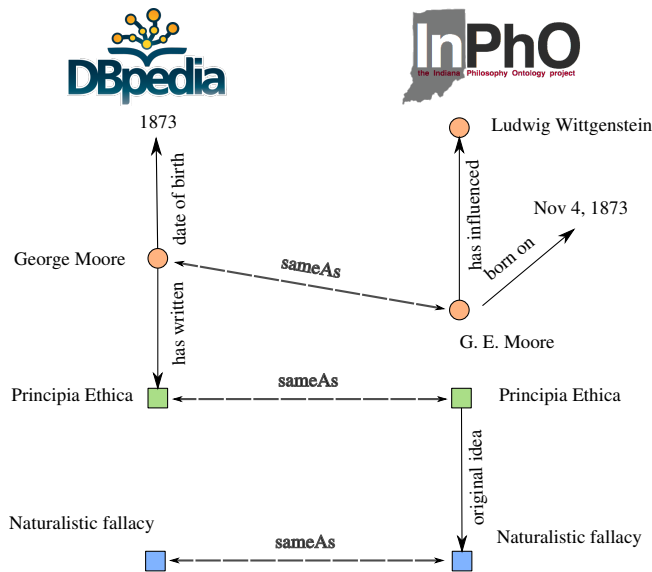


Figure 1: Graphical representation of a fragment of RDF data. The most common semantic links between repositories (here DBpedia and InPhO) are *sameAs* links that assert equality between entities.

Linked Data has become increasingly important in the area of digital humanities. There have been an overabundance of projects like JEROMEDL [29], TALIA [47] and, more recently, the Digitized Manuscripts to Europeana (DM2E)⁴ project, whose aim is to enable humanities researchers to work with manuscripts in the Linked Open Web.

The primary motivation for the LINKEDHUMANITIES project is that data and knowledge in isolation does not leverage its full potential. By including links between entities within a data collection and to external resources, novel information is created and inferred, making the resulting collection more valuable than the sum of its parts and giving information context and interoperability.

With this paper, we present the linked open data enhancer (LODE) framework, which is the main result of the bi-lateral LINKEDHUMANITIES project with the goal to create and maintain data exploration and integration tools tailored to digital humanities collections so as to help build a machine-readable web of humanities data [48, 49, 4]. LODE features (a) an explorer component that allows digital humanists to browse and explore local RDF repositories; (b) a linking components that facilitates the linking of local RDF repositories to external RDF repositories such as DBPEDIA; and (c) an enhancement component for populating and extending the local RDF repository by exploiting the previously created links. In particular, we target use cases that emphasize high quality data requiring human supervision. The linking components provides two different candidate ranking algorithms, which are both suitable for high quality interactive matching candidate selection. We evaluate these linking algorithms

⁴ <http://dm2e.eu/>

with respect to typical digital humanities entities such as documents, concepts, and persons.

LODE’s data integration and enhancement component was designed so as to serve the needs of typical digital humanities projects. In fact, existing projects that maintain RDF collection can benefit from LODE. We have explored and report on concrete use cases such as the Indiana Philosophy Ontology (INPHO)[38, 39, 10]⁵ project and the Stanford Encyclopedia of Philosophy (SEP). Furthermore, we show that typical information extraction projects such as the Never Ending Language Learning (NELL)[11]⁶ project, can be used with LODE. NELL applies machine learning algorithms to continuously extract knowledge from the web and has already accumulated more than 50 million facts in form of object-predicate-subject triples.

The LODE framework is a collection of integrated digital humanities tools working “out of the box” and shielding most of the technological standards and intricacies from its users. The major assumption is that digital humanities projects are run by humanists and librarians not computer scientists. Indeed, most existing work has focused on improving specialized algorithms for entity linking and ontology matching [19]. While the LODE approach does take advantage of such technologies, the main focus is on user-friendly interfaces. It satisfies all of the criteria for digital humanities infrastructure, as outlined by [12], including: 1) named entities (via URIs); 2) a cataloging service (via the RDF relations); 3) structured user contributions (via the linking interface); 4) custom, personalized data (via tools powered by the open-access querying interfaces). The design also follows the guidelines of Stollberg et al.[53], by providing a concrete example of a semantic web portal.

2. LINKED OPEN DATA

The term *linked data* describes an assortment of best practices for publishing, sharing, and connecting structured data and knowledge over the web [8]. These standards include the assignment of URIs to each datum, the use of the HTTP protocol, RDF data model (Resource Description Framework), and hyperlinks to other URIs [6]. Whereas the traditional World Wide Web is a collection of documents and hyperlinks between these documents, the data web extends this to a collection of arbitrary objects (resources) and their properties and relations. For example, rather than containing article content, DBPEDIA represents each WIKIPEDIA article as its own entity, and leverages the link structure between articles as well as structured “infobox” data to establish semantic relations [4].

These relations are modeled using the resource description framework (RDF)[33], a generic graph-based data model for describing objects and their relationships with each other. Further semantic relations like **broader**, **narrower**, and **dis-joint with** have been standardized in the Simple Knowledge Organization System (SKOS) [5] and in the Web Ontology Language (OWL 2)⁷. We forward the reader to Table 1 for a very small subset of such relations.

⁵ <http://InPhO.cogs.indiana.edu>

⁶ <http://rtw.ml.cmu.edu/rtw/>

⁷ <http://www.w3.org/TR/owl2-overview/>

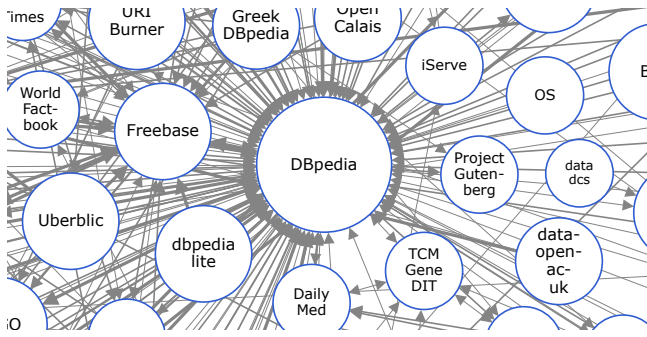


Figure 2: A small fragment of the web of data. DBpedia is a de-facto hub of the linked open data cloud.

