Datenbank Spektrum manuscript No.
(will be inserted by the editor)

Extraktion, Mapping und Verlinkung von Daten im Web

Phasen im Lebenszyklus von Linked Data

Sören Auer · Jens Lehmann · Axel-Cyrille Ngonga Ngomo · Claus Stadler · Jörg Unbehauen

Received: date / Accepted: date

Zusammenfassung In diesem Artikel geben wir einen Überblick über verschiedene Herausforderungen des Managements von Linked Data im Web. Mit der DBpedia Wissensextraktion aus Wikipedia, dem skalierbaren Linking von Wissensbasen und dem Mapping relationaler Daten nach RDF stellen wir drei Ansätze vor, die zentrale Phasen des Lebenszyklus von Daten im Web ausmachen.

Schlüsselwörter RDF · Linked Data · Semantic Web

1 Einführung

Linked (Open) Data ist eine auf Web-Standards basierende Methode zur einheitlichen Speicherung, Publikation und Verknüpfung von Daten unterschiedlicher Art und Herkunft im RDF-Format. Seit der Prägung des Begriffs im Jahre 2006 wurden bereits hunderte von Datenquellen bzw. Wissensbasen unterschiedlicher Domänen als Linked Open Data (LOD) veröffentlicht und miteinander vernetzt. Auch viele Unternehmen (u.a. Google [9], Yahoo! [21], Oracle [7, 36]), Organisationen (u.a. BBC [16], New York Times) und wissenschaftliche Projekte haben begonnen Linked Data zu veröffentlichen und zu nutzen. Die standardisierte und generische Repräsentation der Informationen ermöglicht eine einfache Zugänglichkeit und Nutzung heterogener Informationen in vielzähligen Anwendungen. Weiterhin verspricht man sich durch eine zunehmende Verlinkung der Datenquellen signifikante Vorteile in Bezug auf deren semantische Integration, die insbesondere für Webdaten mit traditionellen Ansätzen (Data Warehouses, Mediator/ Wrapper-Architekturen) nur mit hohem Realisierungsaufwand und geringer Skalierbarkeit realisierbar ist [30, 20]. Linked Data verspricht effektive Datenintegrationsmöglichkeiten u.a. durch die generische Repräsentation und Verwaltung heterogener Daten, die semantische Beschreibung der Daten mittels Ontologien sowie die explizite semantische Verlinkung von Objekten verschiedener Quellen, insbesondere durch `sameAs`-Äquivalenzbeziehungen. Solche Links ermöglichen die Zusammenführung von sich entsprechenden Informationen, die zur Anreicherung und Verknüpfung mit weiteren Informationen nutzbar ist. Somit könnten etwa zum Autor einer Publikation weitere Informationen gewonnen werden, z.B. zu seinem Institut, seinen Kollegen etc. Über Produktverlinkungen können Preise unterschiedlicher Anbieter sowie Produktbeschreibungen und Nutzerbewertungen aus verschiedenen Quellen aggregiert und ausgewertet werden.

Bisherige Arbeiten im Bereich Linked Data fokussierten zunächst auf die Basisaufgaben der Generierung bzw. Extraktion von RDF-Daten sowie die effiziente Speicherung und Nutzung einzelner Datenquellen. Abbildung 1 stellt verschiedene Phasen des Lebenszyclus von Linked Data im Web dar [3], die wir im Folgenden detailierter beschreiben.

(a) Extraktion. In vielen Fällen werden RDF-Daten nicht manuell erstellt, sondern aus bestehenden Quellen extra-

Sören Auer (✉) · Jens Lehmann · Axel-Cyrille Ngonga Ngomo · Claus Stadler · Jörg Unbehauen
Agile Knowledge Engineering and Semantic Web (AKSW)
Universität Leipzig, Institut für Informatik, BIS
04109 Leipzig, Deutschland
Tel.: +49 341 97-32367
Fax: +49 341 97–32329
E-Mail: auer@informatik.uni-leipzig.de

Jens Lehmann
E-Mail: lehmann@informatik.uni-leipzig.de

Axel-Cyrille Ngonga Ngomo
E-Mail: ngonga@informatik.uni-leipzig.de

Claus Stadler
E-Mail: cstadler@informatik.uni-leipzig.de

Jörg Unbehauen
E-Mail: unbehauen@informatik.uni-leipzig.de

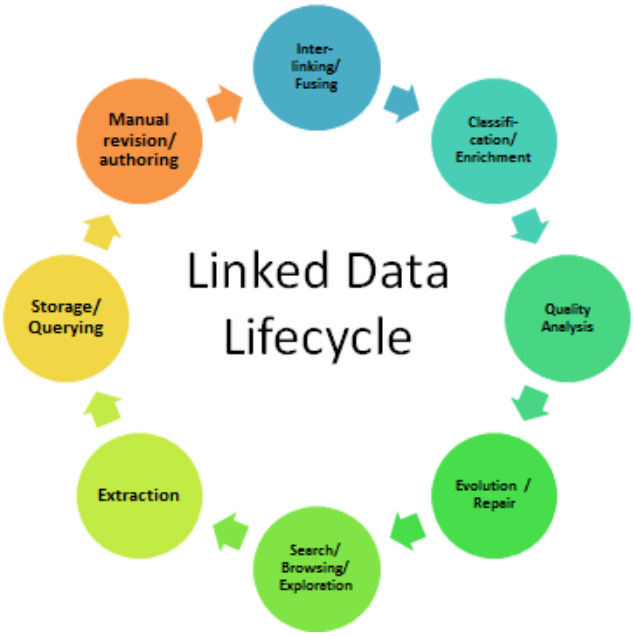


Abb. 1: Phasen des Lebenszyclus von Linked Data im Web.

hiert. Extraktions-Ansätze werden in der Regel entsprechend des Strukturierungsgrades der zugrundeliegenden Datenquellen klassifiziert: unstrukturierte, semi-strukturierte und strukturierte Quellen. Die Extraktion von Fakten aus den Klartext umfassenden unstrukturierten Quellen erfordern Methoden des Natural Language Processing. Mit dem *Natural Language Interchange Format* (NIF) wurde dafür eine Interoperabilitäts-Schnittstelle entworfen, die Linked-Data-Konzepte zur Referenzierung und Annotation von Texten und Textteilen nutzt [13]. Semi-strukturierte Quellen, z.B. Wiki-Markup, können mit anwendungsspezifischen Extraktionsmethoden, wie dem DBpedia-Extraktionsframework [18], erschlossen werden. DBpedia nutzt eine Reihe von semi-strukturierten Elementen in Wikipedia (wie z.B. Infoboxen, Kategorien und Links zwischen verschiedenen Spracheditionen) zur Extraktion und Repräsentation von Faktenwissen in RDF. Durch die Abdeckung einer Vielzahl von Informationsdomänen, die Verfügbarkeit in verschiedenen Sprachen und den Gemeinschafts-Konsens, den Wikipedia/DBpedia widerspiegelt, hat sich DBpedia zu einem zentralen Knotenpunkt im Linked Data Web entwickelt, der im Moment mehr als eine Milliarde Fakten umfasst. Eine Vielzahl von Forschungsansätzen [27] und kommerziellen Anwendungen (z.B. Zemanta.com, OpenCalais.com) nutzen DBpedia inzwischen als Basistechnologie. Strukturierte Quellen umfassen insbesondere relationale Datenbanken, für welche bereits mehrere Ansätze zum RDF-Mapping existieren (insbesondere *Triplify/Sparqlify* [4,37], *D2RQ* [6]), deren Standardisierung Ziel der W3C RDB2RDF Arbeitsgruppe[1] ist.

