FREIE UNIVERSITÄT BERLIN

Fachbereich Wirtschaftswissenschaft

**Diplomarbeit**

zur Erlangung des Grades

eines Diplom Volkswirtes

# Design of a Quality Assessment Framework for the DBpedia Knowledge Base

eingereicht bei Professor Dr. Christian Bizer
von cand. rer. pol. Paul Kreis Matr.-Nr.: 3948143
Anschrift: Schönhauser Allee 150, 10435 Berlin
Tel.: 015771412205
Berlin, den 28.02.2011

## Abstract

Information Quality is a significant aspect for the Web of Data and its future generation of applications. Until now it is neither possible to quantify the quality of the DBpedia knowledge base, nor to quantify the improvement that is made from one release to another. Within this study a Quality Assessment Framework (QAF) is developed to document the quality of the knowledge base and furthermore the upgrading of DBpedia's extraction framework. The main idea of this framework is a comparison between a manually created best-case dataset and the output from DBpedia's ontology based extraction.

The QAF estimates the precision of the extraction framework and the completeness of DBpedia compared to its source Wikipedia. In a first run, which evaluates the DBpedia 3.5.1 release, a completeness of 46 % and precision of 91.1 % was perceived and tasks to improve the extraction framework were detected. Before the DBpedia 3.6 release, some of the given tasks were implemented. After the 3.6 release, the evaluation was repeated with the improved extraction framework. The completeness increased to 61 % and the precision to 92.3 %. Nonetheless, there are many open tasks left that are detected during this study.

## Eigenständigkeitserklärung

Hiermit bestätige ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und andere als die angegebenen Hilfsmittel und Quellen nicht benutzt habe. Die Arbeit hat keiner anderen Prüfungsbehörde vorgelegen. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Berlin, den 28.02.2011

# Contents

**1. Introduction** **1**

**2. Background** **2**

2.1. Wikipedia . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.2. DBpedia Project . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2.3. Web of Data & the RDF Data Model . . . . . . . . . . . . . . . . . . . . . 6

**3. Quality Assessment Framework** **12**

3.1. Problem definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.2. Information Quality & relevant Quality Dimensions . . . . . . . . . . . . . 13

3.2.1. Intrinsic Dimensions . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.2.2. Contextual Dimensions . . . . . . . . . . . . . . . . . . . . . . . . 15

3.2.3. Representational Dimensions . . . . . . . . . . . . . . . . . . . . . 17

3.2.4. Accessibility Dimensions . . . . . . . . . . . . . . . . . . . . . . . 18

3.2.5. Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

3.3. Quality Assessment Metrics . . . . . . . . . . . . . . . . . . . . . . . . . 19

3.4. Gold Standard and Pattern Categories . . . . . . . . . . . . . . . . . . . . 21

3.4.1. Plain Property . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

3.4.2. Lists . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

3.4.3. One-Property Tables . . . . . . . . . . . . . . . . . . . . . . . . . 23

3.4.4. Multi-Property Tables . . . . . . . . . . . . . . . . . . . . . . . . 25

3.4.5. Coordinates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

3.4.6. Number-Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

3.4.7. Intervals . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3.4.8. Open Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.4.9. Open Property Tables . . . . . . . . . . . . . . . . . . . . . . . . 29

3.4.10. Internal Templates . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3.4.11. Merged Properties . . . . . . . . . . . . . . . . . . . . . . . . . . 31

**4. Experiment** **31**

4.1. Sample Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

4.2. Implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

4.3. User Instructions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

4.4. Evaluation Process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

4.4.1. Preprocessing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

4.4.2. Dataset Initialization . . . . . . . . . . . . . . . . . . . . . . . . . 40

4.4.3. Triple Equality Comparison (TEC) . . . . . . . . . . . . . . . . . . . 40
4.4.4. Predicate Neutral Comparison (PNC) . . . . . . . . . . . . . . . . . 41
4.4.5. Object Similarity Comparison (OSC) . . . . . . . . . . . . . . . . . . 42
4.4.6. Snippet Completeness Check . . . . . . . . . . . . . . . . . . . . . . 44

