Multilingualität und Linked Data

Philipp Cimiano und Christina Unger

CITEC, Universität Bielefeld
Inspiration 1, 33619 Bielefeld, Deutschland
{cimiano|cunger}@cit-ec.uni-bielefeld.de

Zusammenfassung

Linked Data ist eine flexible Technologie für die Repräsentation und Verlinkung von heterogenen Daten. Solche Daten können offene Daten sein, die im Kontext des Webs verfügbar sind, aber auch interne Unternehmensdaten, die miteinander über Organisationseinheiten und Strukturen hinweg verknüpft sind. In beiden Fällen werden diese Daten oft über Länder- und Sprachgrenzen hinweg erzeugt und benutzt. Das ist insbesondere der Fall für Unternehmen, die international präsent sind und dementsprechend Filialen in mehreren Ländern betreiben, oder für Organisationen, die international operieren. Dabei stellen sich zwei Fragen: Wie können Daten oder Prozesse, die auf Daten zugreifen, über Länder und Sprachen hinweg synchronisiert oder gar integriert werden? Und wie können Daten sprachübergreifend zugänglich gemacht werden? Für beide Zwecke ist es unabdingbar, dass Daten und Prozesse mit Informationen angereichert werden, wie sie in verschiedenen Sprachen verbalisiert werden.

In diesem Kapitel geben wir eine Übersicht über das Themenfeld Multilingualität und Linked Data, vor allem in Hinblick auf neuere Entwicklungen und deren mögliche Anwendungen. Nach einer kurzen Einführung und Motivation zeigen wir die Herausforderungen auf, die sich aus der Nutzung von Linked Data über Sprachgrenzen hinweg ergeben. Dann besprechen wir Verfahren, mit denen Datenschemas, die für verschiedene Länder entwickelt wurden, synchronisiert werden können, um die Aggregation und Integration von Daten über Länder und Sprachgrenzen hinweg zu ermöglichen. Darüber hinaus diskutieren wir, wie Linked Data mit linguistischen Informationen angereichert werden kann, und betrachten einige Anwendungen, die zeigen, wie solche Informationen für die Generierung und die Interpretation natürlicher Sprache verwendet werden können, um einen sprachübergreifenden Zugang zu Linked Data zu ermöglichen.

1 Einführung

Die Internationalisierung nimmt in vielen Bereichen zu. Unternehmen sind immer stärker international tätig und pflegen Kundenbeziehungen oder Beziehungen zu anderen Unternehmen oder Lieferanten in verschiedenen Ländern. Auch im Rahmen der Politik nimmt die Notwendigkeit von internationalen Absprachen und die Einigung auf gemeinsame Regeln immer weiter zu, z.B. in Bereichen wie Import und Export, Einwanderung oder Transaktionssteuern.

In solchen Szenarien und Anwendungen müssen Prozesse und Regeln über Sprachgrenzen hinweg harmonisiert werden, Daten in verschiedenen Sprachen integriert und möglicherweise gemeinsame sprachübergreifende Taxonomien oder Terminologien entwickelt werden.

Das Überbrücken von Sprachbarrieren ist eine nicht-triviale Aufgabe. Durch die konsequente Verwendung von URIs und RDF als Datenmodell unterstützt zwar Linked Data prinzipiell die Integration von Daten, es bedarf aber dennoch Algorithmen, die Entsprechungen (*Alignments*) zwischen Daten in verschiedenen Sprachen finden können, sowie Verfahren, die Daten oder Vokabulare in eine bestimmte Sprache übersetzen. Letzteres wird in Analogie zur derselben Problematik in der Softwareentwicklung *Lokalisierung* genannt. Lokalisierung ist essentiell für Anwendungen, in denen Linked Data als Datenbasis verwendet wird und Sprecher unterschiedlicher Kulturen oder sprachlicher Kontexte mit diesen Daten interagieren oder sie abfragen können sollen.

In diesem Kapitel geben wir einen Überblick über Probleme und Fragestellungen, die sich aus der Notwendigkeit ergeben, auf Linked Data über Sprachen hinweg zugreifen zu können, sowie verlinkte Datensätze oder Vokabulare und Taxonomien, die zur Beschreibung von Daten verwendet werden, über Sprachgrenzen hinweg zu harmonisieren. Zuerst geben wir einen Überblick über die Behandlung von Multilingualiät in Linked Data. Wir besprechen dann insbesondere Verfahren zur Lokalisierung bzw. Übersetzung von Vokabularen, Ontologien und Taxonomien sowie für das sprach-übergreifende Alignment von Ontologien oder Vokabularen, die in unterschiedlichen Sprachen vorliegen. Insbesondere präsentieren wir das Lexikonformat *lemon* und diskutieren einige Anwendungen für *lemon*-angereicherte Linked-Data-Quellen.

2 Multilingualität in Linked Data

Für viele Anwendungen ist es wichtig, Linked Data mit Informationen darüber anzureichern, wie bestimmte Vokabularelemente in verschiedenen Sprachen ausgedrückt werden, insbesondere Individuen, Klassen und Eigenschaften.

Betrachten wir als Beispiel den fiktiven Fall eines Weinhändlers, der die von ihm angebotenen Weinsorten auf seiner Webseite veröffentlichen möchte. Er hat Kunden in Deutschland, im Vereinigten Königreich und in Spanien und möchte die Informationen daher in drei Sprachen anbieten. In RDF können Bezeichnungen für eine Klasse durch die Eigenschaft `label` ausgedrückt werden (Beispiele werden hier und im Folgenden immer in Turtle-Syntax angegeben, wobei `onto` ein Präfix ist, das die URI der Ontologie abkürzt):

```
1 onto:Rotwein rdf:type rdfs:Class ;
2              rdfs:label "Rotwein"@de ;
3              rdfs:label "red wine"@en ;
4              rdfs:label "vino rojo"@es .
```

Gleiches gilt natürlich für Eigenschaften, die in einem Vokabular verwendet werden. Betrachten wir z.B. die Eigenschaft `hatPreis`, die den Preis eines Weines angibt. Auch diese Eigenschaft kann in mehreren Sprachen ausgedrückt werden:

```
onto:hatPreis rdf:type rdfs:Property ;
              rdfs:label "Preis"@de ;
              rdfs:label "price"@en ;
              rdfs:label "precio"@es .
```

Die URIs `onto:Rotwein` und `onto:hatPreis` sind dabei sprachunabhängige Identifikatoren, die für die Klasse der Rotweine bzw. für die Eigenschaft, einen bestimmten Preis zu haben, stehen. Der Einfachheit halber haben sie hier mnemonische Namen, es sollte aber beachtet werden, dass URIs nur beliebige Zeichenfolgen sind. Um auf Daten, die diese Identifizierer verwenden, in verschiedenen Sprachen zugreifen zu können, ist es daher wichtig, dass Labels in diesen Sprachen zur Verfügung stehen. Leider ist das Hinzufügen von multilingualen Labels bisher keine gängige Praxis im Semantic Web. Bisherige Untersuchungen haben gezeigt, dass lediglich 21% der RDF-Literale einen Sprach-Tag haben. Darüber hinaus ist Englisch eindeutig die führende Sprache im Linked-Data-Web—in dem Sinne, dass sie 85% aller Sprach-Tags ausmacht (siehe [14]).

Zusätzlich ergeben sich in der Praxis zwei Herausforderungen. Die erste entsteht dadurch, dass zuweilen über Länder hinweg verschiedene Taxonomien und Ontologien mit verschiedenen URIs verwendet werden. Es könnte zum Beispiel sein, dass ein weiterer fiktiver Weinhändler eine ähnliche Weinontologie wie die obige entwirft und dabei den Identifikator `RedWine` für die Klasse der Rotweine benutzt sowie eine Eigenschaft `price` definiert, die der Eigenschaft `hatPreis` entspricht. Um Daten, die in beiden Ontologien ausgedrückt werden, aggregieren zu können, müssen die entsprechenden Vokabulare aufeinander abgebildet werden. Dazu will man festhalten, dass bestimmte Klassen und Eigenschaften, obwohl sie verschiedene URIs haben, ein und dasselbe Konzept darstellen. Das ist möglich mit Hilfe der Eigenschaften `owl:equivalentClass` und `owl:equivalentProperty` (und analog `owl:sameAs` für Individuen):

```
onto:RedWine owl:equivalentClass    onto:Rotwein .
onto:price   owl:equivalentProperty onto:hatPreis .
```

In Abschnitt 3 werden Verfahren genauer betrachtet, die Ontologien bzw. Vokabulare automatisch in mehrere Sprachen übersetzen, sowie Verfahren, die Vokabulare in verschiedenen Sprachen aufeinander abbilden.

