

{ COD1NG }
DA V1NC1

RHEIN-MAIN 2018

Einführung in Wikidata und Semantic Web

Coding Da Vinci Rhein-Main 2018

Timo Homburg

1 Einführung in Wikidata und Semantic Web

- Das Semantic Web

2 Wikidata Tutorial

- HandsOn Wikidata
- HandsOn SPARQL
- Wikidata in meiner Anwendung

3 Weitere Repositorien

- Linked Geodata
- MusicBrainz
- Linguistic Linked Open Data

4 Zusammenfassung

Wer bin ich?

- **Timo Homburg, M.Sc.**
- **Doktorand an der Hochschule Mainz**
- **Forschung in**
 - **Linked Data/Semantic Web**
 - **Geodaten**
 - **Computerlinguistik**

Für wen ist dieser Vortrag...

- **Leute die noch nie etwas von Semantic Web gehört haben**
- **Leute die gerne etwas zu Wikidata und anderen Datenquellen erfahren möchten**
- **Leute die mit Wikidata in ihren Applikationen arbeiten wollen**

Einführung in Wikidata und Semantic Web

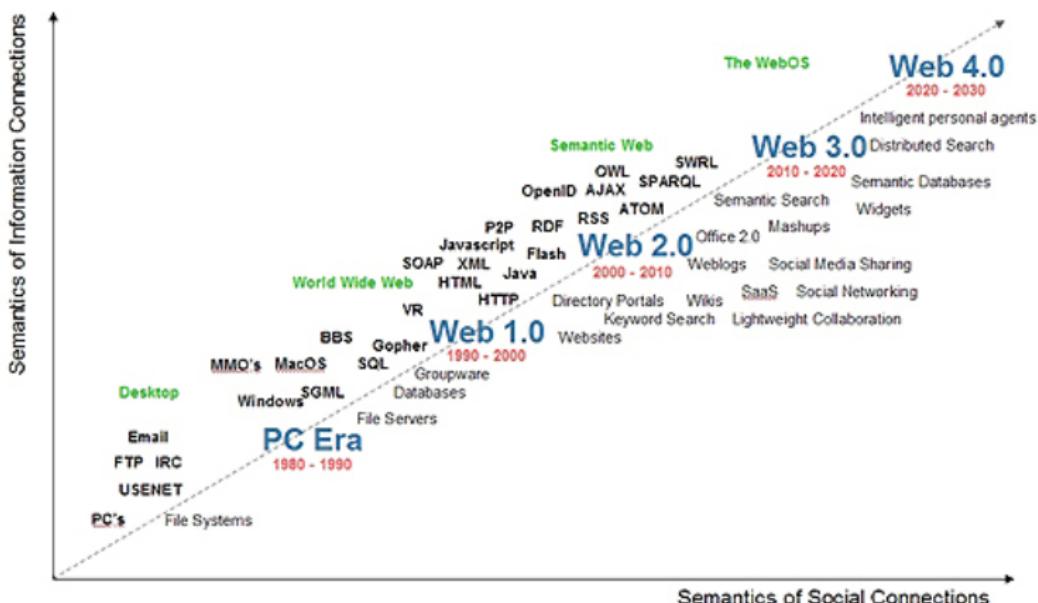
1 Einführung in Wikidata und Semantic Web

- Das Semantic Web

Motivation

- Das Web ist die größte Datensammlung der Welt
- Das Web ist leicht erreichbar
- Das Web kann aktuell sein
- Das Web ist (das beste was wir haben?)

Evolution des WWW



Source: Radar Networks & Nova Spivack, 2007 – www.radarnetworks.com

Ziel: Thematisch (semantisch) strukturierte Daten

- **Mit dem Computer menschlich verständliches Wissen verarbeiten**
- **Thematisch verwandte Datensätze verknüpfen**
- **Beziehungen zwischen Datensätzen explizit semantisch definieren**
- **Einheitliche Vokabularen für Sachverhalte zu definieren**
- **Duplikate zu kennzeichnen bzw. durch Verlinkung zu vermeiden**

Das Semantic Web

Definition

"The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries"

Für uns:

- Toll da haben Leute freie strukturierte, maschinenlesbare Daten ins Netz gestellt, die ich verwenden kann!