Figure 1 depicts a small fragment of an RDF graph modeling the domain of philosophers and philosophical works and ideas. Each node in the graph represents a particular object and carries a unique identifier. Links between entities in different data repositories establish semantic connections. For instance, the object with the label “George Moore” in DBPEDIA is identical to the object with the label “G.E. Moore” in the Indiana Philosophy ontology which is expressed by the `sameAs` link between the two entities. Using RDF representation allows us to not only resolve ambiguities but also to *aggregate* knowledge about individual entities.

When published under open licenses, data sets may join the Linked Open Data cloud, which can help increase awareness of new collections and facilitates integration with other LOD repositories. For a small fragment of the LOD web see Figure 2.

3. RELATED WORK

Apart from the INPHO project, there have been many attempts to digitize data in humanities by utilizing Semantic Web technologies. The system TALIA [47] enables, for example, philosophy scholars to compare manuscripts, search for specific topics in hand-written paragraphs, and link movie files to the topics. TALIA employs RDF as underlying representation formalism. TALIA has been developed within the OAC project that creates a model and data structure to enable the sharing of scholarly annotations across annotation clients, collections, media types, applications, and architectures [46]. One of the project’s goals is the generalization of the tools developed specifically for the philosopher Nietzsche in the Hyper-Nietzsche project [14].

Additionally, there are several projects which focus on providing a Semantic Web architecture for specific fields. For instance, the BRICKS project maintains a service-oriented infrastructure to share knowledge and resources in the Cultural Heritage domain with RDF [52] while the JEROMEDL project designed an architecture for social semantic digital libraries [29].

More recently, the Digitized Manuscripts to Europeana (DM2E)⁸ project developed tools which enable humanities researchers to interact with the Semantic Web. If we com-

⁸ <http://dm2e.eu/>

pare DM2E with our linked humanities project, DM2E allows users to annotate digital humanities collections with existing vocabularies such as SKOS while LODE allows users to use their own project-specific RDF representation (like e.g. in the InPhO project) and provides a framework for browsing, linking, and enhancing this representation. As such, the projects complement one another and we will continue to explore possible synergies of the two projects. One of DM2E’s objectives is to parse manuscripts and make their data available in Europeana [24], which is a multi-lingual online collection of millions of digitized items from European museums. In this context, DM2E members identified, for instance, some challenges for building an ontology about the philosopher Ludwig Wittgenstein by pointing out different ontological concepts and modeling alternatives [50]. Máchá et al. [32] extends this work by approaching the problem of modeling agreement, differences, and disagreement. Another important aspect of DM2E is provenance tracking [16, 17] which makes it possible to identify the sources of linked data with the help of a consistent data model. In this context DM2E developed the tool PUNDIT [23] which allows the annotation, augmentation, contextualization, and externalization of Web Resources of manuscripts so as to make these manuscripts available as machine-processable data.

This large amount of projects with the aim of providing tools for the creation or maintenance of small and specialized digital humanities datasets, underlines the need to establish high-quality links between these datasets and the LOD cloud. These links can then be utilized for controlled enrichment of datasets with further information.

With respect to the problem of entity matching, a large amount of automated and semi-automated instance matching frameworks have emerged. In the context of the DM2E project, the linking framework SILK [27] is utilized for establishing links between data items within different Linked Data sources. It has been integrated into the workflow system OMNOM [22]. In OMNOM the user can for example parse different file formats and upload them to the DM2E triple store. Thereby, the input and output parameters of the work-flow steps are intuitively connected via drag-and-drop. Furthermore, each work-flow can be executed by creating a specific instance of the work-flow with concrete input and output files. Within SILK the user can define work-flows in form of a tree structure, which describe how the interlinking process is performed. Creating these work-flows, however, requires a significant knowledge about RDF and about specific linking techniques such as different syntactic similarity measures. To that end, SILK provides an approach to actively learn the work-flows by forcing the user to accept and decline a number of matching candidates [26]. The LODE framework aims to hide even more complexity from the user, by providing the user valuable suggestions and concise information about the respective candidate entities, without the need to learn a linking scheme. Consequently, we enable domain experts, who have very limited knowledge of Semantic Web technologies to link and enrich their data by using simple drag-and-drop techniques. Furthermore, our use case is such that the user needs to have full control over the link creation process in order to preserve the quality of the local repository. Thus, any automatic link establishment tech-

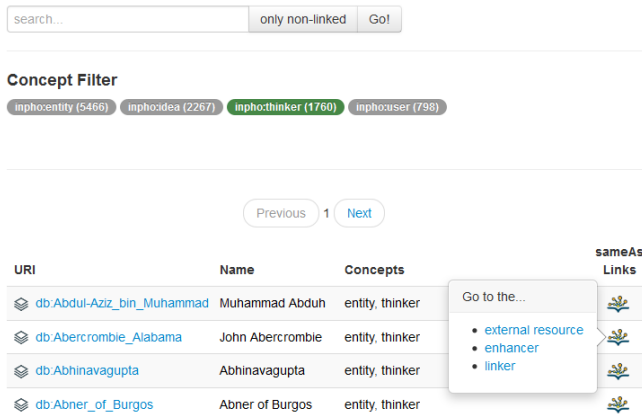


Figure 3: Screenshot of the browser interface that lists all search results for “InPhO:thinker”. The concept filters can be used to focus on particular objects types, here the InPhO types “entity”, “thinker”, and “user”.

niques without the supervision of the user is not suitable for our use case.