**5. Results 45**

5.1. Results for the Pattern Categories . . . . . . . . . . . . . . . . . . . . . . 46
5.2. Tasks for improving the DBpedia framework . . . . . . . . . . . . . . . . . 53
5.2.1. Combined infoboxes . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
5.2.2. Various infoboxes to one entity . . . . . . . . . . . . . . . . . . . . . 54
5.2.3. Entity recognition . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
5.2.4. Ontology Property Finder . . . . . . . . . . . . . . . . . . . . . . . . 56
5.2.5. Extraction of additional display and design templates . . . . . . . 57
5.2.6. Rendered page extraction to solve internal templates . . . . . . . . 57
5.2.7. Mapping extensions . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.3. Second roll after improvement . . . . . . . . . . . . . . . . . . . . . . . . 58

**6. Conclusion 60**

**References 62**

**A. Appendix 65**

A.1. How to read the Excel file . . . . . . . . . . . . . . . . . . . . . . . . . . 65
A.2. Sample Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
A.3. Completeness & Precision Statistics . . . . . . . . . . . . . . . . . . . . . . 69
A.4. Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77

# 1. Introduction

DBpedia is a project developed by two working groups of the Free University Berlin and the University of Leipzig. Within the context of *Linked Data*[1], the DBpedia project extracts structured information from Wikipedia and makes the resulting knowledge base publicly available on the web. Due to the technical difficulties by getting structured data from an unstructured source, the extraction process is not free of errors. There can be simple conversion problems by standardising units of measurement, or complex issues by the needed breakdown of tables into their elements. These are only two of many error sources.

This study is the first approach to an integrated assessment of the knowledge base quality and of possible error sources to be faced within the extraction. As new DBpedia versions are published regularly, it would be helpful to estimate the quality improvement of each new release. Therefore, this study introduce an assessment framework that is designed to evaluate the quality of the knowledge base concerning its completeness and precision in relation to Wikipedia.

The main idea of the Quality Assessment Framework (QAF) is a comparison of best-case and actual data. For a sample of Wikipedia articles, the best-case extraction results (gold standard) are manually created and compared with the actual extraction result. The gold standard has an important role. It is the upper quality bound, to which DBpedia will run. This upper bound is not fixed for the future. There might be new requirements to the data or mayor changes in Wikipedia's information supply, which would make it necessary to update the gold standard. Therefore, the QAF provides an user interface that makes it easier to create and edit the gold standard. Related to the properties and structures in which the extracted information are arranged in the Wikipedia articles, the information snippets in the gold standard are classified in *Pattern Categories* like lists or tables. That allows a fine-grained evaluation of the knowledge base. The gold standard created for this study is based on a Wikipedia dump generated on 24th of March 2010. The DBpedia 3.5.1 release is based on that dump and the main object of the analysis. Subsequently, the extraction result from the 3.5.1 extraction framework version is compared with the extraction framework version used for the DBpedia 3.6 release. This comparison allows the quantification of improvement of the DBpedia extraction framework.

Related work for the general topic of Information Quality (IQ) is given by Wang & Strong in [WS96, Wan98]. Naumann & Rolker identify criterion classes for IQ and give

[1] Linked Data describes a concept of connecting and publishing pieces of data. Further details about LD and the Web of Data can be found in Section 2.3.

detailed assessment methods for each criterion [NR00]. Pipino et al. [PLW02] describe principles that can help develop usable quality metrics. Bizer [Biz07] gives an overview about IQ assessment in the context of web-based systems. Finally, Javanmardi & Lopes [JL10] did some research about the content quality in Wikipedia articles.