(b) Speicherung / Abfragen. Das starke Wachstum von Linked-Data-Quellen lieferte wichtige Impulse zur Entwicklung von Datenbank-Techniken für die native Verwaltung von RDF-Daten. Wenngleich relationale Datenbanken bei vergleichbaren Datenstrukturen noch eine wesentlich höhere Leistung als native RDF-Datenbanken aufweisen, konnte auch deren Skalierbarkeit durch die Entwicklung effizienter Speicher- und Clustering-Lösungen deutlich verbessert werden. Native RDF-Datenbanken eignen sich vor allem in Situationen mit sehr heterogenen Daten, die nicht leicht in einem relationalen Schema erfasst werden können.

(c) Klassifizierung / Anreicherung. Aufgrund der noch oft schwach ausgeprägten oder fehlenden Strukturinformationen bzw. Ontologiebezüge in LOD-Quellen ist deren Anreicherung um solche Strukturen für viele Anwendungen wesentlich, insbesondere zur Datenintegration und für fortgeschrittene Ansätze zur Link Discovery. Da sowohl OWL als auch RDF auf Beschreibungslogiken beruhen, kommen prinzipiell korrekt und vollständig eingestufte Reasoner-Implementierungen (z.B. ElVira, Pellet, FACT++, HermiT, KAON und Racer) zur Wissensanreicherung in Frage. Allerdings machen deren Leistungsanforderungen sie für die Anwendung auf für Wissensbasen der Größenordnung der LOD-Datenquellen ohne Anpassungen und Preprocessing unbrauchbar. Deshalb wurden mehrere unvollständige Inferenz-Methoden entwickelt, u.a. innerhalb des LarKC-Projekts [8]. Anreicherungs-Methoden basieren unter anderem auf induktiver logischer Programmierung und formaler Konzeptanalyse. Vielversprechende Ansätze in diesem Bereich sind ORE [19], DL-Learner [17] und OntoComp [33].

(f) Qualitätsanalyse. Linked Data Anbieter im Web unterstützen die Repräsentation von Wissen auf verschiedenen Abstraktions-Ebenen, verschiedene Ansichten und unterschiedliche Absichten. Informationen können falsch, tendenziös, widersprüchlich oder veraltet sein. Für die Filterung und Bewertung von Informationen aus verschiedenen Quellen wurden daher verschiedene Metriken entwickelt, die z.B. Inhalt, Kontext, Reputation und Bewertung berücksichtigen. Von besonderer Bedeutung und aktuelles Forschungsthema ist die Bewertung ob bestimmte Daten für einen bestimmten Anwendungsfall geeignet sind.

(d) Evolution und Reparatur. Es gibt eine Vielzahl von wissenschaftlichen Arbeiten im Bereich Ontologie-Evolution, die sich zunächst vor allem auf Schema-Evolution konzentrierten, zunehmend jedoch auch die Daten-Evolution mit leichtgewichtigeren Evolutions-Mustern unterstützen. Die

[1] `http://www.w3.org/2001/sw/rdb2rdf/`

effiziente und effektive Unterstützung der verteilten Evolution von Linked Data im Web ist ebenfalls Gegenstand aktueller Forschung.

In den letzten Jahren hat sich der Grad der Automatisierung in den einzelnen Phasen als Ergebnis der Grundlagenforschung und reifer Software-Implementierungen erhöht. Mit der DBpedia Wissensextraktion aus Wikipedia (Abschnitt 2), dem skalierbaren Linking von Wissensbasen (Abschnitt 3) und dem Mapping relationaler Daten nach RDF (Abschnitt 4) stellen wir in den folgenden Abschnitten drei Ansätze vor, die zentrale Bausteine des Web der Daten ausmachen.

2 DBpedia

Das Ziel des DBpedia-Projektes (siehe [2, 18, 22]) ist es Wissen aus Wikipedia, der größten Web-Enzyklopädie, zu extrahieren und es mit Hilfe von semantischen Technologien frei zur Verfügung zu stellen. Wikipedia ist eine der populärsten Webseiten überhaupt[2] mit vielen Millionen Nutzern. Allein innerhalb eines Monats editieren über 100 000 verschiedene Nutzer Artikel[3]. Seit der Gründung von Wikipedia im Jahr 2001 ist es zu einer der am weitesten verbreiteten Wissensbasen weltweit angewachsen und ist ein Musterbeispiel für die gemeinschaftliche Erstellung von Inhalten.

Trotz aller Vorteile von Wikipedia bietet es nur begrenzte Möglichkeiten für Abfragen und Suche an und ist nicht gut mit anderen Wissensbasen verknüpft. Zum Beispiel ist es schwierig mit Hilfe von Wikipedia Bürgermeister zu finden, die zu einer bestimmten Partei gehören und Kleinstädte ($<$ 50.000 Einwohner) verwalten. Die Information um solche und unzählige weitere Abfragen zu beantworten sind in Wikipedia vorhanden, aber über viele Artikel verstreut und nicht einheitlich zugreifbar[4]. Das DBpedia-Projekt erlaubt durch Extraktion von Informationen aus Wikipedia genau solche Abfragen.

Durch die Unterstützung und Integration von DBpedia in alle Phasen des Linked Data Lebenszyklus erlangte das Projekt große Bedeutung in Forschung und Praxis. Es wurde zu einem der Stützpfeiler der Linked Open Data Initiative und ein wichtiger Knotenpunkt zur Verknüpfung mit zahlreichen weitere Wissensbasen. DBpedia unterstützt inzwischen über 100 verschiedene Sprachen und extrahierte in seiner aktuellen Version 3.8 mehr als 1,8 Milliarden RDF-Tripel.

DBpedia arbeitet mit einer Reihe von Extraktoren um aus vielen Bereichen eines Wikipedia-Artikels Informationen zu extrahieren, z.B. Geo-Daten, Personen-

```
{{Infobox settlement
 | official_name     = Algarve
 | settlement_type   = Region
 | image_map         = LocalRegiaoAlgarve.svg
 | mapsize           = 180px
 | map_caption       = Map showing Algarve
                       Region in Portugal
 | subdivision_type  = [[Countries of the
                       world|Country]]
 | subdivision_name  = {{POR}}
 | subdivision_type3 = Capital city
 | subdivision_name3 = [[Faro, Portugal|Faro]]
 | area_total_km2    = 5412
 | population_total  = 410000
 | timezone          = [[Western European
                       Time|WET]]
 | utc_offset        = +0
 | timezone_DST      = [[Western European
                       Summer Time|WEST]]
 | utc_offset_DST    = +1
 | blank_name_sec1   = [[NUTS]] code
 | blank_info_sec1   = PT15
 | blank_name_sec2   = [[GDP]] per capita
 | blank_info_sec2   = Euro 19,200 (2006)
}}
```

Abb. 2: MediaWiki infobox Syntax für Algarve (links) und generierte Wikipedia Infobox (rechts) aus [22].