In addition to SILK, many other tools incorporate the idea of active or supervised learning for matching. Existing supervised approaches are based on learning linear classifiers or learning threshold based Boolean classifiers [3, 26]. An example for the first category is MARLIN (Multiple Adaptive Record Linkage with Induction) [7] which uses support vector machines for learning. Examples for the second category include ACTIVE ATLAS [54] and TAILOR [18] which utilize decision trees for record linkage. In active learning, Arasu et al. [2] developed an approach with which a certain precision can be reached. However, recall might still be low after learning. The tool RAVEN [35] focuses on Boolean and weighted classifiers. Genetic algorithms [13, 36, 26] are a common technique for finding solutions for active learning approaches. There also exist several fully automated non-supervised instance matching systems like LOGMAP [28], CODI [43, 25, 44], RIMOM [31]. Moreover, there is a large body of work on approaches that exploit schema information to make the resulting alignments more coherent [34, 40, 41, 28, 45, 37]. While these systems do not require any additional user input they often rely on an expressive ontology for the alignment process. However, these systems’ alignments are often error-prone [42] and thus not suitable for establishing links of very high quality. We refer the reader to the instance matching track of the ontology alignment evaluation initiative (OAEI) for a listing of further systems [1]. Again, the focus of the LODE project is the combination of linking algorithms that do not require learning and an expressive schema with intuitive user interfaces for semi-automatic alignment tasks. Furthermore, LODE requires the linking algorithms to be interactive, responding to a query within less than a second. Moreover, since the link creation is semi-automatic, a ranking of possible entities is required.

4. THE LODE FRAMEWORK



Figure 4: Screenshot of the linking interface listing the linking candidates for the InPhO entity “Ludwig Wittgenstein.” The link candidate ranking on the right is influenced by the lexical similarity and the semantics of the source entity. Users can easily identify the correct linking candidate by browsing its semantic properties and Wikipedia abstract.

The linked open data enhancer (LODE)⁹ framework is a set of integrated tools that allow digital humanists, librarians, and information scientists to connect their data collections to the linked open data cloud. The initial step is to model the respective collection with some RDF serialization. For this task, tools from e.g. the DM2E project can be utilized. Once an RDF representation exists, the LODE framework loads the RDF representation of the collection and provides several components for browsing, integrating, and enriching the collections. While it leverages state of the art concepts and algorithms, the focus is on intuitive interfaces that shield the users from the algorithmic intricacies and the complexities of RDF serialization.

In the following, we describe the three modules of LODE in more detail.

4.1 Content Browsing

The content browser allows to explore the RDF dataset in an intuitive way by providing a search-based interface that resembles those of standard search engines. Users can enter keywords to search for entities in the locally stored RDF serialization of the project content. All the objects in the RDF dataset that match a given keyword query are categorized according to their types. The search field features auto completion and allows filtering by type. The syntax for this latter filter technique is adapted from the typical search engine syntax which allows searching for terms within a specific site with the command *site:url searchterm*. We adopted this syntax and applied it to types such that the user can search with e.g. *concept:human Wittgens* for an individual whose label matches the string *Wittgens* and which is an instance

⁹ <http://lode.informatik.uni-mannheim.de>

Vocabulary ¹¹	Relation	Type	Usage ¹²
owl	sameAs	I	34.40%
rdfs	subClassOf	C	15.28%
rdfs	subPropertyOf	P	11.53%
owl	inverseOf	P	6.65%
skos	broader	I/C	4.87%
owl	equivalentClass	C	4.40%
skos	narrower	I/C	3.84%
owl	disjointWith	C/P	3.56%
owl	equivalentProperty	P	3.09%
skos	related	I/C	2.81%

Table 1: Most frequent relation properties in the LOD web.

of the type *Human*. In addition, the LODE search interfaces provide dynamic faceted search. The results are clustered according to the *sameAs* relations so that every unique entity is only displayed once in the result. Figure 3 provides a screenshot of the main functionality of the content browser.

The content browser is the starting point for navigation to the instance overview, linker, and enhancer. The overview displays all data of an instance which are stored in the local triple store. For each local entity, the linking module suggests high-quality linking candidates from the Linked Open Data cloud (here: DBpedia) to the user. If a link exists, the enhancement module gives the user the possibility to decide which information of the LOD cloud is reliable enough to be added to the local RDF repository.

As with every LODE component, the browsing interface works “out of the box” for arbitrary RDF repositories and does not need prior configuration steps. Like all other pages, the browser provides tooltips which contain the full URI of any displayed instance, concept, or property. The tooltips are clickable and lead the user to the original resource taking advantage of the important linked open data principle that every entity’s URI is resolvable.

4.2 Content Linking

When an entity is selected in the browsing interface the user can initiate the linking component. At this point, the LODE framework supports linking to DBPEDIA which is a hub in the link open data cloud (see Figure 2), providing links to numerous LOD collections through *sameAs* links.

Please note that, for the particular applications LODE is aimed at, the user requires full control over the linking process in order to assure the quality standard of the local RDF repository. Hence, the purpose of the linking component is to recommend high quality suggestions from which the user then can select the correct one. Additionally, we especially

¹¹Prefixes taken from <http://prefix.cc>

¹²The usage has been computed from data representing the LOD web. The data has been crawled by taking seeds from the Billion Triple Challenge <http://km.aifb.kit.edu/projects/btc-2012/> and Datahub <http://datahub.io>.

Vocabulary ¹¹	Property	Usage ¹²
foaf	name	53.14%
rdfs	label	40.02%
foaf	givenname	21.46%
foaf	accountname	20.34%
foaf	family_name	18.46%
foaf	firstname	13.96%
foaf	surname	13.03%
skos	preflabel	8.62%
foaf	openid	7.50%
dcterms	identifier	5.81%

Table 2: Most frequent label properties in the LOD web.

aim at supporting non-technical domain experts for data integration by designing simple user interfaces and providing them valuable additional information of the linking candidates to facilitate the alignment decision. Figure 4 depicts the linking interface for the INPHO entity “Ludwig Wittgenstein” and some of the linking candidates.

In addition to *sameAs* links, LODE supports different types of links modeling relationships between individuals, typed (concept), and properties. We utilize as subsets of SKOS [5] and also include several relations from the Web Ontology Language (OWL 2). Table 1 lists the core link types supported by LODE. We differentiate between relations between concepts (C), properties (P), and individuals (I). This list can be extended by the user at any time.