This study is organised as follows: Section 2 deals with the related background, in which Section 2.1 introduce the online-encyclopedia Wikipedia, which is the base for the DBpedia project described in Section 2.2. In Section 2.3 an overview about the Web of Data and its technical background is given. Section 3 introduce the QAF in theory with an specification of the gold standard and its Pattern Categories in Section 3.4. The actual realisation of the QAF is described in the experiment Section 4 that includes the sample structure in Section 4.1, the implementation in Section 4.2, user instructions in Section 4.3 and the process of the evaluation with its metrics Section 4.4. Section 5 discuses the results for each Pattern Category and gives task to improve the extraction framework. Finally, Section 6 draws conclusions and summarise the results.

# 2. Background

## 2.1. Wikipedia

Wikipedia is an online-encyclopedia written from its users in a cooperative process. It describes itself as[2]:

> "[...] a free, web-based, collaborative, multilingual encyclopedia project supported by the non-profit Wikimedia Foundation. Its 17 million articles (over 3.5 million in English) have been written collaboratively by volunteers around the world, and almost all of its articles can be edited by anyone with access to the site[3]. Wikipedia was launched in 2001 by Jimmy Wales and Larry Sanger[Mil08] and has become the largest and most popular general reference work on the Internet[4] [Tan07, Woo07] [...]."

The easy and open way in which authors can publish information at Wikipedia, are two of the key concepts that lead to its success. With the *Wikipedia Markup*[5], users are able to create and edit wiki pages easily. The MediaWiki software converts the Wikipedia markup into the *HyperText Markup Language*[6] (HTML), which web browsers

[2] `http://en.wikipedia.org/wiki/Wikipedia/` (retrieved 31/01/2011).

[3] In some parts of the world, the access to Wikipedia had been blocked.

[4] `http://www.alexa.com/topsites/` (retrieved 31/01/2011).
`http://www.alexa.com/siteinfo/wikipedia.org/` (retrieved 31/01/2011).

[5] For more information, see `http://en.wikipedia.org/wiki/Wiki_markup/` (retrieved 31/01/2011).

[6] For more information, see `http://en.wikipedia.org/wiki/HTML/` (retrieved 31/01/2011).

can read and display as web pages. But, like browsers are lenient on defective HTML, the MediaWiki software is lenient on mistakes authors made in their Wikipedia markup. This tolerance is an accommodation to a higher usability. For a browser or the MediaWiki software it is at last about displaying the content. If that is done well, the quality of the underlying markup does not really matter. This leads to problems by an automatic extraction based on the markup. That is exactly what DBpedia's Extraction Framework does.

## 2.2. DBpedia Project

The DBpedia project has two mayor goals, which are pursued in a community effort. First, to extract structured information from Wikipedia. Second, to make the resulting data accessible on the Web [BLK$^+$09]. The extraction framework is maintained by working groups from the Free University Berlin[7] and the University of Leipzig[8]. The knowledge base is hosted by OpenLink Software[9]. DBpedia is an open source project, therefore all source code can be downloaded and modified locally. As DBpedia is derived from Wikipedia, it is distributed under the same licensing terms as Wikipedia itself. Since DBpedia 3.4 the data is licensed under the terms of the *Creative Commons Attribution-ShareAlike 3.0* license[10] and the *GNU Free Documentation License*[11]. The current DBpedia data set[12] describes more than 3.5 million things, including 364,000 persons, 462,000 places, 99,000 music albums, 54,000 films, 16,500 video games, 148,000 organizations, 148,000 species and 5,200 diseases. These 3.5 million entities have labels and abstracts in up to 97 languages. There are 1.85 million links to images, 5.9 million links to external web pages and 6.5 million external links to other knowledge bases. The project claims to be a crystallization point of the arising Web of Data [BLK$^+$09]. The big DBpedia knowledge base with its wide spreaded domains is a good initial point for this ambition. Consequently, many other datasets are linked to DBpedia. Currently, there are 16.48 million[13] outgoing links into other datasets, and 11.54 million links from other datasets to DBpedia. With a look at the *Linking Open Data* cloud diagram[14], one can acclaim the importance of DBpedia in the current state. Figure 10 shows the diagram, it

[7] `http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/index.html` (retrieved 01/02/2011).