Daten, Bilder, Verknüpfungen zu anderen Sprachversionen, Disambiguierungs-Informationen und vieles mehr. Der Kern von DBpedia besteht jedoch in der Extraktion von Infoboxen in Wikipedia, die meistens rechts oben in Artikeln dargestellt werden. Deren Extraktion wurde erstmals in [5] beschrieben. In Abbildung 2 ist ein Beispiel einer Infobox dargestellt, einschließlich des Wiki-Markups, welches zu dieser Infobox gehört. Infoboxen fassen wichtige Fakten zum betrachteten Objekt zusammen und eigenen sich deshalb hervorragend zur Extraktion von Fakten. Nachteilig ist jedoch, dass sich das Infobox-System in Wikipedia über Jahre hinweg ohne zentrale Koordination entwickelt hat. Dadurch werden viele verschieden Infobox-Typen für gleiche oder ähnliche Arten von Objekten verwendet, zum Beispiel `infobox_city_japan`, `infobox_swiss_town` als auch `infobox_town_de`. Innerhalb der Infoboxen gibt es verschiedene Möglichkeiten Attribute zu spezifizieren, so gibt es Dutzende Varianten um das Geburtsdatum einer Person festzulegen, beispielsweise die Attribute `birthplace` und `placeofbirth`. Ähnliche Probleme betreffen die Datentypen. Ein Herzstück von DBpedia besteht deshalb aus Abbildungen (Mappings) von Infoboxen-Typen, -Attributen und -Datentypen auf eine zentrale DBpedia-Ontologie. Mappings für über 2.500 Infobox-Typen werden derzeit von der DBpedia-Gemeinschaft gepflegt[5]. Diese Mappings erlauben die Übersetzung von Infoboxen in RDF-Tripel, sowie die Typisierung von Resourcen und die Integration über mehrere Wikipedia-Sprachversionen hinweg.

[2] Siehe `http://www.alexa.com/topsites`.

[3] `http://en.wikipedia.org/wiki/Special:Statistics`

[4] Für wichtige Abfragen erstellen Wikipedia-Nutzer eigene Listen, aber das deckt nur populäre Abfragen ab und solche Listen müssen manuell verwaltet werden.

[5] Siehe `http://mappings.dbpedia.org`

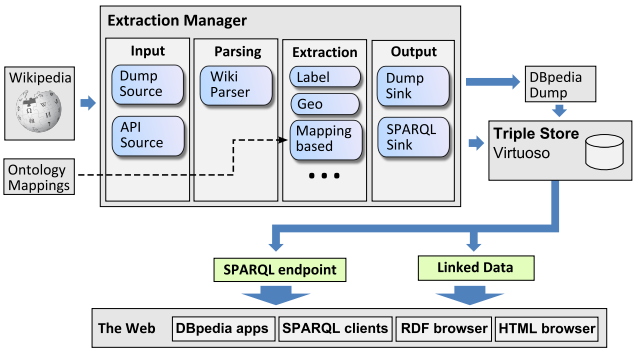


Abb. 3: Illustration der DBpedia-Architektur.

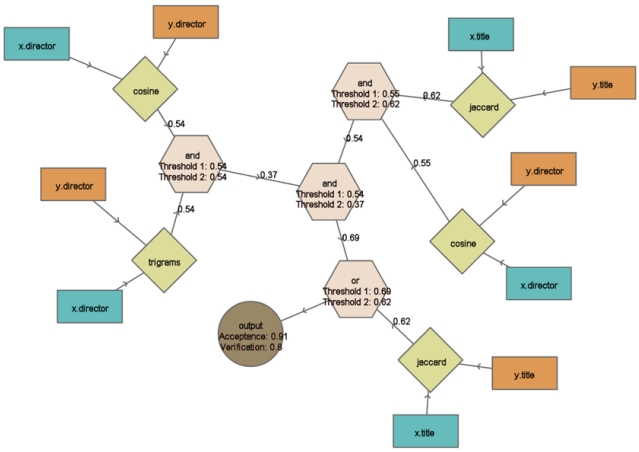


Abb. 4: Grafische Darstellung einer Link Spezifikation zum Verknüpfen von Filmen in LinkedMDB und DBpedia.

Die Gesamtarchitektur des Extraktionsframeworks ist in Abbildung 3 dargestellt. Wikipedia-Artikel werden entweder über Datenbankdumps oder einen Datenstrom aktueller Änderungen eingelesen und zur weiteren Bearbeitung abgelegt. Das Extraktionsframework lädt diese Seiten und führt zahlreiche Extraktoren auf deren Inhalt, d.h. dem Wiki Mark-Up, aus. Dabei können die Extraktoren auf verschiedene Parser zur Unterstützung von unterschiedlichen Datentypen zurückgreifen. Jeder Extraktor liefert eine Menge von RDF-Tripeln. Diese können dann entweder in Dateien geschrieben oder in einen Triple Store eingelesen werden. Basierend auf dem Triple Store können die Daten anschließend abgefragt und als Linked Data publiziert werden.

Die resultierenden Daten wurden bereits in zahlreichen Kontexten verwenden, zum Beispiel im DeepQA-Projekt, welches im IBM Watson System resultierte. Dieses System erzielt in *Jeopardy* bessere Ergebnisse als menschliche Spieler. Für die Daten wurden weiterhin facettenbasierte Browser entwickelt [10], die eine einfache Navigation und die Beantwortung einfacher Abfragen erlauben. Für DBpedia wurden zahlreiche weitere Visualisierungswerkzeuge wie gFacet [11] und RelFinder [12]. Die extrahierten Daten werden darüber hinaus in zahlreichen Unternehmen, z.B. Zemanta, Neofonie, BBC und Thomson Reuters eingesetzt.

3 Linking

Das vierte Linked Data Grundprinzip ist die Bereitstellung von Links zwischen Wissensbasen. Solche Links sind von genauso zentraler Bedeutung für das Linked Data Web wie Hyperlinks für das heutige Internet und erlauben die integrierte Verwendung von Daten aus unterschiedlichen Datenquellen für Zwecke wie die Beantwortung komplexer Fragen [34], Datenintegration [23] und die Abarbeitung von Wissensbasis-übergreifende (engl. *federated*) SPARQL Anfragen [31]. In diesem Abschnitt gehen wir zunächst auf zwei der Herausforderungen ein, die bei der Generierung von Links zwischen Wissensbasen auftreten. Wir gehen anschließend auf existierende Lösungen für beide Probleme ein.

3.1 Komplexität und Linking

Formal betrachtet zielt Linking (auch Link Discovery genannt) darauf ab, Verknüpfungen zwischen Elementen zweier Mengen S und T von RDF Ressourcen zu generieren. Diese Mengen könnten zum Beispiel Filme aus der Linked Movie Database (LinkedMDB) und Filme aus DBpedia enthalten. Die Generierung der Links erfolgt indem berechnet wird, welche Paare (s,t) von Elementen aus $S \times T$ gewissen Ähnlichkeitskriterien genügen. Die formale Beschreibung dieser Mengen und Kriterien wird *Link Spezifikation* genannt. In unserem Beispiel könnten die Namen der Filme sowie die Namen ihrer Direktoren miteinander verglichen werden. Naive Ansätzen würden nun die Eigenschaften jedes Films aus LinkedMDB mit denen von jedem Film aus DBpedia vergleichen. Da jede dieser Wissensbasen ca. 85 Tsd. Filme umfasst, würden naive Ansätze ca. 14 Mrd. Ähnlichkeitsberechnungen durchführen müssen, um diese zwei Datensätze miteinander zu verknüpfen. Folglich würde die Berechnung von Links zwischen diesen im Vergleich zu üblichen Mengen von Ressourcen kleinen Datensätzen ca. ein halbes Jahr dauern, wenn 1ms pro Berechnung erforderlich wäre. Die Entwicklung von Vergleichsalgorithmen, welche die *Laufzeitkomplexität* von Linking reduzieren, ist die erste Herausforderung in diesem Forschungsgebiet.