The content linker performs the following steps to retrieve and display the linking candidates for a candidate entity E to the user.

First, the linker component extracts a set of search terms from property assertions of entity E in the local RDF repository. To identify these terms, the algorithm maintains a list of the most frequent properties describing the instance (like e.g. the *label*). Table 2 depicts a list of common lexical properties of entities and their usage statistic. Of course, it is possible to modify and extend this list. However, if the local RDF repository follows modeling standards common to linked data repositories the list of properties should be sufficient as it covers a large fraction of the properties used for labeling entities.

With the previously extracted search terms as input, the linking component generates a list of potential linking candidates for E based on two algorithms. Both algorithms are required to be interactive, returning a result ranking within one second. Due to common hashing and indexing techniques our algorithms’ complexity is sublinear with respect to the total number of possible instances. Section 4.2.1 and Section 4.2.2 provide further details about the linking algorithms.

Finally, LODE extracts context for each linking candidate to help the user identify the correct alignment without overwhelming her with too much information. The context is extracted so as to help the user discriminate between entities with identical labels and names. The underlying selection

process is explained in Section 4.2.3. Figure 4 shows how context (abstract, labels, etc.) is presented to the user so as to help the user with the linking decision.

4.2.1 SPARQL-Algorithm

The first linking algorithm uses SPARQL queries to search for matching candidates in the LOD cloud. As an example, we employ DBPEDIA as SPARQL endpoint. However, please note that we are able to apply the following search technique to any other triple store.

The SPARQL queries search for the exact search terms within the label and the abstract. Listing 1 provides a simplified example of such a SPARQL query. Especially, when we include the search within the abstract, we obtain a relatively large amount of linking candidates.

```
PREFIX dbp: <http://dbpedia.org/property/>
SELECT DISTINCT ?instance ?value
WHERE {
  ?instance dbp:abstract ?value .
  ?value <bif:contains> searchterm .
}
```

Listing 1: Search by abstract

This leads to the requirement to rank the retrieved linking candidates in a second step. For this ranking, we apply the Levenshtein similarity [30] between the search term of the local instance and the linking candidates. If more than one search term exists, the maximum similarity is taken. Intuitively, the higher the similarity, the higher the ranking of the candidate. We used the Levenshtein similarity since it can handle spelling errors like e.g. *Ludwig* and *Ludwik*.

Within this algorithm, we also consider structural information by leveraging known semantic relationships between types of the involved RDF datasets [44]. A matching candidate is inferred to be disjoint if its types are disjoint with the types of the searched entity. For instance, if the “Thinker” type in INPHO and the “PhilosophicalTradition” type in DBPEDIA are disjoint then the linking interface will exclude all entities of the later type as linking candidates for equivalence links. The disjointness relationships has to be established only once by the user of LODÉ.

Finally, we apply some DBPEDIA specific optimizations. In particular, we evaluate whenever the URI of the found instance is a *redirect* or a *disambiguation page* and resolve the URI if this is the case.

4.2.2 WikiStat Algorithm

The WIKI STAT algorithm is based on the idea of exploiting Wikipedia’s link structure to compute, for a given search string, the conditional probability of a Wikipedia article given the search string. Consider, for example, the article about philosophy which contains a link to the article with URI <http://en.wikipedia.org/wiki/Plato> and anchor text “Plato.” This link would increase the conditional probability of the URI given the search string “Plato.” As in Dutta et al. [15], we utilize the WIKIPEDIA preprocessor WIKIPREP [20, 21] which computes a table consisting of the anchor-text a , the source URI, and the corresponding target URI u . We use

Anchor a	Simplified URI u	number $\#(u, a)$
Plato	Plato (Philosopher)	3560
PLATO	PLATO (computer system)	47
Plato	Plato, Missouri	20
Plato	Plato (crater)	15
Plato	Beer measurement	13
Plato	Plato, Magdalena	9
Platon	Plato (Philosopher)	6

Table 3: An excerpt of the search result of the WikiStat table when searching for the anchor texts “Plato” and “Platon”. The entries are ordered by $\#(u, a_n)$, descending.

these tables to compute the conditional probabilities. However, as opposed to previous work, we have to incorporate multiple search strings a_1, a_2, \dots, a_n since an entity can have multiple properties that relate the entity to its label or name. Each search term extracted from the properties in Table 2 has to be matched against possible anchors used to link to an article in Wikipedia.

Let u be a Wikipedia URI and a_1, \dots, a_n be the extracted search strings. Then, the ranking of the matching candidates is based on the following conditional probability

$$P(u|a_1 \vee \dots \vee a_n) = \frac{P(u, a_1 \vee \dots \vee a_n)}{P(a_1 \vee \dots \vee a_n)} = \frac{\#(u, a_1)/N + \dots + \#(u, a_n)/N}{\#(a_1)/N + \dots + \#(a_n)/N},$$

where $\#(u, a_i)$ is the number of (u, a_i) pairs, that is, the number of Wikipedia links to entity u with anchor text a_i , $\#(a_i)$ is the number of Wikipedia links with anchor a_i , and N is the number of all Wikipedia links. The ranking of the linking candidates is determined by sorting the conditional probability of all URIs u in descending order.

Since we are only interested in the final ranking, we are able to further simplify the above equation. In fact, it is sufficient to compute

$$\#(u, a_1) + \dots + \#(u, a_n)$$

for every URI u because $\#(a_1) + \dots + \#(a_n)$ is constant for given anchor texts a_1, a_2, \dots, a_n and N cancels out.