Die zweite Herausforderung ergibt sich aus der Tatsache, dass die Konzeptualisierung von Daten in einer Ontologie und die natürlichsprachliche Konzeptualisierung der Welt nicht immer übereinstimmen. Dadurch sind in manchen Fällen linguistisch komplexe Informationen nötig, die nicht allein über Labels ausgedrückt werden können. Wenn man zum Beispiel angeben will, dass die Eigenschaft `Erzeuger`, die den Erzeuger eines Weines angibt, mit Hilfe des Verbes "anbauen" verbalisiert werden kann, ist zusätzlich hilfreich anzugeben, dass das Tripel

```
onto:Eiswein onto:Erzeuger onto:Weingut_Grau .
```

durch die Sätze “Das Weingut Grau baut Eiswein an” oder “Eiswein wird vom Weingut Grau angebaut” ausgedrückt werden kann, nicht aber durch “Das Weingut Grau wird von Eiswein angebaut”.

Als weiteres Beispiel nehmen wir an, obiger Weinhändler möchte auf seiner Webseite Weine mit dem Adjektiv “preiswert” beschreiben, die über die Eigenschaft `hatPreis` mit einem Wert bis zu 12 EUR (oder 10 Pfund für die britischen Kunden) verbunden sind. Die Klassen der preiswerten Weine lässt sich zwar als Restriktionsklassen (`owl:RestrictionClass`) definieren, aber gerade wenn man eine Ontologie nicht selber entwickelt und pflegt, sondern eine schon vorhandene nutzt, kommt es vor, dass man sprachliche Ausdrücke für Konstrukte braucht, die in der Ontologie nicht explizit definiert und benannt sind. Labels sind dann nicht ausreichend, da man das Label “preiswert” weder der Eigenschaft `hatPreis` noch den Zahlenwerten 1–12 zuordnen kann.

In Abschnitt 4 stellen wir daher *lemon* vor, ein Lexikonmodell für Ontologien, das es erlaubt, komplexe linguistische Informationen zu erfassen und (einfachen oder komplexen) Ontologiekonzepten zuzuordnen.

3 Ontologie-Lokalisierung und sprachübergreifendes Ontologie-Alignment

In diesem Abschnitt führen wir kurz in die Probleme der Ontologie-Lokalisierung und des Ontologie-Alignments ein. Im Bereich der Ontologie-Lokalisierung besprechen wir die unterschiedlichen Typen von Lokalisierungaktivitäten und verweisen auf aktuelle Entwicklungen und Prototypen, ohne jedoch auf technische Details einzugehen. Anschließend gehen wir auf das Ontologie-Alignment ein und beschreiben als Beispiel ein System, das im Kontext des Monnet-Projektes entwickelt wurde, um Vokabulare für die Finanzberichterstattung aus unterschiedlichen Ländern Europas aufeinander abzubilden.

3.1 Ontologie-Lokalisierung

Laut Cimiano et al. [6] kann die Lokalisierung einer Ontologie wie folgt definiert werden:

> *Ontologie-Lokalisierung bezeichnet den Prozess der Anpassung einer gegebenen Ontologie an die Anforderungen einer bestimmten Gemeinschaft, die durch eine gemeinsame Sprache, Kultur oder geo-politische Umgebung charakterisiert ist.*

Die Aufgabe, eine Ontologie zu lokalisieren ist analog zum Problem der Software-Lokalisierung zu verstehen. In der Software-Industrie müssen Software-Produkte an die kulturellen Gegebenheiten und auch an die Sprache der Nutzer angepasst werden. In erster Linie muss die Dokumentation und auch die graphische Oberfläche übersetzt werden. In einigen Fällen muss aber auch die Funktion der Software angeglichen werden, zum Beispiel wenn sich die Anforderungen von Nutzern verschiedener sprachlicher und kultureller Kontexte unterscheiden. Im

ersten Fall bleibt die eigentliche Funktion der Software unberührt und die Anpassung findet nur oberfächlich statt, nämlich nur an den Stellen wo eine Interaktion mit dem Nutzer stattfindet. Im zweiten Fall sind tiefergehende Anpassungen an einem Softwareprodukt notwendig.

Ähnlich ist die Situation bei der Lokalisierung von Ontologien. In einigen Fällen ist es ausreichend, die *lexikalische Ebene* zu übersetzen. Praktisch bedeutet das, dass lediglich die Labels der entsprechenden Klassen, Individuen oder Eigenschaften übersetzt werden. In anderen Fällen ist eine tiefergehende Anpassung der *konzeptuellen Ebene* notwendig, d.h. es müssen Begriffe neu definiert werden oder gar neu eingeführt werden, um die Ontologie an die Gegebenheiten einer anderen Kultur anzupassen.

Eine weitere wichtige Dimension ist der Zweck, für den die Ontologie angepasst wird. Man unterscheidet hier zwischen einer *funktionalen Anpassung* und einer *beschreibenden Anpassung*. Im Falle der funktionalen Anpassung muss die angepasste Ontologie für die Zielgemeinschaft den gleichen Zweck bzw. die gleiche Funktion erfüllen, welche die ursprüngliche Ontologie für die kulturelle Umgebung erfüllt, für die sie entwickelt wurde. Im Falle einer Anpassung zu beschreibenden Zwecken ist das Ziel, die Ontologie für die Zielgemeinschaft zugänglich zu machen, indem die Begriffe in der Sprache der Zielgemeinschaft beschrieben werden. Betrachten wir als Beispiel den Fall einer Ontologie, die politische Ämter modelliert. Eine deutsche Ontologie würde Begriffe wie "Staatsoberhaupt" oder "Regierungschef" definieren sowie die entsprechenden Unterbegriffe "Bundespräsident" und "Bundeskanzler". Falls nun einer anderen kulturellen Gemeinschaft, zum Beispiel Sprechern des angelsächsischen Sprachraums, Zugang zu dieser Ontologie gegeben werden soll, zum Beispiel um das politische System Deutschlands zu verstehen, würde es ausreichen die obigen Begriffe wörtlich zu übersetzen, wie in folgender Tabelle angegeben:

DE	EN	ES
Staatsoberhaupt	Head of State	Jefe de Estado
Regierungschef	Head of Government	Jefe de Gobierno
Bundespräsident	President	Presidente
Bundeskanzler	Federal Chancellor	Canciller Federal

Falls aber die Ontologie zu funktionalen Zwecken übersetzt wird, müssen die Begriffe durch funktional äquivalente Begriffe aus der Zielgemeinschaft ersetzt werden, zum Beispiel wie folgt:

DE	EN	ES
Staatsoberhaupt	Head of State	Jefe de Estado
Regierungschef	Head of Government	Jefe der Gobierno
Bundespräsident	Queen/King	Rei/Reina
Bundeskanzler	Prime Minister	Presidente

Dabei reicht es in der Regel nicht aus lediglich den Konzepten Labels in einer anderen Sprachen zu geben, da sich die Konzepte inhaltlich unterscheiden. Die Begriffe "Bundeskanzler" und "Prime Minister" zum Beispiel beschreiben Ämter, zwischen denen es länderspezifische Unterschiede hinsichtlich der Befugnisse, der

Rolle und des Amstverständnisses gibt. Also wird bei der Lokalisierung das Konzept "Bundeskanzler" durch ein neues Konzept "Prime Minister" ersetzt, welches der Welt der Zielgemeinschaft entspricht.

In einigen Fällen müssen Konzepte nicht nur durch neue ersetzt, sondern auch verfeinert oder verallgemeinert werden. Betrachten wir den Begriff "Fluss" im Deutschen bzw. "river" im Englischen. Bei der Anpassung dieses Begriffes für eine französischsprechende Gemeinschaft muss beachtet werden, dass im Französischen die Unterscheidung gemacht wird zwischen "rivière", einem Fluss, der in einen anderen Fluss mündet, und einem "fleuve", einem Fluss, der in ein Meer mündet. Diese Unterscheidung führt dazu, dass bei der Lokalisierung einer deutschen oder englischen Ontologie der Begriff "Fluss" bzw. "river" entsprechend dadurch verfeinert werden muss, dass Unterklassen für "fleuve" und "rivière" eingeführt werden. Die entgegengesetzte Situation besteht dann bei der Lokalisierung einer französischen Ontologie für einen deutschsprachigen oder englischsprachigen Kontext. In diesem Fall können die Unterklassen für "fleuve" und "rivière" entfernt werden. Alternativ dazu kann man auch die Unterscheidung belassen und entweder keine Lexikalisierung der Unterklassen in der Zielsprache angeben oder die allgemeineren Begriffe "Fluss" und "river" als Labels für beide Unterklassen verwenden.