[8] `http://aksw.org/About/` (retrieved 01/02/2011).

[9] `http://www.openlinksw.com/` (retrieved 01/02/2011).

[10] `http://creativecommons.org/licenses/by-sa/3.0/legalcode` (retrieved 04/02/2011).

[11] `http://www.gnu.org/licenses/fdl.html` (retrieved 04/02/2011).

[12] DBpedia 3.6 was released at January 17, 2011.

[13] DBpedia 3.5.1 version. For details see `http://ckan.net/package/dbpedia/` (retrieved 25/01/2011).

[14] The LOD cloud "[...] shows datasets that have been published in Linked Data format, by contributors to the Linking Open Data community project and other individuals and organisations." - `http://richard.cyganiak.de/2007/10/lod/`

can be found in the Appendix. For each extracted entity, DBpedia defines a global unique identifier that is dereferenceable according to the Linked Data principles[BL06, BHBL09]. Section 2.3 takes a closer look at these principles and the whole idea of the Web of Data.

Wikipedia articles do not only consist of free text, additionally they contain different kinds of templates, links to other language versions or categorisation information. These data are the entry point for the DBpedia framework to extract structured information. Particularly suitable for this task are the infobox templates. These are standardised boxes placed on the right side of many Wikipedia articles, which contain simple facts about the described entity. Figure 1 shows the Wikipedia article of Elvis Presley. The infobox is marked as A and the different language links are marked as C. Because all infoboxes are grown out of Wikipedias community driven development, there are multiple infoboxes used for the same type. For instance, we found five infoboxes describing cricket players. There are the *Infobox recent cricketer*, *infobox Old Cricketer*, *Infobox historic cricketer*, *Infobox cricketer* and the *Cricketer Infobox*. In the meanwhile the Wikipedia community has merged these infoboxes to the *Infobox cricketer*, but there are more of these examples. The attribute names of all cricketer infoboxes could differ from each other, but the meaning could be synonymously. For instance, each of these cricketer infoboxes has an attribute for the birth date of the cricketer. But the attribute names of these properties could differ from each infobox. For example, the *Cricketer Infobox* could have the property *born*. On the contrary, the *infobox Old Cricketer* could have the property *dateOfBirth*. Both properties describe the date of the cricketers birth. In a consistent knowledge base, each person should have the same property describing its birth date. To overcome the problems of synonymous attribute names and multiple infoboxes being used for the same type of things, the Wikipedia templates are mapped to an ontology. Each infobox property can be mapped to an DBpedia ontology property. It is possible to define fine-grained rules, which determine how an infobox value must be parsed and define target data types. For instance, if a mapping defines a target data type to be a geo-coordinate, the parser searches for a coordinate pattern and ignores additional annotations that might be present in the property value. The participating DBpedia community can write these mappings and keep them up to date. On the date of the DBpedia 3.6 release, the ontology holds 272 classes with 629 object properties and 706 data type properties. The mapping statistics[15] were:

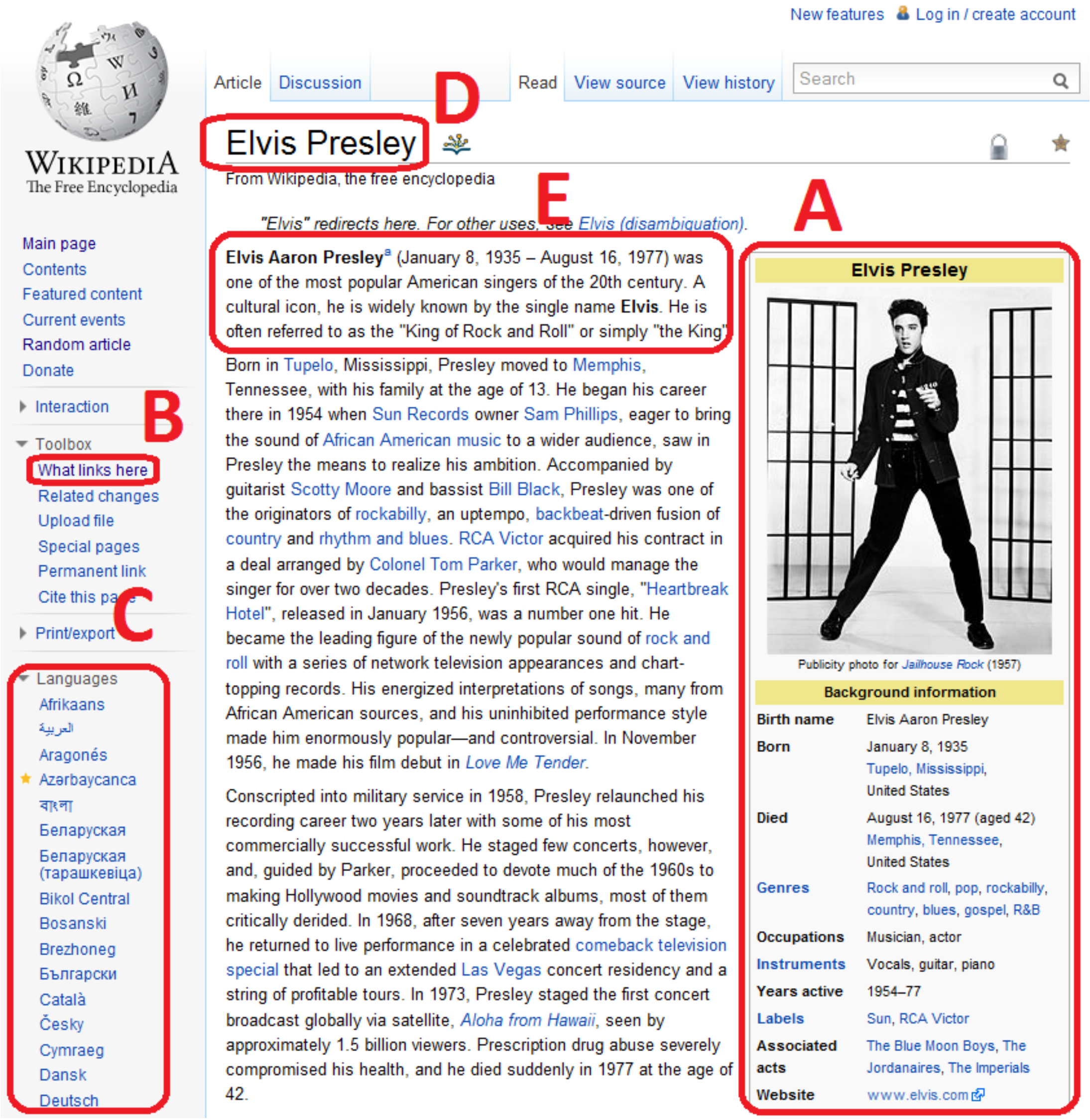
- 315 English infobox mappings, which covers 1124 templates including redirects[16].

[15] `http://blog.dbpedia.org/2011/01/17/dbpedia-36-released/` (retrieved 2011/02/07).

[16] There are many infoboxes that are just redirects to other infoboxes. For instance, the infobox president redirects to the infobox officeholder. Thus, the infobox president uses the attributes of the infobox officeholder and the name of the infobox is merely used for categorisation issues.

- 137 Greek infobox mappings, which covers 192 templates including redirects.
- 111 Hungarian infobox mappings, which covers 151 templates including redirects.
- 36 Croatian infobox mappings, which covers 67 templates including redirects.
- 9 German infobox mappings.
- 4 Slovenian infobox mappings.