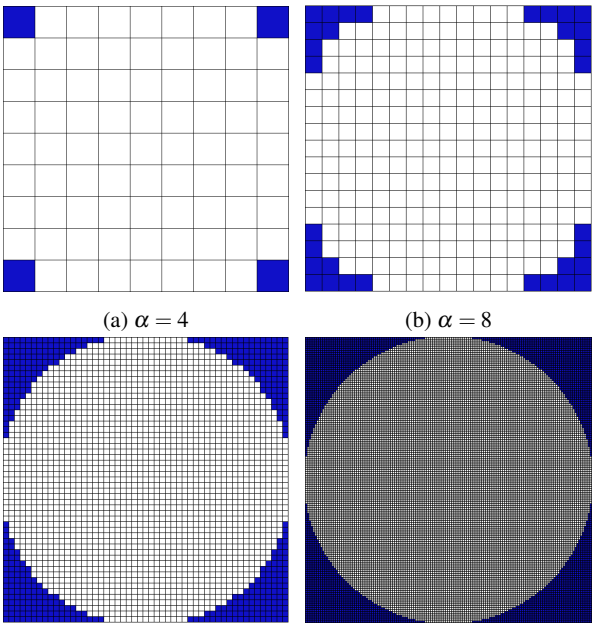
Die zweite Herausforderung ist die *Erstellung von hochwertigen Link Spezifikation* für die zu berechnende Menge an Links. Dabei gilt eine Spezifikation als hochwertig, wenn ihre Ausgabe möglichst genau (Precision) und vollständig (Recall). Abbildung 4 zeigt ein Beispiel einer hochwertigen Link Spezifikation für das deklarative Framework LIMES [26, 24, 25]. Diese Link Spezifikation nutzt

zwei Eigenschaften von Filmen aus LinkedMDB und DBpedia zum Linking. Die Auswahl der zu verwendenden Eigenschaften erfordert explizites Wissen über die Struktur der zu verknüpfenden Datenquellen. Zudem ist die korrekte Einstellung der Spezifikationsparameter für die Generierung korrekter Links keine triviale Aufgabe. Hier werden maschinelle Lernverfahren erforderlich, welche eine (semi-)automatische Auswahl in der Spezifikation genutzten Eigenschaften und der Spezifikationsparameter ermöglichen.

3.2 Zeiteffizientes Linking

Die Grundidee hinter zeiteffizientem Linking ist die Reduktion der Anzahl von durchgeführten Vergleichen. Dies wird meistens durch eine Partitionierung des Problems, den Einsatz von zeiteffizienten Approximationen oder eine Kombination der Partitionierung und Approximationen erreicht. Einer der ersten Ansätzen für die effiziente Berechnung von Links ist der in [26] vorgestellte LIMES Ansatz. Das Linking Problem wird hier auf die Berechnung der Distanz zwischen allen Paaren von Elementen aus zwei Mengen von Punkten in einem metrischen Raum reduziert. LIMES verwendet die in metrischen Räumen geltende Dreiecksungleichung um pessimistische Approximationen von Distanzen zeiteffizient zu berechnen. Dazu wird zunächst eine Partition des metrischen Raums durch Exemplare durchgeführt. Die Verteilung der Exemplare erfolgt indem eine festgelegte Anzahl von möglichst weit von einander entfernten Punkten aus der Menge S von Ressourcen berechnet wird. Jedes Exemplar gilt dann als Repräsentant für die Punkte in seiner Umgebung. Aus der Distanz zwischen jedem Exemplar und den Punkten in seiner Umgebung kann die Distanz zwischen allen Punkten aus S und allen Punkten aus T approximiert werden. Die genaue Distanz zwischen zwei Punkten s resp. t aus S resp. T wird nur dann berechnet wenn die Approximation dieser Distanz fehlschlägt, d.h., wenn keine Verletzung der von der Spezifikation geforderten maximalen Distanz zwischen s und t aus der Approximation heraus festgestellt werden kann. Eine Evaluation dieses Ansatzes zeigt, dass er mehr als 80% der unnötigen Distanzberechnungen in metrischen Räumen verwerfen kann.

Der später im gleichen Framework implementierte $\mathscr{HR}^3$-Algorithmus [24] nutzt genauso die Eigenschaften von Metriken und zielt darauf ab, einer Partitionierung des Raumes zu erzielen. Hier werden jedoch keine Exemplare verwendet. Stattdessen wird eine Diskretisierung des Raumes durch Hyperwürfel durchgeführt. Anhand dieser Diskretisierung wird anschließend errechnet, welche Teilmengen von S und T miteinander zu vergleichen sind. Dabei wird ausgenutzt, dass Spezifikationen in metrischen Räumen Hyperkugeln beschreiben. Ziel der Diskretisierung ist es dementsprechend, eine Hyperkugel möglichst genau zu approximieren. Die Genauigkeit der Approximation kann



(a) $\alpha = 4$ (b) $\alpha = 8$

(c) $\alpha = 25$ (d) $\alpha = 75$

Abb. 5: Approximation von $\mathscr{HR}^3$ für verschiedene Werte von α in einem zwei-dimensionalen euklidischen Raum. Mit steigendem α steigt die Genauigkeit der Approximation aber auch die Anzahl der Hyperwürfel.

durch den Granularitäts-Parameters α von $\mathscr{HR}^3$ festgelegt werden. Eine Erhöhung von α führt dazu, dass die Anzahl der unnötigen Distanzberechnungen reduziert wird (siehe Abbildung 5) aber auch dazu, dass die Anzahl der Hyperwürfel (und somit die Größe des Index von $\mathscr{HR}^3$) wächst. Der Hauptvorteil von $\mathscr{HR}^3$ liegt insbesondere darin, dass er reduction-ratio-optimal ist, d.h., dass die Anzahl der unnötigen Distanzberechnungen gegen 0 strebt wenn α gegen $+\infty$ strebt.

Andere Ansätze basieren auf multi-dimensionalem Blocking (siehe z.B. [15]). Hier ist die Idee, dass die gesamte Spezifikation in einem (nicht notwendigerweise metrischen) Raum abgebildet wird. Dieser Raum wird dann in überlappende Blöcke geteilt. Alle Berechnungen werden ausschließlich innerhalb der vorberechneten Blöcke durchgeführt. Auch dieser Ansatz ist in der Lage, eine Vielzahl unnötiger Berechnungen zu verwerfen. Die zeiteffiziente Berechnung von Links garantiert jedoch nicht, dass die Ergebnisse vollständig oder gar genau sind. Zu diesem Zwecke werden maschinelle Lernverfahren eingesetzt.

3.3 Genaues und vollständiges Linking

Die Genauigkeit und Vollständigkeit (Recall) einer Spezifikation sind von entscheidender Bedeutung für ihren Einsatz bei der Verknüpfung von Wissensbasen im Linked Da-

ta Web. Die meisten Verfahren, die darauf abzielen, Spezifikationen mit einer hohen Precision und einem hohen Recall zu errechnen, basieren auf maschinellem Lernen. Die Grundidee hinter diesen Verfahren besteht darin, die bestmögliche Kombination von atomaren Metriken, Gewichten und Schwellwerten zu finden, welche zu maximalen Precision und Recall führt. Formal ist dies dazu äquivalent, eine Klassifikationsfunktion in einem vorgegebenen Raum zu errechen. Zu diesem Zweck werden überwachte [27,28, 14] und neuerdings auch unüberwachte Lernverfahren [29] verwendet.