Semantic Web - Die Struktur

- Wir nehmen uns kurz 5 Minuten Zeit und entwickeln eine MindMap zum Thema
- MAINZ
- 3,2,1... GO!

MindMap - Was ist das?

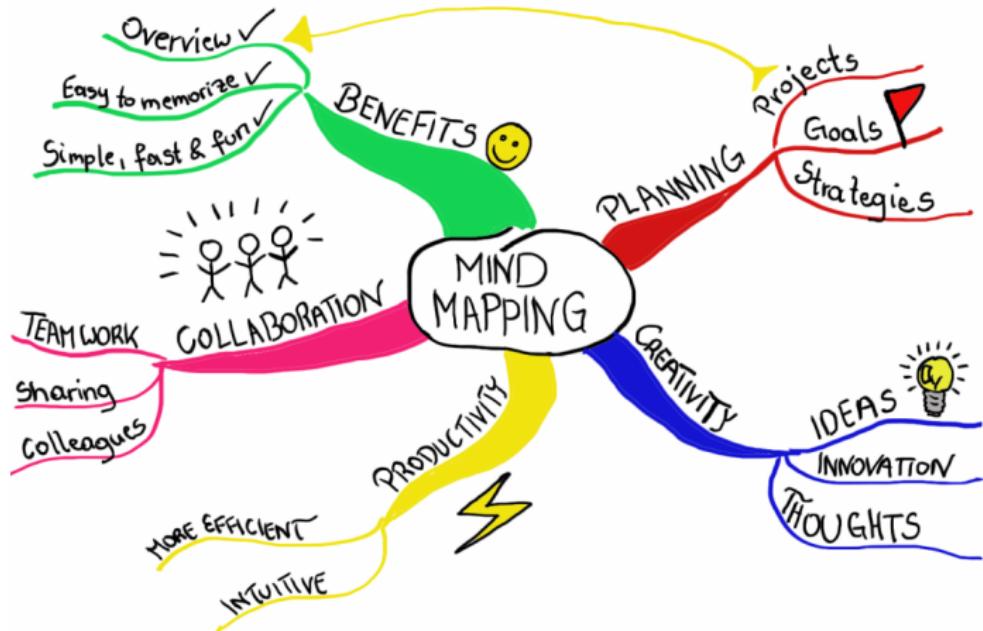


Figure: MindMap Beispiel <https://www.mindmeister.com/blog/why-use-mindmaps/>

Semantic Web - Die Struktur II

- Die Struktur der MindMap entspricht der eines Graphen
- Das Semantic Web ist ebenfalls ein Graph mit ein paar Modifikationen!
- Semantic Web Graphen bestehen aus Triples!

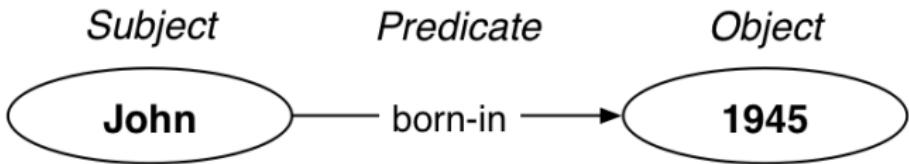


Figure: Triple Beispiel (Quelle: W3C)

Triples sind typisiert!

■ Sowohl Subjekt, Prädikat und Objekt haben Typen

An ontology creates a structure within which we describe the world.