For efficiency reasons, we precomputed all numbers $\#(u, a)$ for every URI u and anchor text a and stored these in a relational database table. Table 3 depicts an excerpt of the table for the anchor texts a_1 = “Plato” and a_2 = “Platon”. The ranking of the linking candidates is now computed by selecting every row in the table where the anchor a matches a search string, aggregating the result set with respect to the URI, and sorting the aggregation according to the sum of all numbers $\sum_i \#(u, a_i)$ in descending order. Listing 2 shows an example SQL query which again queries for anchor texts a_1 = “Plato” and a_2 = “Platon”.

```
SELECT u, SUM(\#(u, a)) AS s FROM table
WHERE a = 'Plato' or a = 'Platon'
```

GROUP BY u ORDER BY s DESC;

Listing 2: WikiStat SQL query for a_1 =“Plato” and a_2 =“Platon”

In this query, the highest ranking is achieved by the simplified URL u =“Plato (Philosopher)” having the quantity $\#(u, a_1) + \#(u, a_2) = 3566$.

4.2.3 Context Selection

The URI of an entity is often not sufficient for the user to select the correct entity from the set of matching candidates. Even in the presence of labels, choosing the correct entity might be difficult due to ambiguous labels. Therefore, we provide the user with contextual information in form of the entity’s properties that more closely characterizes each of the matching candidates and help the user to select the correct entity. Presenting every property of an entity would overwhelm the user with information. Hence, we developed an algorithm that presents *discriminating* properties and types only. After experimenting with alternative, more sophisticated adaptations of TF-IDF, we noticed that the frequency of properties is most helpful in identifying valuable assertions. Thus, we implemented the following algorithm.

Let p be a property and \mathcal{E} be the set of all entities in DBPEDIA. Furthermore, let $I_p : \mathcal{E} \rightarrow \{0, 1\}$ be an indicator function for property p defined as:

$$I_p(\mathbf{E}) = \begin{cases} 1 & \text{if } \mathbf{E} \text{ has as least one assertion for } p \\ 0 & \text{otherwise} \end{cases}$$

Then, the frequency f_p for the property p is defined as

$$f_p = \frac{\sum_{\mathbf{E} \in \mathcal{E}} I_p(\mathbf{E})}{|\mathcal{E}|}.$$

The frequencies f_p are precomputed for each property. The properties are now sorted according to their frequency in descending order. Finally, we present only the k most frequent properties of an entity as its context. Analogously to properties, we apply the same approach to the types of an entity presented to the user.

4.3 Content Enhancing

After a link between a local and external entity has been established, the enhancing component facilitates the addition of content from the Linked Open Data cloud to the local repository. Since the data of several information extraction projects such as DBPEDIA contains factual errors and inaccuracies, we allow the user to manually drag and drop LOD content to the local repository. This ensures that the quality of the local collection is not compromised. The human domain expert verifies the correctness of facts by dragging these facts to the local repository.

The main objective of the component is to support non-technical users with (a) an intuitive interface and (b) high quality enhancement suggestions. Figure 5 shows a screenshot of our enhancement interface. The local RDF repository is depicted on the left while DBPEDIA is located on the right. The interface avoids overwhelming the user with too many potential enhancement candidates by presenting

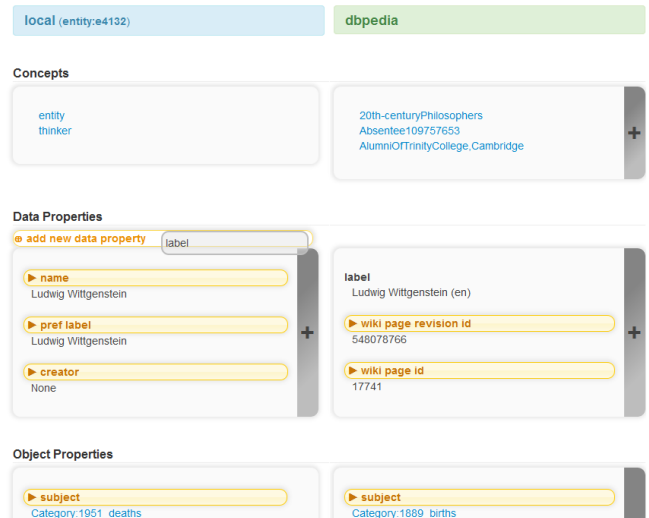


Figure 5: Screenshot of the enhancer interface listing the data and meta-data of the local RDF repository (on the left) and related DBPedia content (on the right). Data, concepts, and properties related to the source entity (here: Wittgenstein) can be dragged and dropped to the local repository to enrich the project content.

only excerpts of the most frequent class and property assertions. Here, we utilize the same algorithm as described in Section 4.2.3.

If the user has decided to enhance a specific class or property value, she can simply drag and drop it to the desired position. During this process, the user gets all possible drop areas highlighted. In Figure 5, the user decided to enhance the local entity “Ludwig Wittgenstein” with a property assertion stating that the entity has a label “Ludwig Wittgenstein”.

In case of property assertions, the user has the choice between adding the value to an existing property or creating a new property and assigning the value to this property. If there exists more than one value for a specific property, the user can select which of the given values are to be added to the local collection. Additionally, we provide the possibility to delete concepts, properties, or values.

Internally, LOD E creates new RDF triples in the local RDF repository for each enhancement operation. In our example, LOD E will add the new triple “thinker:t4132 rdfs:label ’Ludwig Wittgenstein’@en” to the local RDF repository. By keeping the target URI of DBPEDIA unchanged, it is easily possible to identify the provenance of the enhancement.

5. EXPERIMENTS

Many digital humanities collections are concerned with three different types of entities, namely persons, documents, and concepts. The following experiments assess the performance of the two linking algorithms described in Section 4.2 on these different types of entities by using INPHO data. Moreover, we also use a large collection of subject predicate ob-

algorithm	abstract	label
SPARQL-A	✓	
SPARQL-L		✓
SPARQL-AL	✓	✓
WikiStat	-	-

Table 4: Configurations of the SPARQL and WikiStat algorithms (“A” = abstract considered, “L” = label considered)

	evaluated	total
INPHO thinker	1452	1758
INPHO journal	219	1122
INPHO idea	236	2322
NELL	921	≈ 2 Mio.

Table 5: Number of entities of the gold standard compared to the the total number entities in the datasets.

ject triples extracted by a well-known information extraction system, namely the Never Ending Language Learning (NELL) project. The experiments are meant to investigate the quality of the real-time interactive linking algorithms with respect to varying types of entities and varying quality of the data.