Einfache überwachte Lernverfahren [14] erwarten eine Menge von positiven und negativen Beispielen für die zu berechnenden Links. Sie versuchen dann anhand dieser Beispiele Spezifikationen zu errechnen, welche möglichst alle positiven und keins der negativen Beispiele erzeugen. Das Hauptproblem dieser Ansätze ist, dass sie große Mengen positiver und negativer Beispiele benötigen, um hochwertige Linkspezifikationen zu errechnen. Eine Lösung für dieses Problem liegt in der Verwendung von neugierigen Algorithmen (engl.: curious classifiers) [27,28]. Diese sind in der Lage, vom Nutzer zu erfragen, ob ein Link als positives oder negatives Beispiel zu verwenden ist. Durch die selbständige Auswahl von möglichst informativen Links benötigen solche Klassifikationsverfahren wesentlich weniger positive und negative Beispiele, um gleichwertige Spezifikationen zu errechnen. Der Vorteil der oben genannten Ansätze ist, dass sie für die Berechnung aller Arten von Links eingesetzt werden können.

In neuen Arbeiten wurden die oben vorgestellten Paradigmen erweitern. Zum Beispiel wurde gezeigt, dass unüberwachte Verfahren im Falle der Berechnung von `owl:sameAs` Links verwendet werden können [29]. Hier wird eine Approximation der Precision und des Recalls von Spezifikationen ohne Nutzerfeedback errechnet. Diese Approximation wird sukzessiv optimiert bis sie einen maximalen Wert erreicht hat oder festgelegte Anzahl von Iterationen durchlaufen wurde. Manche Linking-Ansätze verabschieden sich ebenso vom Lernen in einem vorgegebenen Raum [35] und versuchen den Raum so anzupassen, dass einfache Klassifikationsfunktionen zeiteffizient gefunden werden können. Während die ersten Ergebnisse in diesem Forschungsbereich vielversprechend sind, wirft die Transformation von Räumen neue Effizienzfragen auf, da existierende Verfahren nur teilweise in solchen modifizierten Räumen eingesetzt werden können.

4 RDF-Mapping relationaler Daten

In diesen Abschnitt erläutern wir den Ansatz des Mappings von relationalen Daten nach RDF (RDB2RDF) als einen integralen Bestandteil einer Linked Data basierten Integration. Zudem werden verschiedene Ansätze vorgestellt und aktuelle Herausforderungen am Beispiel von LinkedGeoData erläutert.

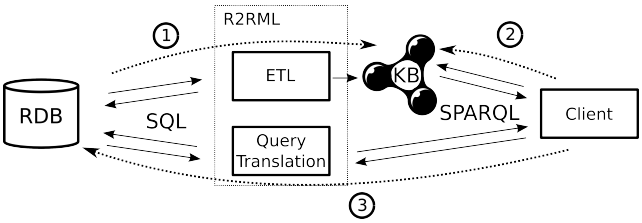


Abb. 6: Einsatzszenarien von RDB2RDF Mappings. RDF Daten werden mittels Mapping aus einer relationalen Datenbank erstellt (1) und abgefragt (2). Alternative wird die SPARQL Anfrage in SQL übersetzt und das Anfrageergebnis rückübersetzt (3).

4.1 Motivation und Szenario

Das relational Datenmodell ist der vorherrschende Ansatz strukturierte Daten zu speichern, sowohl im Unternehmen als in Webanwendungen. Für den Einsatz eines relationalen Datenbankmanagementsystems anstelle eines Triple Stores sprechen, abhängig vom Einsatzszenario, oft folgende Gründe. Zum einen ist die Ausführungsgeschwindigkeit von Abfragen bei relationalen Datenbanken durch die effizientere Indizierung höher. Zudem bieten relationale Datenbanken Funktionen wie Transaktionen oder Integritätsbedingungen. Darüber hinaus verwenden die meisten Anwendungen, sei es im Unternehmen oder im Web, eine oder mehrere relationale Datenbanken zur Datenhaltung; eine Migration zu einem Triple Store wäre aufwendig. Daher ist das Ziel des Mappings relationaler Datenbanken diese als Linked Data bereit zu stellen und in den Lebenszyklus zu integrieren. RDB2RDF Mapping-Werkzeuge fungieren daher als ein Bindeglied zwischen relationalen Datenbanken und dem Data Web. Zahlreiche große, offene Datensammlungen wie *OpenStreetMap*[6] oder *MusicBrainz*[7] konnten mit Hilfe von RDB2RDF Mapping-Werkzeugen wie Triplify [4] mit einer RDF-Schnittstelle versehen werden und dadurch in das Linked-Data-Web integriert werden.

Beim RDB2RDF-Mapping können zwei Ansätze unterschieden werden die in Abbildung 6 dargestellt sind. Zum einen kann mit Hilfe eines Mappings RDF materialisiert werden und beispielsweise in einen Triple Store geladen werden (1). Diese materialisierten Daten können dann mittels SPARQL abgefragt werden (2), als Linked Data bereit gestellt werden und mit anderen Datenquellen verknüpft werden. Der zweite Ansatz ist die direkte Übersetzung von SPARQL Abfragen in SQL. Diese Szenario benötigt kei-

[6] `http://www.openstreetmap.org/`
[7] `http://musicbrainz.org/`

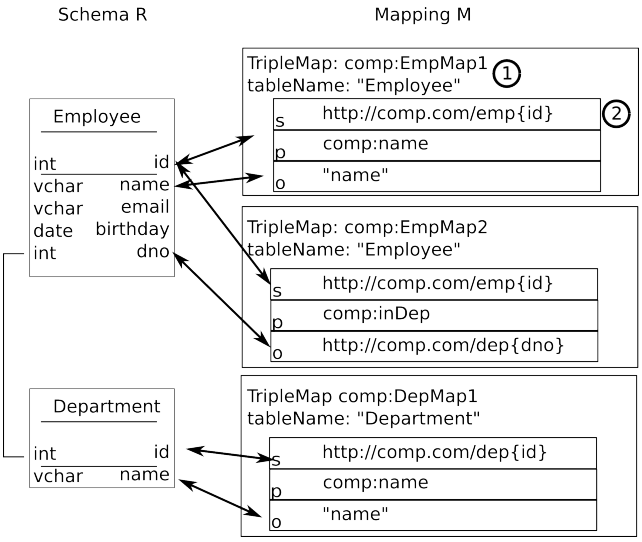


Abb. 7: Beispielhaftes Mapping der Tabellen “employee” und “department”.

nen zusätzlichen Triple Store und die Daten werden nicht dupliziert. Ebenso ist eine Nutzung der bestehenden, effizienten Indexstrukturen des Datenbankmanagementsystems möglich.

4.2 Mappings und Ansätze

Mit der Standardisierung der RDB to RDF Mapping Language (R2RML)[8] wurde im September 2012 eine Basis für Interoperabilität zwischen den zahlreichen RDB2RDF Mapping Tools gelegt. Beispiele für diese Mapping Tools sind etablierte Werkzeuge wie Triplify, D2RQ [6] oder Virtuoso RDF Views[9], aber auch neuere Ansätze wie Ultrawrap [32], SparqlMap [37] oder Sparqlify[10]. Während Triplify direkt RDF aus der Datenbank erzeugt und dieses als Linked Data oder als materialisierten Graphen zur Verfügung stellt, sind die anderen Tools SPARQL-to-SQL Rewriter und basieren auf einem virtuellen Graph. Die dieser Transformation zu Grunde liegenden Mappings bedienen sich trotz syntaktischer Unterschiede des selben Prinzips. Beispielhaft ist solch ein Mapping in einer an R2RML angelehnten Struktur in Abbildung 7 dargestellt. Die TripleMap (1) verknüpft die Tupel einer Datenbanksicht mit TermMaps. TermMaps erzeugen RDF Terme (IRIs, Blank Nodes oder Literale) aus den Tupeln oder aus konstanten Werten. In der in Abbildung 7 dargestellte TermMap (2) ist die Bildungsvorschrift für eine IRI aus der *id* Spalte der *employee*-Sicht definiert, die zur Erzeugung des Subjekt-RDF-Terms verwendet wird.