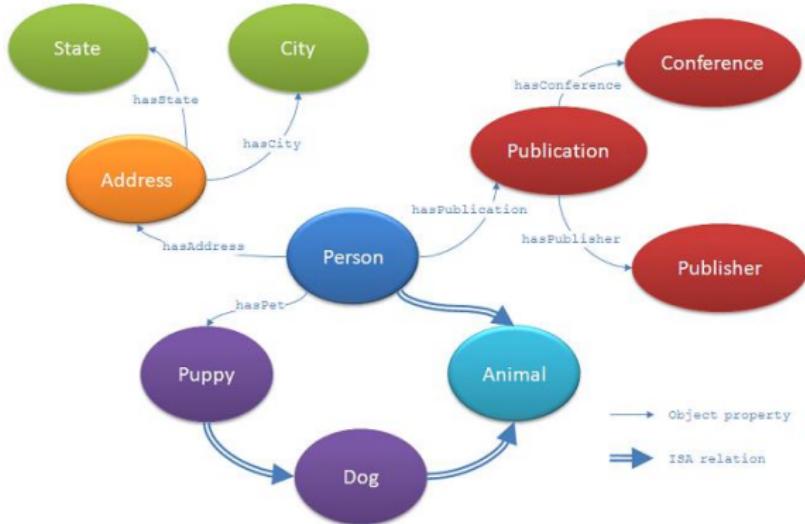


Figure: Triple Beispiel <http://nemo.nic.uoregon.edu/wiki/File:An>

Ein Beispiel

```
1 <Bob> <is a> <person> .  
2 <Bob> <is a friend of> <Alice> .  
3 <Bob> <is born on> <the 4th of July 1990> .  
4 <Bob> <is interested in> <the Mona Lisa> .  
5 <the Mona Lisa> <was created by>  
6 <Leonardo da Vinci>.  
7 <the video 'La Joconde à Washington'>  
8 <is about> <the Mona Lisa> .
```

Ein Beispiel II

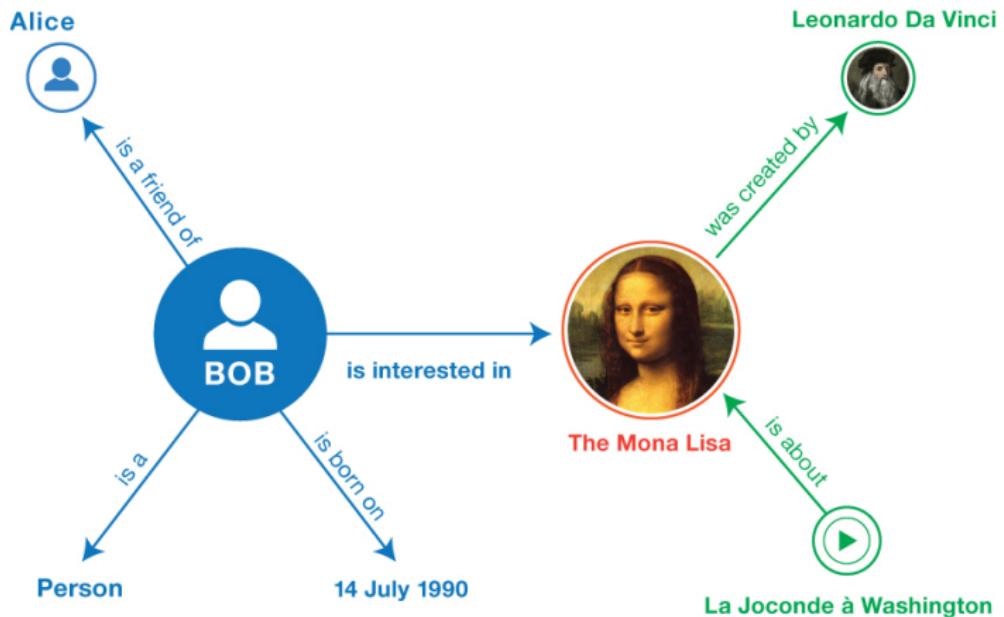


Figure: Mona Lisa Example (by W3C)

Wikidata Tutorial

- 
- 2 Wikidata Tutorial**
- **HandsOn Wikidata**
 - **HandsOn SPARQL**
 - **Wikidata in meiner Anwendung**

Was ist Wikidata?