In den letzten Jahren sind verschiedene Methoden und Werkzeuge entwickelt worden, welche die Lokalisierung einer Ontologie unterstützen. Im Rahmen des NeOn-Projektes[1], zum Beispiel, wurde der *LabelTranslator* entwickelt [19], ein regelbasiertes System, das eine Vielzahl von Übersetzungsquellen (auch solche, die im Web verfügbar sind) konsultiert um passende Übersetzungen zu ermitteln und verschiedene Übersetzungsalternativen zu gewichten. Der *LabelTranslator* wurde in das *Neon Toolkit*[2] integriert. Im Rahmen des Monnet-Projektes wurden desweiteren Werkzeuge entwickelt, die Verfahren der statistischen maschinellen Übersetzung anwenden, wie sie zum Beispiel auch in *Google Translate* oder *Bing Translate* verwendet werden, um Wahrscheinlichkeiten für verschiedene Übersetzungskandidaten zu ermitteln. Dieses Übersetzungsmodul wurde in das *Be Informed Studio*[3] integriert (siehe [8]), einem Modellierungswerkzeug der niederländischen Firma *Be Informed*.

3.2 Ontologie-Alignment

In vielen Anwendungsfällen sind tatsächlich verschiedene Ontologien für unterschiedliche sprachliche Gemeinschaften vorhanden. Für viele Zwecke ist es nötig, diese unterschiedlichen Ontologien aufeinander abzubilden, um zum Beispiel die Integration und Interoperabilität von Daten zu gewährleisten. Nehmen wir zum Beispiel einen Analysten, der den Jahresumsatz, das Kapital und die Liquidität verschiedener IT-Firmen in Europa vergleichen möchte. Zwar sind in den meisten Ländern Unternehmen verpflichtet ihre Kennzahlen jährlich offenzulegen,

[1] Siehe: `http://www.neon-project.org`, aufgerufen am 17.03.2014

[2] Siehe: `http://neon-toolkit.org/wiki/Main_Page`, aufgerufen am 17.03.2014

[3] Siehe: `http://www.beinformed.nl/BeInformed/website/en/EN/Studio`, aufgerufen am 17.03.2014

allerdings verwenden verschiedene Länder unterschiedliche Konzeptualisierungen und Vokabulare – eine Situation, die den Vergleich, Integration und Aggregation der Daten deutlich erschwert. Zum Beispiel wird in Deutschland die GAAP-Taxonomie des Handelsgesetzbuches verwendet und in Italien die *Tassonomia relativa ai Principi Contabili Italiani*. Ohne eine Abbildung der verschiedenen Taxonomien aufeinander können Daten nur sehr schwer integriert und verglichen werden. Das Ontologie-Alignment hat daher zum Ziel, Begriffe verschiedener Taxonomien aufeinander abzubilden. In multilingualen Kontexten kann man nach Spohr et al. [22] folgende Fälle unterscheiden:

- **Einsprachiges (monolinguales) Ontologie-Alignment:** Die Ontologien, die aufeinander abgebildet werden sollen, benutzen eine gemeinsame Sprache, die für das Alignment genutzt werden kann.
- **Mehrsprachiges (multilinguales) Ontologie-Alignment:** Die Ontologien, die aufeinander abgebildet werden sollen, haben mehrere Sprachen gemein, so dass auch Übereinstimmungen zwischen den Labels in verschiedenen Sprachen für das Alignment verwendet werden können.
- **Sprachübergreifendes (crosslinguales) Ontologie-Alignment:** Die Ontologien, die aufeinander abgebildet werden sollen, teilen keine Sprache. In diesem Fall werden die Labels der einen Ontologie in die Sprache der anderen Ontologie übersetzt, oder die Labels beider Ontologien werden in eine dritte Sprache (eine sogenannte *Pivot-Sprache*) übersetzt.

Im Rahmen des Monnet-Projektes haben Spohr et al. [22] ein Verfahren für das Alignment von verschiedenen Ontologien entwickelt, das in den drei oben genannten Szenarien eingesetzt werden kann. Dazu werden einerseits statistische maschinelle Übersetzungsdienste wie *Bing Translate*[4] benutzt, um Labels in verschiedene Sprachen zu übersetzen. Andererseits basiert das Verfahren auf maschinellen Lernverfahren, die anhand gegebener Beispielabbildungen eine lineare Gewichtung verschiedener Merkmalsindikatoren lernen. Mit Hilfe dieser kann dann, gegeben ein Konzept aus einer Taxonomie, das passendste Konzept aus einer anderen Taxonomie bestimmt werden. Dabei werden folgende Indikatoren verwendet:

- **Ähnlichkeit der Labels auf der Ebene der Zeichenkette**: Hier werden zum einen Ähnlichkeitsmaße verwendet, welche die Reihenfolge der Wörter in einem Label betrachten (z.B. Levensthein-Distanz), und zum anderen solche, die die Reihenfolge ignorieren (z.B. Kosinusähnlichkeit). Die Ähnlichkeitswerte werden über die verschiedenen Labels in den verschiedenen Sprachen aggregiert, da Konzepte sogar innerhalb einer Sprache verschiedene Labels haben können.
- **Strukturelle Merkmale** nutzen die taxonomische Struktur, insbesondere die Ober- und Unterbegriffe des abzubildenden Begriffes, um diese mit den Ober- bzw. Unterbegriffen eines Kandidatenkonzeptes zu vergleichen.

[4] Siehe: `http://www.bing.com/translator`, aufgerufen am 17.03.2014

Im Falle der Reporting-Taxonomien, die im Rahmen der beschriebenen Fallstudie betrachtet wurden, werden taxonomische Beziehungen verwendet, um rekursiv zu spezifizieren, wie bestimmte Größen, zum Beispiel die Liquidität eines Unternehmens, aus anderen Größen berechnet werden. Denn die Information darüber, wie bestimmte Kennzahlen aus anderen Kennzahlen berechnet werden, ist entscheidend bei der Frage, ob zwei Begriffe wirklich äquivalent sind. Um solche Informationen zu vergleichen, wird sowohl die Anzahl der verschiedenen Kennzahlen verglichen, aus denen sich beide Begriffe zusammensetzen, als auch die Labels dieser Kennzahlen.

Diese Indikatoren und ihre Berechnung sowie das Verfahren für das Training der *Support Vector Machines* ist detailliert in Spohr et al. [22] beschrieben. Wir abstrahieren an dieser Stelle von technischen Details und gehen nur auf die Anwendung im Kontexts des Alignments von Business-Reporting-Taxonomien ein. Insbesondere wurde das Verfahren auf die folgenden drei Taxonomien angewendet:

- Die *XEBR Kerntaxonomie* wurde von der *XBRL Europe Business Registers Working Group*[5] entwickelt und umfasst 269 Buchhaltungskonzepte, die in vielen nationalen Taxonomien vorkommen und mit englischen Labels versehen sind.
- Die Taxonomie *Principi Contabili Italiani*[6] (ITCC) aus dem Jahr 2011 umfasst 444 Begriffe mit englischen, italienischen, französischen und deutschen Labels.
- Die *GAAP-Taxonomie des Deutschen Handelsgesetzbuches*[7] (HGG) aus dem Jahre 2011 umfasst 3 146 Begriffe mit englischen und deutschen Labels.

Die Ergebnisse für die verschiedenen Szenarien (monolingual, multilingual und sprachübergreifend), die Spohr et al. [22] berichten, lassen folgende Schlussfolgerungen zu: In den meisten Fällen verbessert die Verwendung verschiedener Sprachen die Ergebnisse. Auch die Verwendung von strukturellen Informationen verbessert die Qualität der Ergebnisse, im Allgemeinen um ungefähr 5 %. Und selbst für den sprachübergreifenden Fall, in dem keine Labels in einer gemeinsamen Sprache vorhanden sind und die Übersetzung automatisch in verschiedenen Sprachen erfolgt, sind die Ergebnisse positiv: In 50 % der Fälle ist das korrekte Konzept an der ersten Position des Rankings, in über 70 % der Fälle unter den ersten 5, und in fast 80 % der Fälle unter den ersten 10. Ein Experte müsste also pro Begriff eine kleine Anzahl von Vorschlägen manuell überprüfen, was den Aufwand und die benötigten Ressourcen, um zwei Taxonomien aufeinander abzubilden, deutlich senkt.

Eine Übersicht über das Problem des Ontologie-Alignments sowie den gängigen Ansätzen dafür geben Euzenat und Shvaiko [10]. Einen genaueren Überblick

[5] Siehe: `http://www.xbrleurope.org/working-groups/xebr-wg`, aufgerufen am 17.03.2014
[6] Siehe: `http://www.xbrl.org/it`, aufgerufen am 17.03.2014
[7] Siehe: `http://www.xbrl.de`, aufgerufen am 17.03.2014

über Verfahren im Bereich des cross-lingualen Ontologie-Alignments geben Trojahn et al. [24]. Ein umfassende Einführung in das Thema bietet außerdem das Buch *Ontology Matching* [10].