[8] http://www.w3.org/TR/r2rml/

[9] http://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html

[10] http://sparqlify.org

Eine Materialisierung des Graphen ist möglich in dem für alle TripleMaps alle verknüpften Datenbanksichten dementsprechend ausgewertet werden. Durch die in [1] bewiesene Gleichheit der Ausdrucksstärke von SQL und SPARQL ist es möglich beliebige SPARQL Anfragen in äquivalente SQL Anfragen umzusetzen. Die effiziente Implementierung dieses Vorgangs ist Gegenstand aktueller Forschung. Ein Beispiel für eine solche Umsetzung wird im nachfolgenden Abschnitt vorgestellt.

4.3 Mapping großer Datenbanken am Beispiel von LinkedGeoData

Im Rahmen des *LinkedGeoData*[11] (LGD) Projekts werden *OpenStreetMap*[12] (OSM) Daten nach RDF konvertiert. Ziel von OSM ist die Erstellung einer freien Karte der Welt, wobei die Modellierung über drei Entitäten geschieht, welche in folgender Häufigkeit bei OSM auftreten:

- *Knoten*: Punktobjekte mit Latitude und Longitude, ca. 1.78 Mrd. Vorkommen.
- *Wege*: Geordnete Sequenz an Knotenreferenzen, ca. 171 Mio. Vorkommen.
- *Relationen*: Jede Entität kann als Mitglied mit einer bestimmten Rolle in einer Relation auftreten, ca. 1.8 Mio. Vorkommen.

Allen Entitäten ist gemein, dass sie durch eine Menge von Schlüssel-Wert Paaren, genannt *Tags*, ausgezeichnet werden können. Die daraus in LGD generierten RDF Datensätze sind als Linked Data, Downloads und per SPARQL Endpunkte frei zugänglich. Zu dessen Erzeugung wird ein SPARQL-to-SQL Rewriter eingesetzt, da dadurch Werkzeuge der OSM Community zur Synchronisation mit relationalen Datenbanken wiederverwendet werden können. Die Architektur von LGD ist in Abbildung 7 dargestellt, und wird im Folgenden kurz erläutert.

OSM bietet die Daten des gesamten Planeten zum Download an[13]. Des weiteren werden sogenannte *Changeset* Dateien veröffentlicht, welche zeitliche Änderungen an den Daten enthalten und in Minuten-, Stunden-, Tages- und Wochenintervallen generiert werden. Mittels der Software *Osmosis*[14] können die Quelldaten in eine mit PostGIS[15] erweiterten PostgreSQL[16] Datenbank geladen, und in weiterer Folge mit den Changesets synchronisiert werden.

Bei der RDF Konvertierung von OSM wurde großer Wert darauf gelegt, keine Schemamodifikationen vorzunehmen, welche die Kompatibilität mit den OSM Tools

[11] http://linkedgeodata.org

[12] http://openstreetmap.org

[13] http://planet.openstreetmap.org

[14] http://wiki.openstreetmap.org/wiki/Osmosis

[15] http://postgis.net/

[16] http://www.postgresql.org/

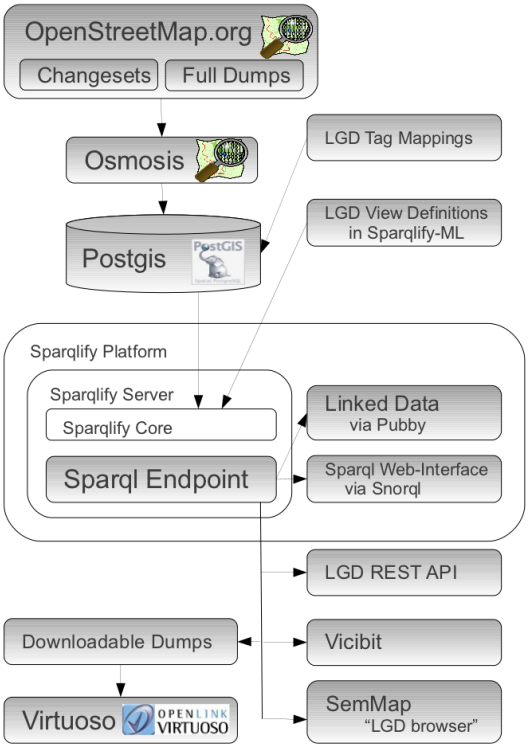


Abb. 8: Architektur des LinkedGeoData Projekts.

beeinträchtigen, vor allem mit Hinblick auf das Synchronisationswerkzeug. Zunächst wurde die OSM Datenbank mit Hilfstabellen (*Tag Mappings*) angereichert, welche beschreiben, wie die Tags nach RDF zu überführen sind. Schließlich wurden manuell RDB-RDF Mappings angelegt, um aus den OSM Daten in Kombination mit den Tag Mappings die RDF Daten zu generieren. Für LGD kommt dabei der SPARQL-to-SQL Rewriter *Sparqlify* zum Einsatz, welcher basierend auf RDB-RDF Mappings in Form von Sichtdefinitionen einen SPARQL Endpoint und ein Linked Data Interface zur Verfügung stellt. Aufgrund der großen Datenmengen kommen bei LGD zwei Instanzen von Sparqlify zum Einsatz: Eine öffentliche Instanz, mit begrenzten Anfragezeiten und Ergebnismengengrößen, und eine interne Instanz, ohne Restriktionen, zur Materialisierung der Graphen zwecks Erstellung der Downloads.

Es folgt ein Beispiel, wie OSM Daten anhand der Tabellen in Abbildung 9 nach RDF überführt werden.

Listing 1 zeigt das Anlegen eines Funktionsindex um effizienten Zugriff auf Tags mit ganzzahligen Werten zu gewährleisten und die Erstellung einer SQL Sicht, welche den Zugriff auf die kombinierten Daten vereinfacht. Die SQL Sicht hat die selben Spalten wie die Tabelle `node_tags`, allerdings ist die Spalte `v` nun ganzzahlig.

node_tags		
id	**k**	**v**
21068295	population	"3132463"

lgd_map_datatype	
k	**datatype**
population	int

Abb. 9: Eine Tabelle aus dem OSM Schema und eine Hilfstabelle aus LGD.

Die Sichtdefinition in Auflistung 2 beschreibt die Erzeugung von RDF Tripel. Dabei ist zu beachten, dass die Variablen auf der rechten Seite der Ausdrücke in der *WITH* Klausel sich auf die Spaltennamen der in der *FROM* Klausel angegebenen Tabelle beziehen.

```
CREATE INDEX idx_node_tags_k_int
 ON node_tags(k, lgd_tryparse_int(v))
 WHERE lgd_tryparse_int(v) IS NOT NULL;

CREATE VIEW lgd_node_tags_int AS
 SELECT a.node_id, a.k, lgd_tryparse_int(a.v) AS v
  FROM node_tags a JOIN lgd_map_datatype b ON a.k=b.k
  WHERE lgd_tryparse_int(a.v) IS NOT NULL AND b.
       datatype = 'int';
```