- Vereinfacht: Eine maschinenlesbare Wikipedia!
- Man nehme die Fakten aus Wikipediaartikeln und stelle sie maschinenlesbar ALS TRIPLES bereit!
- Verlinke Fakten aus Wikipediaartikeln mit anderen Fakten, die bereits im Semantic Web verfügbar sind

Ein paar Zahlen.....

- **Mehr als 51 Millionen Fakten/Konzepte in Wikidata**
- **Englische Wikipedia hat nur 5 Millionen Artikel!**
- **Mehr als 5 Millionen georeferenzierte Wikidata Objekte**
- **Mehr als 1,2 Millionen verlinkte OpenStreetMap Objekte zu Wikidata**

Konzepte? Wie kennt die Maschine das?

- Konzepte haben URLs über die sie eindeutig identifiziert werden können
 - <https://www.wikidata.org/wiki/Q2> - Die Erde
 - <https://www.wikidata.org/wiki/Q316> - Liebe
- Instanzen von Konzepten haben auch URLs
 - <https://www.wikidata.org/entity/Q4176> Kölner Dom (ist eine katholische Kirche Q56242215)
 - <https://www.wikidata.org/entity/Q154797> Bundestag (ist ein Unterhaus Q375928)
- Beziehungen haben auch URLs
 - Albert Einstein (Q937) - sex or gender (P21) - male (Q6581097)
- URLs werden oft mit Prefixen abgekürzt. Hier wd: für Wikidata Konzepte, wdt: für Wikidata Properties
<http://www.prefix.cc>

Wow cool! Wie kann ich das nutzen?

- 1 Passende Wikipediaartikel finden**
- 2 Wikidata Daten dazu anzeigen lassen**
- 3 Wikidata selbst editieren!**
- 4 Verwandte Daten finden!**
- 5 Schauen wir hierzu im HandsOn!**

HandsOn Wikidata

HandsOn Wikidata

https://www.wikipedia.org/wiki/Albert_Einstein

Wikidata IDs

- Wikidata IDs fangen für Konzepte mit Q an
 - Q2=Die Erde
 - Konzepte können frei editiert werden
 - Für Beziehungen mit P
 - P31 - instanceOf
 - Findet heraus welche Beziehungen für euch wichtig sind in der Liste der Properties
 -
- https://www.wikidata.org/wiki/Wikidata:List_of_properties

Was kann ich in Wikidata anfragen?

Falsche Frage: Was kann ich NICHT in Wikidata anfragen?

Es gibt:

- **Personen**
- **Zeitgeschichte**
- **Geodaten**
- **Kategorisierungen**
- **Botanik**
- **Verlinkungen zu anderen Wissensbasen**
- **Weitaus mehr als Wikipedia bereitstellt**

Wie kann ich in Wikidata anfragen?

Einzelne Konzepte:

- Über eine REST API:
[https://www.wikidata.org/w/api.php?
action=wbgetentities&ids=Q24871](https://www.wikidata.org/w/api.php?action=wbgetentities&ids=Q24871)
- Mediawiki API: [https://www.wikidata.org/wiki/
Special:EntityData/Q42.json](https://www.wikidata.org/wiki/Special:EntityData/Q42.json)
- Database Dumps (aber möchte man das?)
- Es gibt bessere Methoden!

Wie kann ich Triple Stores anfragen?

- 1 Triple Stores haben APIs die angefragt werden können**
- 2 Entweder REST APIs oder spezielle SPARQL Endpoints**
- 3 SPARQL ist eine Sprache für die Anfrage von Triple Stores**

Triple Stores anfragen -> SPARQL

PREFIXES

```
2 SELECT ?waswollenwir WHERE {  
    BEDINGUNGEN  
    FILTERBEDINUNGEN  
    SERVICES  
6 } LIMITIERUNGEN
```

- Wir definieren Ergebnisse (Städte, Personen etc.)
- Wir definieren und den passenden Subgraph dazu durch Einschränkungen (Städte in Deutschland, Lebende Personen)
- Das testen wir nun in einem weiteren HandsOn