Figure 1: Wikipedia page of Elvis Presley



Elvis Presley

From Wikipedia, the free encyclopedia

*"Elvis" redirects here. For other uses, see Elvis (disambiguation).*

**Elvis Aaron Presley**[a] (January 8, 1935 – August 16, 1977) was one of the most popular American singers of the 20th century. A cultural icon, he is widely known by the single name **Elvis**. He is often referred to as the "King of Rock and Roll" or simply "the King".

Born in Tupelo, Mississippi, Presley moved to Memphis, Tennessee, with his family at the age of 13. He began his career there in 1954 when Sun Records owner Sam Phillips, eager to bring the sound of African American music to a wider audience, saw in Presley the means to realize his ambition. Accompanied by guitarist Scotty Moore and bassist Bill Black, Presley was one of the originators of rockabilly, an uptempo, backbeat-driven fusion of country and rhythm and blues. RCA Victor acquired his contract in a deal arranged by Colonel Tom Parker, who would manage the singer for over two decades. Presley's first RCA single, "Heartbreak Hotel", released in January 1956, was a number one hit. He became the leading figure of the newly popular sound of rock and roll with a series of network television appearances and chart-topping records. His energized interpretations of songs, many from African American sources, and his uninhibited performance style made him enormously popular—and controversial. In November 1956, he made his film debut in *Love Me Tender*.

Conscripted into military service in 1958, Presley relaunched his recording career two years later with some of his most commercially successful work. He staged few concerts, however, and, guided by Parker, proceeded to devote much of the 1960s to making Hollywood movies and soundtrack albums, most of them critically derided. In 1968, after seven years away from the stage, he returned to live performance in a celebrated comeback television special that led to an extended Las Vegas concert residency and a string of profitable tours. In 1973, Presley staged the first concert broadcast globally via satellite, *Aloha from Hawaii*, seen by approximately 1.5 billion viewers. Prescription drug abuse severely compromised his health, and he died suddenly in 1977 at the age of 42.

| Elvis Presley | |
|---|---|
| Background information | |
| Birth name | Elvis Aaron Presley |
| Born | January 8, 1935<br>Tupelo, Mississippi,<br>United States |
| Died | August 16, 1977 (aged 42)<br>Memphis, Tennessee,<br>United States |
| Genres | Rock and roll, pop, rockabilly, country, blues, gospel, R&B |
| Occupations | Musician, actor |
| Instruments | Vocals, guitar, piano |
| Years active | 1954–77 |
| Labels | Sun, RCA Victor |
| Associated acts | The Blue Moon Boys, The Jordanaires, The Imperials |
| Website | www.elvis.com |

## 2.3. Web of Data & the RDF Data Model

The amount of information we can find in the World Wide Web is growing exponentially [GCM$^+$08]. This makes it harder to find information we are looking for. Search engines like Google or Yahoo crawl the whole Web and provide a text-based search function for it. Mostly, the results of these search engines are a large quantity of documents, in which the user has to search again for explicit information he wants. In rare cases the used keywords were to restrictive and the search engine cannot deliver any useful documents. Another disadvantage of these text-based searches is that the semantics of the keywords are not considered. A user who searches for the keyword "apple" would find many documents to the company and a few documents which deal with the fruit. The user could use more specific keywords like "big apple", "granny smith apple", "apple Beatles" or "apple company" depending on what he is searching. But this implies that the user knows additional information about his searched item and especially that he already knows what he is searching for. The current web of documents is like a huge library with many catalogue drawers in the entrance. If one ask the librarian for some information, he just points to a drawer with thousands of index cards. One has to flip through the cards for an overview and walk down the aisles of bookshelves for detailed information. In addition to the manual work, the possibility exists that the index cards are outdated or the catalogue drawer incomplete, even though they are generated automatically.