4 Das *lemon*-Modell und Anwendungen

In diesem Abschnitt wollen wir eine kurze Einführung in das Lexikonmodell *lemon* geben, das erlaubt, lexikalische Informationen mit Ontologielemementen zu verknüpfen. Anschließend werden wir einige Anwendungen betrachten, die zeigen, wie ein solches Lexikon für die Generierung und die Interpretation natürlicher Sprache verwendet werden kann.

4.1 Das *lemon*-Modell

Das Lexikonmodell *lemon* (*Lexicon Model for Ontologies*) ist ein Modell für die Repräsentation lexikalischer Informationen in Bezug auf eine Ontologie. Das umfasst zum einen morphologische und syntaktische Informationen zu Wortklasse, Wortformen sowie Anzahl und Art der Argumente, zum anderen aber auch semantische Informationen zur Bedeutung von Wörtern und Phrasen in Bezug auf eine Ontologie. Diese Informationen werden in RDF ausgedrückt, so dass ein Lexikon als Linked Data veröffentlicht und geteilt werden kann.

Der Kern des *lemon*-Modells, dargestellt in Abbildung 1, umfasst folgende Elemente:

- Ein *Lexikon* (*lexicon*) ist eine Menge lexikalischer Einträge in einer bestimmten Sprache. Für eine Ontologie können natürlich mehrere Lexika in verschiedenen Sprachen entwickelt werden.
- Ein *lexikalischer Eintrag* (*lexical entry*) ist in der Regel ein Wort (z.B. "Wein") oder eine Phrase (z.B. "nicht-physischer Vermögenswert im Eigentum des Unternehmens").
- Eine *Form* (*form*) stellt die sprachliche Realisierung eines lexikalischen Eintrags dar, üblicherweise die geschriebene Form (*written representation*), aber möglicherweise auch eine phonetische Darstellung.
- Eine *Referenz* ist eine Entität in der Ontologie, d.h. eine Klasse, Eigenschaft oder Entität, und der Kern der Bedeutung eines lexikalischen Eintrags in Bezug auf die Ontologie.
- Die *Bedeutung* (*sense*) eines Eintrages besteht aus einer Referenz, möglicherweise zusammen mit Angaben zu Einschränkungen im Gebrauch eines Wortes oder anderen pragmatischen Informationen.

Zum Beispiel kann man für die Weinhändler-Ontologie aus Abschnitt 2 ein deutsches Lexikon wie folgt definieren:

```
1 @prefix onto:  <http://example.org/ontology#> .
2 @prefix lex:   <http://example.org/lexicon#> .
3 @prefix lemon: <http://www.lemon-model.net/lemon#> .
4
```

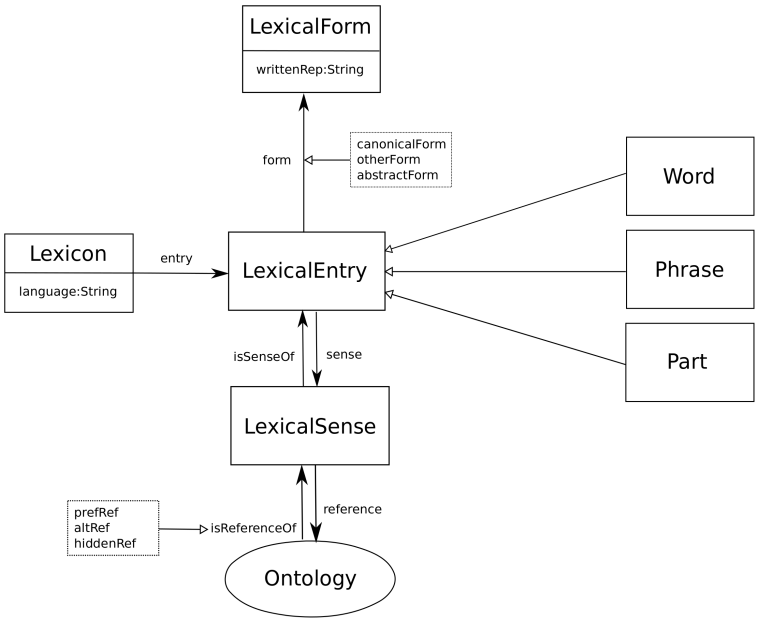

Abb. 1. Der Kern des *lemon*-Modells (Pfeile mit gefüllter Spitze repräsentieren Eigenschaften und Pfeile mit leerer Spitze geben Unterklassen und Untereigenschaften an).

```
5  lex:lexicon a lemon:Lexicon ;
6                lemon:language "de" ;
7                lemon:entry lex:Rotwein ,
8                            lex:Preis ,
9                            lex:anbauen ,
10                           ...  .
```

Das Tripel in Zeile 5 legt die URI für das Lexion fest und definiert es als Individuum vom Typ `Lexicon`, in Zeile 6 wird die Sprache des Lexikons angegeben, Deutsch, und schließlich werden in den Zeilen 7 und 10 die einzelnen lexikalischen Einträge aufgelistet.

Die Bedeutung eines lexikalischen Eintrags wird durch Referenz auf eine spezifische Klasse, Eigenschaft oder ein bestimmtes Individuum in der Ontologie definiert. Um zum Beispiel eine sprachliche Realisierung der Klasse `Rotwein` anzugeben, kann man folgenden lexikalischen Eintrag anlegen:

```
1  lex:Rotwein a lemon:Word ;
2   lemon:canonicalForm [ lemon:writtenRep "Rotwein"@de ] ;
3   lemon:sense         [ lemon:reference  onto:Rotwein ] .
```

Er spezifiziert, dass die Grundform (*canonical form*) von dem lexikalischen Eintrag die geschriebene Form “Rotwein” hat, eine Zeichenkette, die mit dem Sprach-

tag `de` versehen ist, und dass die Bedeutung des Eintrags auf die ontologische Klasse `Rotwein` referiert.

Wichtig ist, dass *lemon* ein Modell ist, dass die Struktur eines Lexikons beschreibt, jedoch keinerlei linguistische Kategorien vorschreibt, also zum Beispiel kein Vokabular bereitstellt, um morphosyntaktische Eigenschaften wie Wortklasse, Wortformen usw. zu erfassen. Zu diesem Zweck kann das Vokabular einer beliebigen linguistischen Ontologie importiert werden. In diesem Kapitel verwenden wir die Ontologie LexInfo [4][8], die über 600 spezifische linguistische Kategorien und Eigenschaften umfasst. Damit kann der Eintrag für Rotwein zum Beispiel wie folgt erweitert werden:

```
lex:Rotwein a lemon:Word;
  lexinfo:partOfSpeech lexinfo:noun;
  lemon:canonicalForm [ lemon:writtenRep "Rotwein"@de;
                        lexinfo:number lexinfo:singular ];
  lemon:otherForm     [ lemon:writtenRep "Rotweine"@de;
                        lexinfo:number lexinfo:plural ];
  lemon:sense         [ lemon:reference onto:Rotwein ].
```

Zusätzlich zu den Kernklassen und -eigenschaften enthält *lemon* verschiedene Module, die weitere Aspekte der Lexikon-Ontologie-Schnittstelle beschreiben, aber nicht für jedes Lexikon relevant sind. Eines dieser Module ist das *Syntax-und-Argumentrollen-Modul*, dargestellt in Abbildung 2, das zusätzliche Klassen und Eigenschaften definiert, welche die Verbindungen zwischen syntaktischen Konstruktionen und semantischen Prädikaten erfassen. Ein sogenannter *Frame* gibt den syntaktischen Kontext an, in dem ein Eintrag vorkommen kann, insbesondere die erforderlichen syntaktischen *Argumente*. Argumente sind häufig markiert, wobei ein *Argumentrollenmarker* (*syntactic role marker*) entweder ein eigener lexikalischer Eintrag sein kann, zum Beispiel eine Präposition, oder eine syntaktische Eigenschaft wie Kasus.

Zum Beispiel würde ein transitiver Verbrahmen festlegen, dass ein Subjekt und ein direktes Objekt erforderlich sind. Da die spezifischen Frames und Argumentrollen nichts mit der Struktur des Lexikoneintrags zu tun haben, sondern bestimmte linguistische Entscheidungen betreffen, wird das Vokabular dafür nicht von *lemon* zur Verfügung gestellt, sondern muss erneut aus einer geeigneten linguistischen Ontologie importiert werden.