HandsOn SPARQL

HandsOn SPARQL

<https://query.wikidata.org>

Eine einfache Anfrage - Einsteins Kinder

```
1 SELECT ?child ?childLabel WHERE {  
2   ?child wdt:P22 wd:Q937.  
3   SERVICE wikibase:label {  
4     bd:serviceParam wikibase:language  
5     "[AUTO_LANGUAGE],en". }  
6 }
```

- wd:Q937 - Albert Einstein
- wdt:P22 - father

Eine einfache Anfrage - Einsteins Vater

```
1 SELECT ?father ?fatherLabel WHERE {  
2   wd:Q937 wdt:P22 ?father .  
3   SERVICE wikibase:label {  
4     bd:serviceParam wikibase:language  
5     "[AUTO_LANGUAGE],en" . }  
6 }
```

- **wd:Q937 - Albert Einstein**
- **wdt:P22 - father**

Eine einfache Anfrage - Katzen

```
1 SELECT ?item ?itemLabel
2 WHERE
3 {
4   ?item wdt:P31 wd:Q146.
5   SERVICE wikibase:label { bd:serviceParam
6     wikibase:language "[AUTO_LANGUAGE],en" . }
7 }
```

- **wd:Q146 - house cat**
- **wdt:P31 - instance of**

Eine einfache Anfrage - Katzen mit Bild

```
1 SELECT ?item ?itemLabel
2 WHERE {
3     ?item wdt:P31 wd:Q146.
4     ?item wdt:P18 ?pic .
5     SERVICE wikibase:label { bd:serviceParam
6         wikibase:language "[AUTO_LANGUAGE],en" . }
```

- **wd:Q146 - house cat**
- **wdt:P31 - instance of**
- **wdt:P18 - image**

Aggregationen - Anzahl der Menschen in Wikidata

```
1   SELECT (COUNT(?item) AS ?count)
2   WHERE {
3       ?item wdt:P31/wdt:P279* wd:Q5 .
4   }
```

- **wdt:P31 - instance of**
- **wdt:P279 - subclass of**
- **wd:Q5 - human**
- *** und sämtliche Unterklassen/Bäume**

Geokoordinaten - Nationalparks

```
1 SELECT ?objectLabel ?objectDescription ?link ?coord
2 WHERE {
3     ?object wdt:P31/wdt:P279? wd:Q46169 .
4     OPTIONAL { ?object wdt:P856 ?link . }
5     OPTIONAL { ?object wdt:P625 ?coord . }
6     SERVICE wikibase:label {
7         bd:serviceParam
8         wikibase:language "[AUTO_LANGUAGE],en,fr"
9     }
}
```

Federated Queries

```
1 SELECT ?item ?itemLabel ?gss ?os_toid WHERE {
2   ?item p:P836 ?gss_stmt .
3   ?gss_stmt ps:P836 ?gss .
4   FILTER NOT EXISTS { ?gss_stmt pq:P582 [] } .
5   FILTER NOT EXISTS { ?item wdt:P3120 [] } .
6
7   SERVICE <http://data.ordnancesurvey.co.uk/datasets
8   /os-linked-data/apis/sparql>
9   {
10   ?os_toid <http://data.ordnancesurvey.co.uk/ontology
11   /admingeo/gssCode> ?gss
12   }
13   SERVICE wikibase:label { bd:serviceParam wikibase:label
14 }
```

- Idee: Ich greife direkt im Query auf andere Triple Stores zu!

SPARQL Filterbedingungen

Es gibt einige Filterfunktionen, aber die wichtigste ist vermutlich die Filterung nach Strings. Ein Beispiel:

```
1 SELECT ?workflow
2 WHERE{
3     ?workflow rdf:type mecontrib:Workflow ;
4         mebase:has-content-type ?ct .
5     ?ct dcterms:title ?ct_title
6     FILTER regex(?ct_title , '^taverna' , 'i ')
7 }
```

Wenn das jetzt zu schnell ging....