Listing 1: Erweiterungen des OSM Schemas

Die RDF Daten werden erzeugt, indem für jede Zeile der Tabelle die Ausdrücke ausgewertet werden, um die Variablen der Tripel Mustern in der *CONSTRUCT* Klausel zu belegen. Umgekehrt lassen sich SPARQL Filter-Ausdrücke ueber solche Sichtdefinitionen zu Bedingungen im SQL umschreiben, sodass intuitiv die Tripel nur für bestimmte Zeilen erzeugt werden.

```
Prefix xsd:<http://www.w3.org/2001/XMLSchema#>
Prefix lgd:<http://linkedgeodata.org/triplify/>
Create View lgd_node_tags_int As
Construct { ?s ?p ?o . }
With
 ?s = uri(concat(lgd:node, ?node_id))
 ?p = uri(concat('http://linkedgeodata.org/ont/',?k))
 ?o = typedLiteral(?v, xsd:int)
From lgd_node_tags_int
```

Listing 2: Sichtdefinition in der Sparqlify Mapping Sprache

```
 SELECT a.node_id, b.v AS v_int, NULL::bool AS v_bool
  FROM node a JOIN lgd_node_tags_int b
  WHERE a.id = b.node_id AND
   ST_INTERSECTS(a.geom, "POLYGON((...))"::geometry)
UNION ALL
 SELECT a_node_id, NULL::int AS v_int, b.v AS v_bool
  FROM node a JOIN lgd_node_tags_bool b
  WHERE a.id = b.node_id AND
   ST_INTERSECTS(a.geom, "POLYGON((...))"::geometry)
UNION ALL   ...
```

Listing 3: Generiertes SQL bei der Anfrage nach allen Tags von Knoten in einem bestimmten Bereich.

Zur Beantwortung von SPARQL Anfragen auf einer Menge an Sichtdefinitionen gibt es etliche Herausforderungen:

- *Kandidatensichtauswahl* Bei einer SPARQL Anfrage muss für jedes in ihr vorkommende Tripel eine passende Menge an Tripel in den Kandidatensichten gefunden werden. Ein naiver Ansatz swählt für jedes Anfrage-Tripel *alle* Sicht-Tripel, welches z.B. bei 5 Anfrage-Tripel über 10 Sichten mit je 4 Tripel $(10 \cdot 4)^5$ = 102.4 Mio. Kombinationen ergibt.
- Durch SPARQL-to-SQL Rewriter generierte SQL Anfragen bestehen meistens aus der Vereinigung mehrerer Unteranfragen (siehe Auflistung 3). Durch das Optimieren von gemeinsamen Teilausdrücken über die Unteranfragen (hier die Bereichsselektion) ließen sich Performanzverbessungen erzielen.
- PostgreSQL unterstützt keine Mehrspaltenindexstatistiken. Dies führt dazu, dass der Anfrageoptimierer in Anbetracht solcher Indexe die Ergebnismengen großzügig Unterschätzt und oftmals Fehlentscheidungen trifft.[17]
- Um effiziente Antworten auf SPARQL Anfragen zu ermöglichen, muss eventuell Aufwand betrieben werden um die Daten entsprechend aufzubereiten und indizieren. Im Falle von LGD war das mittels Funktionsindexe leicht möglich, dennoch hat dies Datenduplikation und Performanzeinbußen beim Schreiben zur Folge.

In Bezug auf allgemeine SPARQL Anfragen ist aufgrund der Herausforderungen bzgl. Geschwindigkeit die Materialisierung der RDF Daten und das Laden in einen Triple Store immer noch die zuverlässigste Variante. Jedoch bieten SPARQL-to-SQL Rewriter Vorteile bei der Datenaktualität, vorallem bei großen Datenbeständen: Hier können die RDF Daten eingesehen werden, noch bevor ein zeit- und platzintensiver Materialisierungsprozess angestoßen wurde. Linked Data Anfragen resultieren in der Regel in einfachen SQL Anfragen, welche direkt auf der Datenbank ausgeführt werden können, sofern die entsprechenden Indexe vorhanden sind. Während bei Triple Stores der Fokus eher auf dem Publizieren von Daten liegt, liegt der Fokus bei RDBMs eher beim Verwalten der Daten für konkrete Anwendungsfälle. Im Rahmen des semantischen Datenmanagements leisten SPARQL-to-SQL Rewriter einen wertvollen Beitrag zum Überbrücken beider Bereiche.

5 Zusammenfassung und Ausblick

In diesem Artikel haben wir einen kurzen Überblick über verschiedene Aspekte des Managements von Linked Data im Web gegeben. Mit DBpedia haben wir ein Wikipedia-Wissensextraktionsprojekt vorgestellt, welches sich inzwischen zu einem zentralen Verknüpfungs-Knotenpunk im Daten-Web entwickelt hat. Wir haben das Problem des Link-Discovery eingeführt und verschiedenen Ansätze zur Steigerung der Skalierbarkeit sowie von Genauigkeit und Vollständigkeit vorgestellt. Eine weitere zentrale Herausforderung ist die effiziente Anbindung relationaler Daten an das Linked Data Web mittels RDB2RDF-Mapping, welches wir im letzten Abschnitt gemeinsam mit dem Anwendungsfall LinkedGeoData vorgestellt haben.

Gemeinsame Herausforderungen in allen Phasen des Lebenszyklus sind die LOD-Daten Heterogenität, Skalierbarkeit sowie die synergetische, phasenübergreifende Integration von Lösungsansätzen. Einige Aspekte wir Datenqualität und Evolution, sind bislang noch nicht umfassend erforscht und reife Werkzeuge zu deren Unterstützung fehlen. Neben dem Einsatz von Linked Data zur Integration von Daten im Web ist besonders die Datenintegration in Intranets großer Organisationen und Unternehmen ein aussichtsreiches Anwendungsszenario. In Ergänzung zu dienst-orientierten Architekturen kann Linked Data die Vernetzung und Integration von Daten aus den hunderten (oder sogar tausenden) Informationssystemen im Intranet großer Unternehmen unterstützen. Ähnlich wie DBpedia einen Kristallisationspunkt im Daten-Web darstellt können Unternehmensthesauri diese Funktion in Unternehmens-Daten-Intranets übernehmen. RDB2RDF-Mapping und Werkzeuge zur Verlinkung spielt eine zentrale Rolle um existierende, auf relationalen Datenbanken basierende Informationssysteme an die entstehenden Daten-Intranets anzubinden und zu verknüpfen.

Danksagung Wir danken den Mitgliedern der Arbeitsgruppe AKSW und den Projektpartnern der EU RP7 Projekte LOD2 (GA no. 257943), GeoKnow (GA no. 318159) und BIG (GA no. 318062), welche die in diesem Artikel vorgestellten Arbeiten unterstützt haben.

[17] `http://wiki.postgresql.org/wiki/Cross_Columns_Stats`

Literatur

1. Angles, R., Gutierrez, C.: The expressive power of sparql. In: 7th Int. Conf. on The Semantic Web, ISWC '08, pp. 114–129. Springer (2008). DOI 10.1007/978-3-540-88564-1_8. URL `http://dx.doi.org/10.1007/978-3-540-88564-1_8`
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: 6th Int. Semantic Web Conf. (ISWC), *LNCS*, vol. 4825, pp. 722–735. Springer (2008). DOI doi:10.1007/978-3-540-76298-0_52
3. Auer, S., Bühmann, L., Lehmann, J., Hausenblas, M., Tramp, S., van Nuffelen, B., Mendes, P., Dirschl, C., Isele, R., Williams, H., Erling, O.: Managing the life-cycle of linked data with the LOD2 stack. In: Int. Semantic Web Conf. (ISWC 2012) (2012). URL `http://svn.aksw.org/lod2/Paper/ISWC2012-InUse_LOD2-Stack/public.pdf`
4. Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., Aumueller, D.: Triplify: Light-weight linked data publication from relational databases. In: 18th Int. Conf. on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009, pp. 621–630. ACM (2009). DOI doi:10.1145/1526709.1526793. URL `http://doi.acm.org/10.1145/1526709.1526793`
5. Auer, S., Lehmann, J.: What have Innsbruck and Leipzig in common? extracting semantics from wiki content. In: Proceedings of the ESWC (2007), *Lecture Notes in Computer Science*, vol. 4519, pp. 503–517. Springer, Berlin / Heidelberg (2007)