In the existing Web, published documents are mostly written in HTML. This documents are linked to other documents that handle related topics. Machines used in this Web do not "understand" the meaning of this documents, they merely display the content or recognise the relations to other documents via the manually set links. The emerging Web of Data is an enhancement of the current Web of Documents. It is the vision of structured data that is interlinked and therefore becomes more useful. Effectively, the Web of Data extends the existing Web, by allowing a machine-readable description and interlinking of its content. Computers are much better enabled to process and "understand" the information in this Semantic Web. Related to the library example above, in the "Library's Web of Data" the content of all books would be classified and linked to each other. If a new book is taken into the library, the catalogue drawers must not be updated. The author of the book has already linked its content to other sources, or he already uses data of other sources in his book. In that case the book will update itself, if the external source updates its data. The librarian would be able to answer a precise question, like "I need all commanders of historic battles in the area of Ukraine". Because he has access to the facts in the books, it is not necessary to search each book about historic battles in Europe. All books together could act like a huge knowledge base.

The *World Wide Web Consortium* (W3C) is the international standards organization for the World Wide Web. The W3Cs Semantic Web Project proposes concepts and technologies for the realisation of such a Web of Data. Tim Berners-Lee, the director of the W3C, has coined the term *Linked Data* in a design note discussing issues[BL06] around the W3Cs Semantic Web Project. "Linked Data refers to a set of best practices for publishing and connecting structured data on the Web"[BHBL09, p.1]. Berners-Lee points out four rules for publishing Linked Data[BL06]:

1. "Use URIs as names for things.
2. "Use HTTP URIs so that people can look up those names."
3. "When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL) "
4. "Include links to other URIs, so that they can discover more things."

Berners-Lee himself states these four points as rules, but also notes that they are "expectations of behavior". Breaking them does not destroy the Web of Data, but the full potential would be restricted. Those rules are merely the same that lead to the success of the Web of Documents.

The first rule says, that things should be named with *Uniform Resource Identifiers* (URIs). A URI is a string of characters used for a definite identification of something. In case of a book, the ISBN would be a good URI. Related to the Web of Documents, this rule says that each document should be named by an *Uniform Resource Locator* (URL). A URL is a URI which gives additional information about the location of the document.

The second rule says, that used URIs should fulfil the standards of the HyperText Transfer Protocol. HTTP is the standard protocol for communication between webservers and web-clients. An example URI for a book could be ***http://example.org/0451524934***.

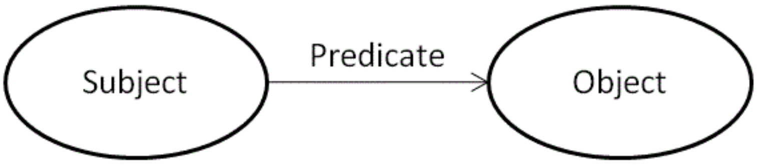
To fulfil the third rule, one must provide information about the URI if someone dereference it, thus for instance look it up in a web browser. The URI ***http://dbpedia.org/resource/Nineteen_Eighty-Four*** represents the novel 1984 by George Orwell. If one look it up in a browser, one will get a generated document with information about the novel from the DBpedia knowledge base. The relations between resources, identified by URIs, are described with the Resource Description Framework (RDF). SPARQL is an RDF query language and its name is an acronym that stands for **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage. RDF is further described below.

The fourth rule it important for weaving the web. It says that URIs should link to other URIs. This relations are expressed with RDF. The novel 1984 links to its author George

Orwell. In DBpedia he is described by the URI `http://dbpedia.org/resource/George_Orwell`. This is a link inside the DBpedia knowledge base. What really tighten the Web of Data, are links to external data sets. For instance the link from the DBpedia URI to the Freebase[17] URI describing the same book: `http://rdf.freebase.com/ns/m/0lz9s`. A short introduction how relations of URIs are described with RDF is given next.