SPARQL Tutorial

How To SPARQL:

<http://rdf.myexperiment.org/howtosparql>

Nutzung in Anwendungen - Wikidata Toolkit

- Eine Java Library um
 - Daten einzutragen
 - Daten anzufragen
 - Statistikqueries
- <https://github.com/Wikidata/Wikidata-Toolkit>

Beispiel:

```
1 WikibaseDataFetcher wbdf = new WikibaseDataFetcher(  
2   ApiConnection.getTestWikidataApiConnection(),  
3   Datamodel.SITE_WIKIDATA);  
  
4 System.out.println("*** Fetching data for one entity:");  
5 EntityDocument q42 = wbdf.getEntityDocument("P176");  
6 System.out.println(q42);
```

Nutzung in Anwendungen I - JavaScript

- Wikidata SDK - eine Library für Wikidata Access in JavaScript
- <https://github.com/maxlath/wikidata-sdk>

Beispiel:

```
authorQid = 'Q535'  
2 sparql = `  
SELECT ?work ?date WHERE {  
4     ?work wdt:P50 wd:${authorQid} .  
OPTIONAL {  
6         ?work wdt:P577 ?date .  
7     }  
8 }`  
url = wdk.sparqlQuery(sparql)
```

Nutzung in Anwendungen II - Java

- Apache Jena
- <https://jena.apache.org>
- RDF4J
- <http://rdf4j.org>

Nutzung in Anwendungen III - Android

- Apache Jena Ports für Android
- <https://github.com/sbrunk/jena-android>
- Zur Not tut es aber auch ein HttpClient der einen SPARQL Endpoint ansprechen kann

Wikidata ist toll - wie kann ich mitmachen?

- **Wikidata Items selbst anlegen!**
- **Andere Projekte z.B. OSM zu Wikidata verbinden**
<https://osm.wikidata.link>
- **Beteiligt euch in Arbeitsgruppen. Es gibt dafür Mailinglisten!**

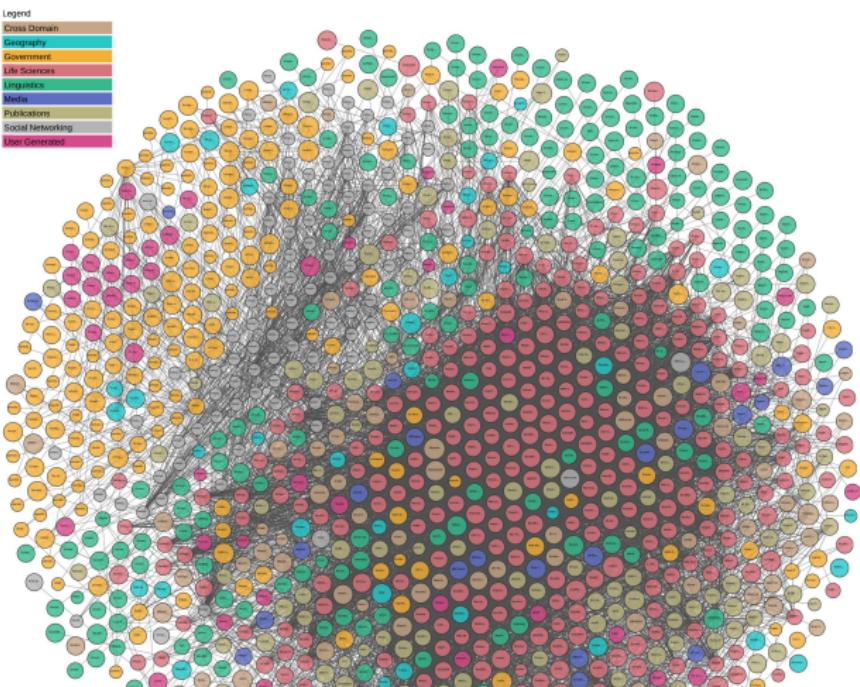
Weitere Repositorien

3 Weitere Repositorien

- **Linked Geodata**
- **MusicBrainz**
- **Linguistic Linked Open Data**

Die Linked Open Data Cloud

LOD Cloud <https://lod-cloud.net>



Weitere Repositorien

- **Die Linked Open Data Cloud ist riesig!**
- **Viele interessante Datenrepositorien sind vorhanden**
- **Es lohnt sich auch eigene erstellte Daten entweder**
 - In eine Repository z.B. Wikidata zu integrieren
 - Selbst zu hosten und Links zur LOD Cloud herzustellen