5.1 Experimental Setup and Datasets

We compare the WIKI STAT algorithm (see Section 4.2.2) with the SPARQL algorithm (see Section 4.2.1) using different configurations. In particular, we considered the properties `rdfs:label` (abbr. *L*) and `rdfs:abstract` (abbr. *A*) for the SPARQL queries. Thus, we obtain four different algorithms, which are depicted in Table 4.

For each of the four methods we take a set of N entities for which a `owl:sameAs` relation to DBPEDIA exists. For each of these entities, we compute 10 matching candidates in form of a ranked list. In order to assess the accuracy of the algorithm we compute the average mean reciprocal rank (MRR). For each entity the linking algorithms generate a ranking of which at most one entry is the correct one. The MRR of a number of rankings is defined as

$$MRR = \frac{1}{|N|} \sum_{i=1}^{|N|} \frac{1}{rank_i}, \quad (1)$$

where $rank_i$ represents the position of correct entity in the returned ranking. By standard convention, we set $\frac{1}{rank_i} = 0$ if the correct entity is not in the ranking. In addition to the MRR, we also measure the average time needed to compute the ranking for one entity. All experiments were executed on a virtual machine running on a two core Intel Xeon 4C E5-2609 80W processor with 2 GB of RAM.

For our evaluation we created gold standards using data from the INPHO and NELL projects. Both projects provide a large collection of subject-predicate-object triples with INPHO focusing on the domain of philosophy and NELL being more focused on popular domains such as sports and movies. For NELL we used an existing gold standard¹⁰ which provides

¹⁰ <https://madata.bib.uni-mannheim.de/65/>

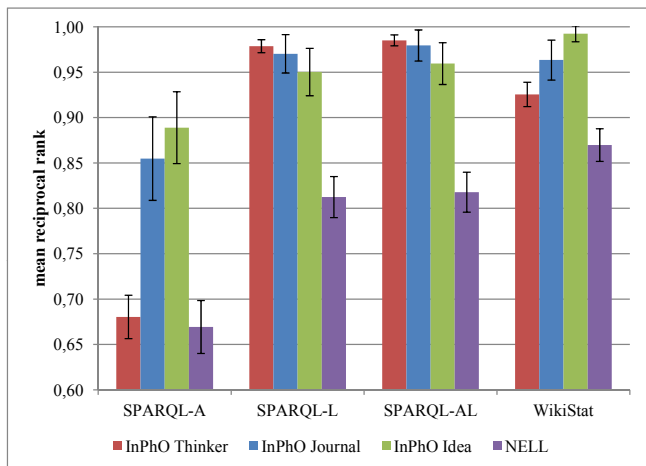


Figure 6: MRR for the different configurations of the search algorithms including the respective 95% confidence intervals. With respect to SPARQL, we observe that considering the label is crucial. The SPARQL-Algorithm is stronger on thinkers and journals while WikiStat has better results for ideas and on the NELL benchmark.

`owl:sameAs` links to DBPEDIA entities for the subject and the object for 1200 NELL triples.

The remaining three gold standard data sets were extracted from the Indiana Philosophy Ontology INPHO¹¹ project. Like in many humanities domains, the data in the INPHO project mainly describes entities representing persons (Thinkers), documents (Journals), philosophical concepts (Ideas), and their relations. For each of these categories, we manually created separate gold standards.

Table 5 illustrates the number of individuals per benchmark and category for the gold standards compared to the total available number of entities.

5.2 Results

The MRR values and the average running time with corresponding 95% confidence intervals are depicted in Figure 6 and Figure 7, respectively. Each figure has four groups of bars representing the four different configurations depicted in Table 4. Overall, the linking algorithms achieve MRR values of over 0.95 with the SPARQL-AL configuration on all INPHO entity types with average running times of 1.5 seconds or less. On the NELL benchmark, the WIKI STAT algorithm has a MMR over 0.85 and average running times of 1.7 seconds.

If we compare the MMR results for the different configurations of the SPARQL-based algorithm, we see that considering only the abstract (abbr. *A*) results in the lowest MRR results. Considering the abstract and the label (abbr. *AL*) produces slightly better results than if we only consider the label (abbr. *L*). However, recall can be slightly increased if the label is also considered. Searching in the abstract only resulted in about the same running times as searching in

¹¹ <https://InPhO.cogs.indiana.edu/>

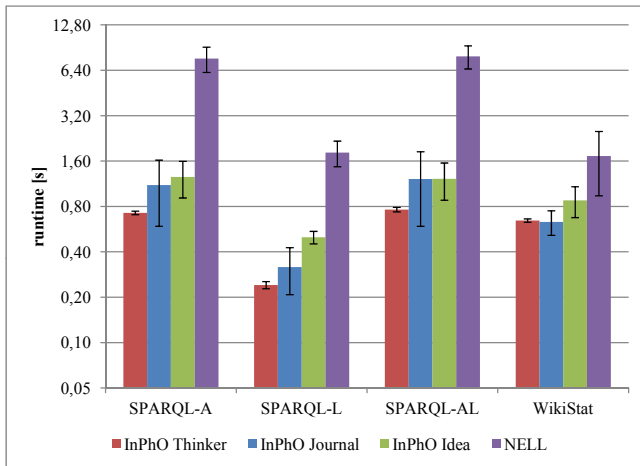


Figure 7: Average runtime for the different configurations of the search algorithms including the respective 95% confidence intervals in logarithmic scale. The lowest average runtimes are measured for the SPARQL-L configuration. Furthermore, average runtimes were longer for the NELL benchmark.

both the label and the abstract. Considering the label only resulted in the shortest running times which is due to the length of a label being shorter than the length of an abstract in most cases.