Nehmen wir als Beispiel die Eigenschaft `Erzeuger` der Weinhändler-Ontologie. Dafür hatten wir in Abschnitt 2 das Label "anbauen" angegeben und die Schwierigkeit entdeckt, dass aus dem Label selber nicht ersichtlich ist, welchen semantischen Argumenten die syntaktischen Argumente entsprechen. Genau das lässt sich nun einfach in einem Lexikoneintrag erfassen:

```
lex:anbauen a lemon:Word;
  lexinfo:partOfSpeech lexinfo:verb;
  lemon:canonicalForm [ lemon:writtenRep "anbauen"@de ];
  lemon:synBehavior   [ a lexinfo:TransitiveFrame;
```

[8] Siehe: `http://www.lexinfo.net/ontology/2.0/lexinfo`, aufgerufen am 17.03.2014

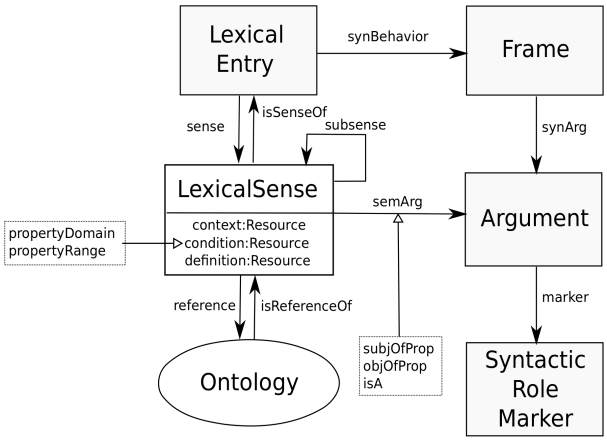

Abb. 2. Das Syntax-und-Argumentrollen-Modul von *lemon*.

```
5                              lexinfo:Subject     _:arg1;
6                              lexinfo:DirecObject _:arg2;
7    lemon:sense             [ lemon:reference onto:Erzeuger;
8                              lemon:subjOfProp    _:arg2;
9                              lemon:objOfProp     _:arg1 ].
```

Der Eintrag hat die Wortform Verb, die kanonische Form "anbauen" und eine Bedeutung, die auf die Eigenschaft `Erzeuger` in der Ontologie referiert. Zusätzlich zu der Referenz auf die Eigenschaft werden die semantischen Argumente dieses Prädikats benannt, `_:arg2` als semantisches Subjekt und `_:arg1` als semantisches Objekt. Dieselben Argument-URIs werden auch bei der Angabe des syntaktischen Kontextes verwendet, in dem der Eintrag vorkommen kann. Der Rahmen wird als `TransitiveFrame`, also der eines transitiven Verbes, festgelegt, mit zwei Argumenten: einem Subjekt, `_:arg1`, das damit dem semantischen Objekt entspricht, und einem direkten Objekt, `_:arg2`, das damit dem semantischen Subjekt entspricht. Damit ist festgelegt, dass das Tripel x `onto:Erzeuger` y sprachlich als "y baut x an" und nicht als "x baut y an" realisiert wird.

Ein weiteres Beispiel, das wir im Abschnitt 2 nicht mit Hilfe von Labels erfassen konnten, ist die Bezeichnung "preiswert" für Weine, deren Preis im Bereich von bis zu 12 EUR bzw. 10 Pfund liegt. Die Klasse aller Entitäten mit einem Preis-Wert zwischen 0 und 12 lässt sich in OWL[9] als Restriktionsklasse wie folgt definieren:

```
1  lex:PreisBis12 a owl:RestrictionClass;
2    owl:onProperty onto:Preis;
```

[9] OWL ist eine Semantic-Web-Sprache zur Repräsentation von Ontologien. Für eine Beschreibung der aktuellen Version, OWL 2, siehe `http://www.w3.org/TR/owl2-primer/` (aufgerufen am 17.03.2014).

```
3    owl:allValuesFrom [ a rdfs:Datatype;
4                        owl:onDatatype xsd:double;
5                        owl:withRestrictions
6                        ( [ xsd:minInclusive  "0.00" ]
7                          [ xsd:maxInclusive "12.00" ] ) ].
```

Diese Klasse ist möglicherweise nicht in der Ontologie definiert, kann aber mit Hilfe des Vokabulars der Ontologie definiert werden. Man kann also sagen, dass sie implizit Teil der Ontologie ist, jedoch in der Ontologie nicht explizit konstruiert und benannt ist. Dieses explizite Konstruieren und Benennen kann nun im Lexikon passieren, was uns erlaubt, einen einfachen Adjektiveintrag für "preiswert" anzugeben, der auf obige Restriktionsklasse referiert:

```
1  lex:preiswert a lemon:Word;
2    lexinfo:partOfSpeech lexinfo:adjective;
3    lemon:canonicalForm [ lemon:writtenRep "preiswert"@de ];
4    lemon:synBehavior   [ a lexinfo:AdjectiveFrame;
5                          lemon:synArg _:arg;
6    lemon:sense         [ lemon:reference lex:PreisBis12;
7                          lemon:isA    _:arg ].
```

Das semantische Argument `isA` steht dabei für ein beliebiges Element der referierten Klasse.

Zusätzlich können in *lemon* Einschränkungen für den Gebrauch eines Eintrages angegeben werden. Zum Beispiel kann es sein, dass der Weinhändler nicht nur Weine, sondern auch Weinzubehör wie Gläser und Dekanter anbietet, und die Ontologie eine einzige Eigenschaft `Hersteller` sowohl für den Erzeuger eines Weines als auch den Hersteller von Weinzubehör benutzt. In diesem Fall sollte diese Eigenschaft als "anbauen" (oder "erzeugen") lexikalisiert werden, falls es sich um Wein handelt, aber als "herstellen", falls es sich um Zubehör handelt. Dazu kann man die Bedeutung eines lexikalischen Eintrags mit einer Bedingung wie `propertyDomain` bzw. `propertyRange` einschränken, die jeweils festlegen, dass das Subjekt bzw. Objekt einer Eigenschaft von einem bestimmten Typ sein muss, in unserem Fall vom Type `Wein` oder `Zubehoer`.

```
1  lex:anbauen a lemon:Word;
2      lemon:sense [ lemon:reference      onto:Hersteller;
3                    lemon:propertyDomain onto:Wein ].
4
5  lex:herstellen a lemon:Word;
6      lemon:sense [ lemon:reference      onto:Hersteller;
7                    lemon:propertyDomain onto:Zubehoer ].
```

Weiterhin würde nun die Bedeutung des Wortes "preiswert" variieren, je nach dem, ob es sich um Wein oder ein bestimmtes Zubehör handelt. Man kann also parallel zu `PreisBis12` weitere Restriktionklassen `PreisBis30` usw. definieren, sowie für den Adjektiveintrag mehrere Bedeutungen angeben, deren Gebrauch jeweils eingeschränkt ist:

```
1  lex:preiswert a lemon:Word ;
```

```
2       lemon:sense [ lemon:reference onto:PreisBis12;
3                     lex:usedFor onto:Wein ],
4                   [ lemon:reference onto:PreisBis30;
5                     lex:usedFor onto:Dekanter ].
6
7   lex:usedFor rdfs:subProperty lemon:condition.
```

Die Einschränkung `usedFor` ist in diesem Fall eine von uns definierte. Ähnliche Einschränkungen sind nützlich, wenn man Lexikalisierungen personifizieren will, wenn man also zum Beispiel bei der Generierung von Texten für fortgeschrittene Endnutzer Fachbegriffe verwenden, für neue Nutzer hingegen einfachere Begriffen verwenden will (siehe z.B. [5]).

Details zu *lemon* sowie weitere Informationen zu den verschiedenen Modulen und zur Modellierung verschiedener lexikalischer Aspekte können im *lemon Cookbook*[10] nachgelesen werden. Außerdem dient *lemon* derzeit als Basis für die Aktivitäten der W3C *Ontology Lexica Community Group*[11], die ein Standard-Modell für die Anreicherung von Ontologien mit lexikalischen Informationen entwickelt.

4.2 Lexika als Linked Data

Da Ontologielexika in RDF, also einer Standard-Semantic-Web-Sprache, repräsentiert werden und auf Elemente einer Ontologie verweisen, stellen sie selber Linked Data dar und können genauso wie Ontologien veröffentlicht und geteilt werden. Außerdem kann *lemon* als eine Art Austauschformat dienen, um lexikalische Ressourcen wie zum Beispiel *WordNet*[12] und *Wiktionary*[13] als Linked Data verfügbar zu machen (siehe [18]).

Schließlich kann man sich vorstellen, dass sich ein Ökosystem von Ressourcen entwickelt, das Ontologien, Lexika für diese Ontologien in unterschiedlichen Sprachen, lexikalische Ressourcen und möglicherweise auch Werkzeuge für die semi-automatische Entwicklung von Lexika sowie für die Generierung von verschiedenen Grammatikformaten aus Lexika umfasst.

4.3 Automatisches Erzeugen von Grammatiken

Die manuelle Entwickung von Grammatiken ist ein ressourcenintensiver Prozess. Besonders bei großen Domänen ist es aufwendig Grammatiken zu konstruieren, sie weiterzuentwickeln und schließlich auch auf andere Sprachen zu portieren.