Geodaten

- Geodaten können sehr nützlich für CdV sein
- OpenStreetMap kann als Karte für Apps dienen
- Wikidata hat sehr häufig Geokoordinaten annotiert

Geodaten - LinkedGeoData

LinkedGeoData

Eine Ontologie/ein TripleStore über OpenStreetMap
Schauen wir mal rein: <http://linkedgeodata.org/About>

MusicBrainz

- **Triple Store für Musikalben, Interpreten und Genres**
- **<https://musicbrainz.org>**
- **[RDF Dump auf http://linkedbrainz.org](http://linkedbrainz.org)**

Linguistic Linked Open Data

- **Linked Open Data Wörterbücher**
- **Gut um Bezüge von Texten zu weiteren Objekten herzustellen**
- **Bildet Ähnlichkeiten und Bezüge zwischen Worten ab**
 - **Toponyme**
 - **Synonyme**
- **Nomen zu Konzept Beziehungen**
- **Verben zu Aktions Beziehungen**
- **Adjektive zu Konzepten existieren auch**
- **Zum Teil in NLP Ansätzen benutzt**

Wordnet Beispiel

WordNet Sub-Graph (English)

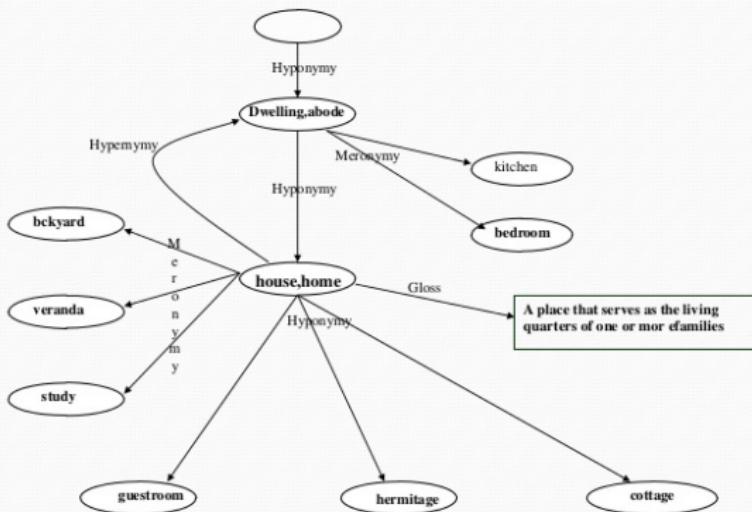


Figure: WordNet Example <https://www.slideshare.net/govindrai15/w>

Linguistic Linked Open Data II

- **WordNet (Englisch)** <https://wordnet.princeton.edu>
- **BabelNet (Multilingual)** <https://babelnet.org>
- **Semantic Dictionary for Ancient Languages**
[https://situx.github.io/Semantic Dictionary](https://situx.github.io/Semantic%20Dictionary)

Wikidata in anderen Wikimedia Projekten

- **Wiktionary setzt Links zu Wikidata**
- **Wikivoyage**
- **OpenStreetMap hat ca. 1,2 Mio Links zu Wikidata Objekten**

Zusammenfassung

4 Zusammenfassung

Zusammenfassung

Das Semantic Web bietet:

- **Strukturierte und kategorisierte Daten**
- **Einfach abrufbar**
- **Einfach in Anwendungen integrierbar**
- **Einfach modifizierbar**

Materialien der Vorstellung?

- Diese Präsentation
- Codebeispiele
- Nützliche Links

In meinem Github!

<http://github.com/situx/cdv2018>

Viel Erfolg!

Noch Fragen?

**Kommt gerne auf mich zu und wir klären Probleme!
Ansonsten viel Spaß bei Coding Da Vinci Rhein-Main!**

<http://github.com/situx/cdv2018>