Our experiments show that the WIKI STAT algorithm compared to the SPARQL algorithm with label and abstract has different strengths. While the SPARQL algorithm has been stronger on entities representing persons (here: *Thinkers*) and documents (here: *Journals*), WIKI STAT achieved better results for philosophical concepts (here: *Ideas*) and the NELL dataset. The reason is that for persons and documents the naming is more accurate while philosophical concepts often have several possible equivalent names. Since the keywords that link to one specific WIKI PEDIA entry cover multiple possible names while the abstract or the label often contains only one name, the NELL algorithm has a higher recall in these cases. Runtimes for the WIKI STAT and SPARQL-AL algorithm were comparable. If we compare the position of the correct matching candidates displayed in Figure 8, we observe that both algorithms were able to rank the correct candidates at the first position in over 97 % of the cases for the INPHO and in over 77 % of the cases for the NELL gold standard.

In all cases we obtain increased running times for the NELL gold standard, since NELL entities have often multiple labels.

6. CONCLUSION AND FUTURE WORK

The aim of our Linked Humanities project is to enable non-technical humanities scholars to integrate and enrich local RDF repositories with the Linked Open Data cloud. To that end, we developed the linked open data enhancer (LODE) which provides intuitive user interfaces for linking and enhancing local RDF repositories while maintaining high quality collections. The evaluation of two linking algorithms showed that they are able to provide high quality linking

candidates with a response time of under 1.5 second. We observed that the SPARQL algorithm performed better when linking persons and documents while WIKI STAT gained higher results for suggesting candidates for philosophical concepts and for the NELL benchmark.

In future work, we will perform a user study to evaluate the entire framework including the enhancement component. One aspect we aim to examine are comparisons between different (more sophisticated) context selection algorithms. Furthermore, we plan to add additional “out of the box” repositories apart from DBPEDIA. We are also continuously improving the linking and enhancement algorithms. Additionally, we plan to extend the enhancement interface so as to also allow the manual addition of novel content.

7. ACKNOWLEDGMENTS

This work has partly been funded by the DFG (NI 1364/1-1) and NEH (HG-50032-11). Many thanks to Kai Eckert for providing us with valuable feedback.

8. REFERENCES

- [1] J. L. Aguirre, B. C. Grau, K. Eckert, J. Euzenat, A. Ferrara, R. W. van Hague, L. Hollink, E. Jiménez-Ruiz, C. Meilicke, A. Nikolov, et al. Results of the ontology alignment evaluation initiative 2012. In *Ontology Matching Workshop*, pages 73–115, 2012.
- [2] A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 783–794. ACM, 2010.
- [3] A. Arasu, M. Götz, and S. Kaushik. Active learning of record matching packages, 2011. US Patent App. 13/084,527.
- [4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference*, pages 722–735. Springer, 2007.
- [5] T. Baker, S. Bechhofer, A. Isaac, A. Miles, G. Schreiber, and E. Summers. Key choices in the design of simple knowledge organization system (skos). *Web Semantics: Science, Services and Agents on the World Wide Web*, 20:35–49, 2013.
- [6] T. Berners-Lee. Linked data – design issues, 2006. <http://www.w3.org/DesignIssues/LinkedData.html>.
- [7] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD conference*, pages 39–48. ACM, 2003.
- [8] C. Bizer, T. Heath, and T. Berners-Lee. Linked data – the story so far. *International Journal on Semantic Web and Information Systems*, 2012.
- [9] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.
- [10] C. Buckner, M. Niepert, and C. Allen. From encyclopedia to ontology: toward dynamic

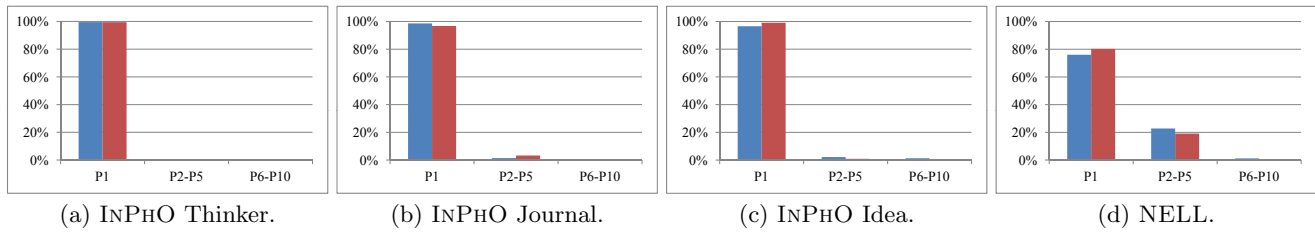


Figure 8: Position of the correct matching candidates on a percentage basis if the entity has been under the top 10 search results (blue = SPARQL, red = WikiStat).

representation of the discipline of philosophy. *Synthese*, 182(2):205–233, 2011.

[11] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of AAAI*, 2010.

[12] G. Crane, B. Fuchs, and D. Iorizzo. The humanities in a global e-infrastructure: a web-services shopping-list. *UK e-Science All Hands Meeting*, 2007.

[13] J. De Freitas, G. L. Pappa, A. S. da Silva, M. A. Gonçalves, E. Moura, A. Veloso, A. H. Laender, and M. G. de Carvalho. Active learning genetic programming for record deduplication. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[14] P. D’Iorio. Nietzsche on new paths: The hypernietzsche project and open scholarship on the web. *Friedrich Nietzsche, Edizioni e interpretazioni, Pisa ETS*, <http://www.hypernietzsche.org/doc/files/new-paths.pdf>, 2007.

[15] A. Dutta, C. Meilicke, M. Niepert, and S. Ponzetto. Integrating open and closed information extraction: Challenges and first steps. In *Proceedings of 1st International Workshop on NLP and DBpedia, October*, pages 21–25, 2013.

[16] K. Eckert. Provenance and annotations for linked data. *International Conference on Dublin Core and Metadata Applications*, 2013.

[17] K. Eckert, D. Ritze, K. Baierer, and C. Bizer. Restful open workflows for data provenance and reuse. In *To appear. Poster at WWW conference*. Citeseer, 2014.

[18] M. G. Elfeky, V. S. Verykios, and A. K. Elmagarmid. Tailor: A record linkage toolbox. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 17–28. IEEE, 2002.