Ontologielexika können dabei helfen den Prozess der Grammatikgenerierung zu automatisieren. Zum einen erlauben sie sehr reichhaltige linguistische Informationen auszudrücken, zum anderen repräsentieren sie diese Informationen

[10] Siehe: `http://lemon-model.net/learn/cookbook.php`, aufgerufen am 17.03.2014
[11] Siehe: `http://www.w3.org/community/ontolex/`, aufgerufen am 17.03.2014
[12] Siehe: `http://wordnet.princeton.edu`, aufgerufen am 17.03.2014
[13] Siehe: `https://www.wiktionary.org`, aufgerufen am 17.03.2014

auf eine kompakte und theorieneutrale Weise und abstrahieren damit von bestimmten Grammatiktheorien. *lemon*, zum Beispiel, ist weitestgehend unabhägig von bestimmten syntaktischen und semantischen Theorien und erst mit der Wahl einer importierten linguistischen Ontologie legt man sich auf bestimmte linguistische Kategorien fest. Darüber hinaus wird die Bedeutung lexikalischer Einheiten in Bezug auf eine Ontologie angegeben, wodurch das Generieren von semantischen Repräsentationen erleichtert wird, die sich mit der Struktur und dem Vokabular einer bestimmten Ontologie im Einklang befinden.

In früheren Arbeiten haben wir die Generierung von Grammatiken aus *lemon*-Lexika für verschiedene Arten von lexikalisierten Grammatikformaten implementiert, unter anderem für *Lexicalized Tree Adjoining Grammars* (LTAG) und *Grammatical Framework* (GF). Dazu werden zuerst für jeden lexikalischen Eintrag alle nötigen Informationen extrahiert: die Wortklasse, die verschiedene Wortformen, der syntaktische Frame sowie die Anzahl und Art der syntaktischen Argumente, die Bedeutung mit vorhandenen semantischen Argumenten sowie deren Entsprechung zu den syntaktischen Argumenten. Da ein *lemon*-Lexikon in RDF-Format vorliegt, können diese Information mit Hilfe der Abfragesprache SPARQL[14] abgefragt werden. Danach werden basierend auf dem syntaktischen Frame oder der Wortklasse Templates gefüllt, welche die allgemeine Form des Eintrags in dem bestimmten Grammatikformat spezifizieren.

Eine Implementierung dieser Methoden ist auf Bitbucket[15] verfügbar. Details zu den Grammatikformalismen sowie zur Rolle von Ontologielexika und aus Lexika generierten Grammatiken für die Interpretation natürlicher Sprache sind in dem Buch *Ontology-based interpretation of natural language* [7] zusammengefasst.

Wichtig anzumerken ist, dass die generierten Grammatiken nur domänenspezifische Ausdrücke umfassen, aber keinerlei domänenunabhängigen Ausdrücke, wie Determinierer, Pronomen, Hilfsverben und Ausdrücke für Negation oder Koordination, da diese keine Entsprechung in der Ontologie haben und damit auch nicht Bestandteil des Lexikons sind. Eine aus einem Lexikon erzeugte Grammatik stellt also nur ein Grammatikmodul dar, dass in eine umfassendere Grammatik eingebettet werden muss, um in Anwendungen nützlich zu sein. Eine solche umfassendere Grammatik könnte wie in Abbildung 3 organisiert sein. Das heißt es gibt ein Modul, das domänenunabhängige Ausdrücke und Satzstrukturen umfasst. Dieses Modul wird üblicherweise manuell konstruiert und kann dann für jede Domäne wiederverwendet werden. Darauf aufbauend gibt es zum einen ein Modul mit domänenspezifischen Ausdrücken, das automatisch aus dem Ontologielexikon generiert wird, und gegebenenfalls ein Modul mit aufgabenspezifischen, zum Beispiel dialogrelevanten, Ausdrücken. Zusammen bilden die Module eine vollständige Grammatik, die in einer Anwendung verwendet werden kann, zum Beispiel in einer natürlichsprachlichen Schnittstelle zum Endnutzer.

Will man die Grammatik auf eine neue Domäne portieren, so genügt es, die Ontologie und das Ontologielexikon auszutauschen und daraus automatisch ein

[14] Siehe: `http://www.w3.org/TR/rdf-sparql-query/`, aufgerufen am 17.03.2014
[15] Siehe: `https://bitbucket.org/chru/lemongrass`, aufgerufen am 17.03.2014

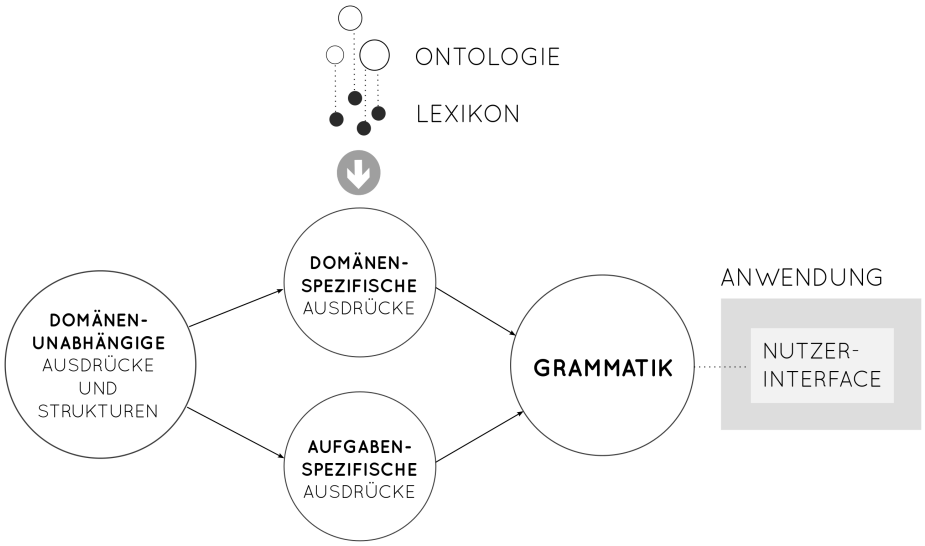

Abb. 3. Grammatikmodularität.

neues domänenspezifisches Modul zu erzeugen. Da mit einer Ontologie mehrere Lexika in verschiedenen Sprachen verknüpft sein können, können außerdem für eine Domäne Grammatikmodule in mehreren Sprachen erzeugt werden. Solch ein multilingualer Kontext setzt allerdings voraus, dass auch die anderen Grammatikmodule in den verwendeten Sprachen vorliegen.

In den folgenden Abschnitten zeigen wir zwei Anwendungsmöglichkeiten für ontologiebasierte Grammatiken: natürlichsprachliche Generierung, zum Beispiel um Linked Data in verschiedenen Sprachen zu verbalisieren, und die Interpretation natürlicher Sprache in Bezug auf eine Ontologie, zum Beispiel um das Abfragen von Linked-Data-Quellen in unterschiedlichen Sprachen zu ermöglichen.

4.4 Natürlichsprachliche Generierung

Eine der Stärken von Linked Data ist, dass Daten in einem strukturierten, maschinenlesbaren Format vorliegen. Aber während auf der einen Seite Maschinen mit diesen Daten operieren, will man sie auf der anderen Seite auch für Menschen zugänglich machen. Unternehmen, die Linked Data benutzen, wollen zum Beispiel ausgehend von diesen Daten Produktbeschreibungen generieren, diese Beschreibungen in verschiedenen Sprachen lokalisieren, usw. Solche Beschreibungen sollen dabei stets im Einklang mit den Daten sein, auch wenn diese sich ändern. Dazu braucht man Methoden, um Linked Data in ein für Menschen leicht verständliches Format wie natürliche Sprache umzuwandeln. Natürlichsprachli-

che Generierung beschäftigt sich daher mit dem automatischen Erzeugen von Texten aus strukturierten Daten.

In den letzten Jahren ist eine Reihe von Systemen entwickelt worden, die sich dieser Aufgabe annehmen [3]. Was alle diese Systeme gemeinsam haben, ist dass sie Wissen brauchen, wie die Elemente einer Ontologie oder Wissensbasis, d.h. die Klassen, Eigenschaften und Individuen, sprachlich ausgedrückt werden. Zum einen können dafür die vorhandenen Labels genutzt werden [20,23], zum anderen bieten Ontologielexika eine reiche Quelle solcher Informationen.