6. Bizer, C., Cyganiak, R.: D2r server – publishing relational databases on the semantic web. Poster at the 5th Int. Semantic Web Conf. (ISWC2006) (2006). URL http://www4.wiwiss.fu-berlin.de/bizer/pub/Bizer-Cyganiak-D2R-Server-ISWC2006.pdf
7. Chong, E.I., Das, S., Eadon, G., Srinivasan, J.: An efficient sql-based rdf querying scheme. In: VLDB, pp. 1216–1227. ACM (2005)
8. Fensel, D., van Harmelen, F., Andersson, B.: Towards LarKC: A platform for web-scale reasoning. In: ICSC, pp. 524–529. IEEE Computer Society (2008). URL http://doi.ieeecomputersociety.org/10.1109/ICSC.2008.41
9. Goel, K., Guha, R.V., Hansson, O.: Introducing rich snippets. Google Webmaster Central Blog (2009). URL http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html
10. Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., Scheel, U.: Faceted wikipedia search. In: Business Information Systems, 13th Int. Conf., BIS 2010, Berlin, Germany, May 3-5, 2010., *Lecture Notes in Business Information Processing*, vol. 47, pp. 1–11. Springer (2010). URL http://dx.doi.org/10.1007/978-3-642-12814-1
11. Heim, P., Ertl, T., Ziegler, J.: Facet graphs: Complex semantic querying made easy. In: 7th Extended Semantic Web Conf. (ESWC 2010), *LNCS*, vol. 6088, pp. 288–302. Springer (2010). URL http://dx.doi.org/10.1007/978-3-642-13486-9_20
12. Heim, P., Hellmann, S., Lehmann, J., Lohmann, S., Stegemann, T.: RelFinder: Revealing relationships in RDF knowledge bases. In: 3rd International Conf. on Semantic and Media Technologies (SAMT), *LNCS*, vol. 5887, pp. 182–187. Springer (2009)
13. Hellmann, S., Lehmann, J., Auer, S.: Linked-data aware uri schemes for referencing text fragments. In: EKAW 2012, LNCS 7603. Springer (2012). DOI doi:10.1007/978-3-642-16438-5_10. URL http://svn.aksw.org/papers/2012/NIF/EKAW_short_paper/public.pdf
14. Isele, R., Bizer, C.: Learning Linkage Rules using Genetic Programming. In: 6th International Ontology Matching Workshop (2011)
15. Isele, R., Jentzsch, A., Bizer, C.: Efficient Multidimensional Blocking for Link Discovery without losing Recall. In: WebDB (2011)
16. Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., Lee, R.: Media meets semantic web - how the bbc uses dbpedia and linked data to make connections. In: ESWC, *LNCS*, vol. 5554, pp. 723–737. Springer (2009)
17. Lehmann, J.: DL-Learner: learning concepts in description logics. Journal of Machine Learning Research (JMLR) **10**, 2639–2642 (2009). URL http://www.jmlr.org/papers/volume10/lehmann09a/lehmann09a.pdf
18. Lehmann, J., Bizer, C., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Journal of Web Semantics **7**(3), 154–165 (2009). DOI doi:10.1016/j.websem.2009.07.002. URL http://jens-lehmann.org/files/2009/dbpedia_jws.pdf
19. Lehmann, J., Bühmann, L.: Ore - a tool for repairing and enriching knowledge bases. In: 9th Int. Semantic Web Conf. (ISWC2010), LNCS. Springer (2010). DOI doi:10.1007/978-3-642-17749-1_12. URL http://svn.aksw.org/papers/2010/ORE/public.pdf
20. Leser, U., Naumann, F.: Informationsintegration - Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen. dpunkt.verlag (2007)
21. Mika, P.: Year of the monkey: Lessons from the first year of searchmonkey. In: GI Jahrestagung, *LNI*, vol. 154, p. 387. GI (2009)
22. Morsey, M., Lehmann, J., Auer, S., Stadler, C., Hellmann, S.: DBpedia and the Live Extraction of Structured Data from Wikipedia. Program: electronic library and information systems **46**, 27 (2012). URL http://svn.aksw.org/papers/2011/DBpedia_Live/public.pdf
23. Ngonga Ngomo, A.C.: Learning conformation rules for linked data integration. In: Proceedings of Seventh International Workshop on Ontology Matching (2012)
24. Ngonga Ngomo, A.C.: Link discovery with guaranteed reduction ratio in affine spaces with minkowski measures. In: Proceedings of ISWC (2012)
25. Ngonga Ngomo, A.C.: On link discovery using a hybrid approach. Journal on Data Semantics **1**, 203 – 217 (2012)
26. Ngonga Ngomo, A.C., Auer, S.: Limes - a time-efficient approach for large-scale link discovery on the web of data. In: Proceedings of IJCAI (2011)
27. Ngonga Ngomo, A.C., Heino, N., Lyko, K., Speck, R., Kaltenböck, M.: Scms - semantifying content management systems. In: ISWC 2011 (2011)
28. Ngonga Ngomo, A.C., Lyko, K.: Eagle: Efficient active learning of link specifications using genetic programming. In: Proceedings of ESWC (2012)
29. Nikolov, A., D'Aquin, M., Motta, E.: Unsupervised learning of data linking configuration. In: Proceedings of ESWC (2012)
30. Rahm, E., Thor, A., Aumueller, D., Do, H.H., Golovin, N., Kirsten, T.: ifuice - information fusion utilizing instance correspondences and peer mappings. In: WebDB, pp. 7–12 (2005)
31. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: Fedx: Optimization techniques for federated query processing on linked data. In: International Semantic Web Conference (1), *LNCS*, vol. 7031, pp. 601–616. Springer (2011). URL http://dblp.uni-trier.de/db/conf/semweb/iswc2011-1.html#SchwarteHHSS11
32. Sequeda, J.F., Miranker, D.P.: Ultrawrap: SPARQL Execution on Relational Data. Poster at the 10th Int. Semantic Web Conf. (ISWC2011) (2011). URL https://files.ifi.uzh.ch/ddis/iswc_archive/iswc/ab/2011pre/iswc2011.semanticweb.org/fileadmin/iswc/Papers/PostersDemos/iswc11pd_submission_94.pdf
33. Sertkaya, B.: OntocomP: A protégé plugin for completing OWL ontologies. In: 6th European Semantic Web Conf., ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, *LNCS*, vol. 5554, pp. 898–902. Springer (2009). URL http://dx.doi.org/10.1007/978-3-642-02121-3
34. Shekarpour, S., Auer, S., Ngonga Ngomo, A.C.: Question answering on interlinked data. In: World Wide Web Conference (2013)
35. Soru, T., Ngonga Ngomo, A.C.: Active learning of domain-specic distances for link discovery. In: Proceedings of JIST (2012)
36. Sundara, S., Atre, M., Kolovski, V., Das, S., Wu, Z., Chong, E.I., Srinivasan, J.: Visualizing large-scale rdf data using subsets, summaries, and sampling in oracle. In: ICDE, pp. 1048–1059. IEEE (2010)
37. Unbehauen, J., Stadler, C., Auer, S.: Accessing relational data on the web with sparqlmap. In: JIST (2012). URL http://svn.aksw.org/papers/2012/SPARQLComponent/JIST_SparqlMap/public.pdf