[19] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. T. dos Santos. Ontology alignment evaluation initiative: Six years of experience. *J. Data Semantics*, 15:158–192, 2011.

[20] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, pages 1301–1306, 2006.

[21] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[22] S. Gradmann, J. Iwanowa, E. Dröge, S. Henricke, and V. Trkulja. Modellierung und ontologien im wissensmanagement. *Information-Wissenschaft und Praxis*, pages 149–165, 2013.

[23] M. Grassi, C. Morbidoni, M. Nucci, S. Fonda, and F. Piazza. Pundit: augmenting web contents with semantics. *Literary and linguistic computing*, 28(4):640–659, 2013.

[24] B. Haslhofer and A. Isaac. data. europeana. eu: The europeana linked open data pilot. In *International Conference on Dublin Core and Metadata Applications*, pages 94–104, 2011.

[25] J. Huber, T. Szttyler, J. Noessner, and C. Meilicke. Codi: Combinatorial optimization for data integration—results for oaei 2011. *Ontology Matching*, page 134, 2011.

[26] R. Isele and C. Bizer. Active learning of expressive linkage rules using genetic programming. *Web Semantics: Science, Services and Agents on the World Wide Web*, 23:2–15, 2013.

[27] R. Isele, A. Jentzsch, and C. Bizer. Silk server-adding missing links while consuming linked data. In *COLD*, 2010.

[28] E. Jiménez-Ruiz, B. C. Grau, Y. Zhou, and I. Horrocks. Large-scale interactive ontology matching: Algorithms and implementation. In *ECAI*, volume 242, pages 444–449, 2012.

[29] S. R. Kruk, T. Woroniecki, A. Gzella, M. Dabrowski, and B. McDaniel. The anatomy of a social semantic digital library. In *Proc. of International Semantic Web Conference (ISWC), Athens, GA, USA*, volume 2006, 2006.

[30] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[31] J. Li, J. Tang, Y. Li, and Q. Luo. Rimom: A dynamic multistrategy ontology alignment framework. *Knowledge and Data Engineering, IEEE Transactions on*, 21(8):1218–1232, 2009.

[32] J. Mácha, R. J. Falch, and A. Pichler. Overlapping and competing ontologies. In *Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: metadata, vocabularies and techniques in the Digital Humanities*, page 5. ACM, 2013.

[33] F. Manola and E. Miller. RDF primer. Technical report, WWW Consortium, Feb 2004. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.

- [34] C. Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, University Mannheim, 2011.
- [35] A.-C. N. Ngomo, J. Lehmann, S. Auer, and K. Höffner. Raven—active learning of link specifications. In *Ontology Matching Workshop*, pages 25–37, 2011.
- [36] A.-C. N. Ngomo and K. Lyko. Eagle: Efficient active learning of link specifications using genetic programming. In *The Semantic Web: Research and Applications*, pages 149–163. Springer, 2012.
- [37] M. Niepert. A Delayed Column Generation Strategy for Exact k-Bounded MAP Inference in Markov Logic Networks. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2010.
- [38] M. Niepert, C. Buckner, and C. Allen. A dynamic ontology for a dynamic reference work. In *Proceedings of the 7th ACM/IEEE joint conference on Digital libraries (JCDL)*, pages 288–297, 2007.
- [39] M. Niepert, C. Buckner, and C. Allen. Answer set programming on expert feedback to populate and extend dynamic ontologies. In *Proceedings of the 21st FLAIRS*, pages 500–505, 2008.
- [40] M. Niepert, C. Meilicke, and H. Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.
- [41] M. Niepert, J. Noessner, C. Meilicke, and H. Stuckenschmidt. Probabilistic-logical web data integration. In *Reasoning Web. Semantic Technologies for the Web of Data*, volume 6848, pages 504–533. Springer Berlin Heidelberg, 2011.
- [42] J. Noessner, F. Fallahi, E. Kiss, and H. Stuckenschmidt. Interactive data integration with mappingassistant. In *Demo Paper at the 10th International Semantic Web Conference ISWC*. Citeseer, 2011.
- [43] J. Noessner and M. Niepert. Codi: Combinatorial optimization for data integration. In *Ontology Matching Workshop*, 2010.
- [44] J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt. Leveraging terminological structure for object reconciliation. In *Proceedings of the 7th Extended Semantic Web Conference*, pages 334–348, 2010.
- [45] J. Noessner, M. Niepert, and H. Stuckenschmidt. Coherent top-k ontology alignment for owl el. In *Proceedings of the 5th International Conference, on Scalable Uncertainty Management (SUM)*, pages 415–427, 2011.
- [46] M. Nucci, M. Barbera, C. Morbidoni, and D. Hahn. A semantic web powered distributed digital library system. In *ELPUB2008*, 2008.
- [47] M. Nucci, S. David, D. Hahn, M. Barbera, et al. Talia: A framework for philosophy scholars. In *SWAP*, 2007.
- [48] M. Paşca. Organizing and searching the world wide web of facts - step two: harnessing the wisdom of the crowds. In *Proceedings of the 16th international conference on World Wide Web*, 2007.
- [49] M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *Proceedings of the 20th AAAI Conference on Artificial Intelligence*, 2006.
- [50] A. Pichler and A. Zöllner-Weber. Sharing and debating wittgenstein by using an ontology. *Literary and linguistic computing*, 28(4):700–707, 2013.
- [51] A. Pollock and A. Hockley. What’s wrong with internet searching. *D-Lib Magazine*, March 1997.
- [52] T. Risse, P. Knezevic, C. Meghini, R. Hecht, and F. Basile. The bricks infrastructure—an overview. In *8th International Conference EVA*, 2005.
- [53] M. Stollberg, H. Lausen, R. Lara, Y. Ding, S.-K. Han, and D. Fensel. Towards semantic web portals. In *Proceedings of the workshop on Application Design, Development and Implementation*, 2004.
- [54] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the eighth ACM SIGKDD conference*, pages 350–359. ACM, 2002.