Die meisten Systeme zur natürlichsprachlichen Generierung unterscheiden zwei Phasen: die Auswahl des zu verbalisierenden Inhalts (Was soll gesagt werden?) und die sprachliche Realisierung desselben (Wie soll es gesagt werden?). Systeme folgen dabei typischerweise einer Pipeline, die von Reiter & Dale [21] vorgeschlagen wurde, und erzeugen Text in den folgenden drei Etappen. Zuerst wird der Inhalt und die Struktur des Textes geplant, d.h. welche Fakten versprachlicht werden sollen und in welcher Reihenfolge bzw. in welchen Gruppen. Im nächsten Schritt wird für jedes dieser Fakten ein Satzmuster festgelegt, das dann schließlich im letzten Schritt mit spezifischen Ausdrücken gefüllt wird.

Vor allem im letzten Schritt, wenn Entscheidungen auf lexikalischer Ebene nötig sind, kann ein Ontologielexikon den Generierungsprozess maßgeblich unterstützen. Der deutlichste Fall ist, dass ein Lexikon üblicherweise eine oder mehrere Lexikalisierungsalternativen für ein Ontologieelement angibt. Diese können mit Informationen darüber verknüpft werden, in welchem Kontext welche Lexikalisierung zu bevorzugen ist. Zum Beispiel können bestimmte Lexikalisierungen als Fachbegriffe markiert werden, die dann nur verwendet werden, wenn die Zielgruppe des Textes Experten sind, nicht aber, wenn ein Text für Anfänger oder Nichtexperten erzeugt wird. Ein System, das ein Ontologielexikon zu diesem Zweck benutzt, ist in [5] beschrieben.

Ein System, dass Ontologien verbalisiert, die, ähnlich zu Ontologielexika, mit RDF-Annotation von linguistischen Informationen auf lexikalischer Ebene, aber auch auf Satzebene und in Bezug auf Nutzermodellierung angereichert sind, ist *NaturalOWL* [12]. Der einzige Nachteil von diesem System ist, dass lexikalische Information und Information über Satzmuster sowie Text- bzw. Dialoginformationen nicht voneinander getrennt sind. Es ist also schwierig, schon vorhandene lexikalische Informationen für die Generierung verschiedener Textformen wiederzuverwenden. Das ist genau das Szenario, dass wir im vorigen Abschnitt beschrieben haben: ein Ontologielexikon erfasst lexikalische Informationen, während domänenunabhängige Grammatikmodule Satzmuster sowie Text- und Dialogformen festlegen, die dann variabel miteinander kombiniert werden können.

4.5 Multilinguales Question Answering

Question Answering ist die Aufgabe, automatisch Antworten zu natürlichsprachlichen Fragen aufzufinden. Die Quellen, in denen nach Antworten gesucht wird, sind dabei unterschiedlicher Natur. Traditionell liegt ein starker Fokus auf Textdaten, also dem Auffinden von Antworten aus Zeitungsartikeln, Webseiten, usw.

Aber bereits in den 1960ern und 1970ern wurde begonnen auch strukturierte Daten zu berücksichtigen, insbesondere aus Datenbanken [1]. Heutzutage spielt vor allem Linked Data eine zunehmend wichtige Rolle. Mit der wachsenden Menge semantischer Daten wächst auch das Interesse an Methoden, diese Daten Endnutzern zugänglich zu machen. Eine Möglichkeit ist *Question Answering*, dessen Aufgabe nun darin besteht, ausgehend von einer natürlichsprachlichen Frage eine formale Abfrage zu konstruieren, welche die Antwort(en) aus einer gegebenen Linked-Data-Quelle extrahiert. Das erlaubt Endnutzern ihren Informationsbedarf auf eine einfache und intuitive Art und Weise auszudrücken – einerseits ohne mit Semantic-Web-Sprachen wie RDF und der dazugehörigen Abfragesprache SPARQL vertraut sein zu müssen, und andererseits ohne das den Daten zugrundeliegende Schema kennen zu müssen.

Im Falle unseres Weinhändlers könnte eine Endnutzeranfrage zum Beispiel folgende sein: "Gib mir alle Weine, die im Breisgau angebaut werden und unter 20 EUR kosten." Die entsprechende SPARQL-Abfrage würde dann wie folgt aussehen:

```
SELECT DISTINCT ?w WHERE {
  ?w rdf:type onto:Wein .
  ?w onto:Erzeuger ?x .
  ?x onto:Ort onto:Breisgau .
  ?w onto:Preis ?p .
  FILTER (?p < 20)
}
```

Das Abbilden von natürlichsprachlichen Fragen auf formale Abfragen stellt einige Herausfordungen bereit. Zuerst einmal müssen natürlichsprachliche Ausdrücke auf URIs der den Daten zugrundeliegenden Ontologie abgebildet werden. In einigen Fällen ist das unkompliziert, zum Beispiel entspricht das Wort "Wein" der Ontologieklasse `Wein` und der Name "Breisgau" dem Individuum `Breisgau`, in anderen Fällen aber ist das schwieriger, zum Beispiel muss "angebaut" auf die Eigenschaft `Erzeuger` und "kosten" auf die Eigenschaft `Preis` abgebildet werden, während die Präposition "unter" einem Filter über dem Objekt dieser Eigenschaft entspricht.

Noch einmal komplizierter wird es, wenn die Abfrage kürzer formuliert wird als "Gib mir alle Weine aus dem Breisgau für unter 20 EUR". In diesem Fall sind die relevanten Eigenschaften der Ontologie nicht explizit benannt, sondern hinter semantisch leichten Ausdrücken wie "aus" und "für" versteckt. Gerade die Interpretation von solchen Präpositionen hängt sehr stark vom Kontext und von der zugrundeliegenden Domäne ab. Betrachten wir einen Satz wie "die Veröffentlichung der jährlichen Kennzahlen ist Pflicht für Unternehmen aus dem Dienstleistungssektor", ist leicht ersichtlich, dass die Präpositionen "für" und "aus" hier anders gebraucht werden als in unserem Weinbeispiel und sich auf ganz andere ontologische Elemente beziehen würden.

Hinzu kommt, dass oft nicht nur die Begriffe und die entsprechenden URIs andere sind, sondern dass sich auch die Struktur der natürlichsprachlichen Frage von der der formalen Abfrage unterscheidet. In der Phrase "Weine aus dem

Breisgau", zum Beispiel, gibt es eine linguistische Relation zwischen "Weine" und "Breisgau", in der Ontologie hingegen sind Weine nicht direkt mit einem Ort verbunden, sondern mit einem Erzeuger, der wiederum an einem Ort verankert ist. Die Relation in der Frage entspricht also zwei Tripeln in der SPARQL-Abfrage, `?w onto:Erzeuger ?x` und `?x onto:Ort onto:Breisgau`, wohingegen für die Interpretation der Phrase "Weinbauern aus dem Breisgau" nur letzteres Tripel wichtig wäre.

Es gibt eine Reihe von Ansätzen zu *Question Answering* über Linked Data und das Forschungsinteresse wächst stetig. Einen guten ersten Überblick bietet [17]. Es gibt sowohl Ansätze, die rein auf Ontologielexika aufbauen, zum Beispiel *Pythia* [26], als auch Ansätze, die versuchen, Fragen unabhängig von solchen Ressourcen in Bezug auf ein beliebiges Ontologievokabular zu interpretieren, und zu diesem Zweck verschiedene Strategien anwenden, um die Lücke zwischen natürlicher Sprache und dem zugrundeliegenden Ontologieschema zu schließen. Neuere Arbeiten umfassen zum Beispiel [11], [15] und [25].

In vielen Fällen kann, unabhängig vom genauen Ansatz eines Systems, ein Ontologielexikon helfen, die Diskrepanz zwischen natürlicher Sprache und der Ontologie zu überbrücken. Eine solche Brücke wird vor allem dann unverzichtbar, wenn Daten in verschiedenen Sprachen vorliegen und abgefragt werden. Multilingualität rückt zunehmend in das Interesse der Semantic-Web-Community, da sowohl die Zahl der Daten, die in anderen Sprachen als Englisch veröffentlicht werden, als auch die Zahl der Nutzer, die auf diese Daten zugreifen wollen und nicht Englisch als Muttersprache sprechen, erheblich wächst. Das stellt eine Herausforderung für die meisten aktuellen Ansätze zu *Question Answering* über Linked Data dar, die oft für das Englische entwickelt wurden und bis auf wenige Ausnahmen nicht multilingual sind. Um diese Richtung in der Forschung zu betonen, konzentriert sich die Evaluationskampagne *Question Answering over Linked Data* [16] (QALD), Teil einer größeren *Question-Answering*-Initiative[16], u.a. auf Multilingualität und bietet einen Benchmark mit Fragen in sieben europäischen Sprachen an (Englisch, Deutsch, Spanisch, Italienisch, Französisch, Niederländisch und Rumänisch).

4.6 Weitere Anwendungen

Weitere Anwendung für lexikalisiertes Linked Data sind zum Beispiel folgende:

- Generierung von natürlichsprachlichen Fragen, die ein gegebener Datensatz beantworten kann, um dem Anweder das Verständnis des Inhaltes der Daten zu vereinfachen (ähnlich zu Mathieu d'Acquin et al. [9], allerdings in natürlicher Sprache)
- Informationsextraktion von Relationen aus textuellen Daten (wie zum Beispiel in [13] beschrieben)
- Validierung von RDF-Fakten anhand textueller Quellen (wie ebenfalls in [13] beschrieben)

[16] Siehe: `http://nlp.uned.es/clef-qa/`, aufgerufen am 17.03.2014

– Generierung von Formularen für die Modellierung oder Abfrage von Daten (siehe zum Beispiel [27])
– Generierung von natürlichsprachlichen Zusammenfassungen für verlinkte Datensätze (wie zum Beispiel in [2] beschrieben)

In all diesen Anwendungen wird Wissen darüber benötigt, wie Klassen, Eigenschaften und Individuen eines Datensatzes sprachlich ausgedrückt werden. Demnach können Ontologielexika in all diesen Anwendungen von Interesse und Nutzen sein.

5 Zusammenfassung und Ausblick

In diesem Kapitel haben wir eine Übersicht über Aspekte der Multilingualität im Kontext von Linked Data gegeben. Wir haben dabei die Wichtigkeit der Anreicherung von Linked Data mit Information darüber, wie die verschiedenen Elemente der verwendeten Vokabulare in verschiedenen natürlichen Sprachen lexikalisiert werden, motiviert und Herausforderungen diskutiert, die sich aus der Nutzung von Linked Data in multilingualen Anwendungen ergeben.

Desweiteren haben wir die wichtigen Aufgaben der Ontologie-Lokalisierung und des sprachübergreifenden Ontologie-Alignments eingeführt und einen kurzen Überblick über die Problemstellung und den Stand der Technik in diesen Bereichen gegeben.

Wir haben das *lemon*-Modell eingeführt und gezeigt, wie es dazu genutzt werden kann, um Linked-Data-Vokabulare mit linguistischen Informationen darüber anzureichern, wie die Elemente eines Vokabulars in verschiedenen Sprachen ausgedrückt werden können. Lexikalisierungen wie sie vom *lemon*-Modell in Form von sogenannten Ontologielexika bereitgestellt werden, werden in der Zukunft als Basis für Anwendungen dienen, in denen zwischen Linked Data und einer sprachbasierten Repräsentation vermittelt werden muss. Wir haben drei solcher Anwendungen ausführlicher besprochen: das automatische Erzeugen von Grammatiken, die natürlichsprachliche Generierung von Texten aus strukturierten Daten und das multilinguale *Question Answering* über Linked Data. Außerdem haben wir einige andere Anwendungen skizziert.

Derzeitige Standardisierungsaktivitäten des W3C bauen auf dem *lemon*-Modell auf, um einen Standard für die Anreicherung von Linked-Data-Quellen mit lexikalischen Informationen zu erarbeiten. Die Vision, auf die die Mitglieder der W3C *Ontology Lexicon Community Group* hinarbeiten, ist die eines Linked-Data-Webs, in dem alle relevanten Vokabulare und Datensätze mit einem entsprechenden Lexikon verlinkt werden, in dem Daten und Ontologien über Sprachen hinweg miteinander verlinkt sind und so den sprachübergreifenden Zugriff auf das Datennetzwerk unterstützen. Dabei wird eine wichtige Frage sein, wie die Kosten für die Erzeugung von Lexika reduziert werden können, z.B. durch induktive Verfahren, die aus Daten lernen (siehe z.B. [28]), und durch Crowdsourcing-Verfahren oder kollaborative Arbeitsteilung.

Literatur

1. Ion Androutsopoulos, Graeme D. Ritchie, and Peter Thanisch. Natural language interfaces to databases – an introduction. *Journal of Natural Language Engineering*, 1(1):29–81, 1995.
2. Kalina Bontcheva. Generating tailored textual summaries from ontologies. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *The Semantic Web: Research and Applications*, volume 3532 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2005.
3. Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. Natural language generation in the context of the semantic web. *Semantic Web – Interoperability, Usability, Application*, to appear.
4. Philipp Cimiano, Paul Buitelaar, John McCrae, and Michael Sintek. LexInfo: A declarative model for the lexicon-ontology interface. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2011.
5. Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. Exploiting ontology lexica for generating natural language texts from rdf data. In *Proceedings of the 14th European Workshop on Natural Language Generation (ENLG 2013)*, 2013.
6. Philipp Cimiano, Elena Montiel-Ponsoda, Paul Buitelaar, Mauricio Espinoza Mejía, and Asunción Gómez-Pérez. A Note on Ontology Localization. *Journal of Applied Ontology*, 5(2), 2010.
7. Philipp Cimiano, Christina Unger, and John McCrae. *Ontology-based interpretation of natural language*. Morgan & Claypool, to appear.
8. Monnet (FP-ICT-4-248458) Consortium. D1.1.2 Final Use Case Definition and Scenario Development, 2011. `http://www.monnet-project.eu/Monnet/Monnet/English/Navigation/D2_2`.
9. Mathieu d'Aquin and Enrico Motta. Extracting relevant questions to an rdf dataset using formal concept analysis. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP)*, pages 121–128. ACM, 2011.
10. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, 2nd edition, 2013.
11. André Freitas, Edward Curry, Jo ao Gabriel Oliveira, and Seán O'Riain. A distributional structured semantic space for querying RDF graph data. *International Journal of Semantic Computing*, 5(4):433–462, 2011.
12. Dimitrios Galanis and Ion Androutsopoulos. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In *Proc. 11th European Workshop on Natural Language Generation (ENLG '07)*, pages 143–146, 2007.
13. Daniel Gerber and Axel-Cyrille Ngonga Ngomo. From RDF to Natural Language and Back. In Paul Buitelaar and Philipp Cimiano, editors, *Towards the Multilingual Semantic Web: Principles, Methods and Applications*. Springer, 2011. to appear.
14. Asunción Gómez-Pérez, Daniel Vila-Suero, Elena Montiel-Ponsoda, Jorge Gracia, and Guadalupe Aguado de Cea. Guidelines for multilingual linked data. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics (WIMS'13)*, pages 3:1–3:12. ACM, 2013.
15. Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

16. Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 21(0):3–13, 2013.
17. Vanessa Lopez, Victoria Uren, Marta Sabou, and Enrico Motta. Is question answering fit for the semantic web? a survey. *Semantic Web Journal*, 2:125–155, 2011.
18. John McCrae, Dennis Spohr, and Philipp Cimiano. Linking lexical resources and ontologies on the semantic web with lemon. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications – Volume Part I*, ESWC'11, pages 245–259. Springer, 2011.
19. Mauricio Espinoza Mejía, Elena Montiel-Ponsoda, and Asunción Gómez-Pérez. Ontology localization. In *Proceedings of the 5th International Conference on Knowledge Capture (KCAP09)*, pages 33–40, 2009.
20. Chris Mellish and Xiantang Sun. The Semantic Web as a linguistic resource: Opportunities for natural language generation. *Knowl.-Based Syst.*, 19(5):298–303, 2006.
21. Ehud Reiter and Robert Dale. *Building natural language generation systems*. Cambridge University Press, 2000.
22. Dennis Spohr, Laura Hollink, and Philipp Cimiano. A machine learning approach to multilingual and cross-lingual ontology matching. In *Proceedings of 10th International Semantic Web Conference*, pages 665–680, 2011.
23. Xiantang Sun and Chris Mellish. An experiment on free generationfrom single RDF triples. In *Proc. 11th European Workshop on Natural Language Generation (ENLG '07)*, pages 105–108, 2007.
24. Cáassia Trojahn, Bo Fu, Ondrej Zamazal, and Dominique Ritze. State-of-the-art in Multilingual and Cross-Lingual Ontology Matching. In Paul Buitelaar and Philipp Cimiano, editors, *Towards the Multilingual Semantic Web: Principles, Methods and Applications*. Springer, 2011. to appear.
25. Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st World Wide Web Conference (WWW 2012)*, pages 639–648, 2012.
26. Christina Unger and Philipp Cimiano. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems (NLDB 2011)*, pages 153–160. LNCS 6717, Springer, 2011.
27. Jeroen Van Grondelle and Menno Gülpers. Specifying Flexible Business Processes using Pre and Post Conditions. In Paul Johannesson, John Krogstie, and Andreas L. Opdahl, editors, *Practice of Enterprise Modeling*, volume 92 of *LNBIP*, pages 1–14. Springer, 2011.
28. Sebastian Walter, Christina Unger, and Philipp Cimiano. A corpus-based approach for the induction of ontology lexica. In *Proc. of the 18th International Conference on Natural Language Processing and Information Systems (NLDB)*, volume 7934 of *Lecture Notes in Computer Science*, pages 102–113, 2013.