**The Resource Description Framework** is a data model for representing information about resources in the WWW. RDF is developed for processing or exchanging information by applications [MMM04]. Its basic idea is to describe resources with properties and property values. Together, resource, property and its value build a statement. Related to the distinction between data and information, the statement could be seen as one information, build from three data items. Another explanation approach for describing resources with RDF comes from the sentence grammar. A statement or RDF-triple has three parts: subject, predicate and object. The subject is the resource we want to describe. The predicate is the property or attribute. Subject and predicate are always resources identified by URIs, though predicates are special resources, which explain the relation between subject and object. For instance the predicate "author" of the novel 1984. In DBpedia the author property is identified by the URI `http://dbpedia.org/ontology/author`. The object of an triple could be another resource, or a literal. Literals are strings of characters used to describe text, numbers or dates. There are plain literals, which are strings combined with an optional language tag. They are used for text in a natural language [MMM04]. And then, there are typed literals, which are strings combined with a data type URI. The data type URI defines the value space of the literal, for instance "years" or "metres". The subject-predicate-object relation can be illustrated by a node and directed-arc diagram, in with subject and object are nodes and the predicate as an arc that denotes the relationship.

Figure 2: Simple RDF Graph



[17] Freebase is commercial knowledge base retrieving data from different sources: `http://www.freebase.com/`.

RDF has two common syntax formats. The first is a XML[18] based notation, simply named RDF/XML. The information: "The author of 1984 is George Orwell" expressed in RDF/XML is

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdf="http://dbpedia.org/" >
  <rdf:Description rdf:about="dbpedia:resource/Nineteen_Eighty-Four">
   <dbpedia:ontology/author>
     <rdf:Description rdf:about="dbpedia:resource/George_Orwell">
     </rdf:Description>
   </dbpedia:ontology/author>
  </rdf:Description>
</rdf:RDF>
```

The first line includes the standard XML tag. The second to fourth line includes the RDF tag, which is the root element of the XML document and two XML namespace definitions. In the third line, the namespace "rdf" is defined as the URI `http://www.w3.org/1999/02/22-rdf-syntax-ns#` and in the fourth line the "dbpedia" namespace is defined as the `http://dbpedia.org/`. In the fifth line, the description tag for the resource `http://dbpedia.org/resource/Nineteen_Eighty-Four` is opened and in the sixth line the author property is added to the description of "Nineteen_Eighty-Four". The author property holds another rdf description about George Orwell. This description tag now, could include properties of George Orwell, for instance a literal with his name or his birth date.

The second common syntax is the N-Triples format. It is much easier to read and a strict triple notation. Our example would look like this:

```
<http://dbpedia.org/resource/Nineteen_Eighty-Four> <http://dbpedia.org/ontology/author>
   <http://dbpedia.org/resource/George_Orwell> .
```

Tis study uses the n-triple syntax for examples of RDF triples, because of its simplicity. The triples URIs will be shortend, therefore they can fit in one line. Thus the example above will look like:

```
<res/Nineteen_Eighty-Four> <ont/author> <res/George_Orwell> .
```

To give a coherent account of linked data a few more triples were added to the example.

[18] XML - eXtensible Markup Language, see W3C Recommendation `http://www.w3.org/TR/2000/REC-xml-20001006/` (retrieved 14/02/2011).

```
<res/Nineteen_Eighty-Four> <http://xmlns.com/foaf/0.1/name> “Nineteen Eighty-Four”@en .
<res/Nineteen_Eighty-Four> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <ont/Book> .
<res/Nineteen_Eighty-Four> <ont/author> <res/George_Orwell> .
<res/George_Orwell> <http://xmlns.com/foaf/0.1/name> “George Orwell”@en .
<res/George_Orwell> <ont/birthDate> “1903-06-25”^^<http://www.w3.org/2001/XMLSchema#date> .
<res/George_Orwell> <ont/birthPlace> <res/Bihar> .
<res/Bihar> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <res/PopulatedPlace> .
<res/Bihar> <ont/spokenLanguage> <res/Bengali_language> .
```

First, we give the resource describing the novel a name property and a type property. For this purpose, we revert to external ontologies. For the name property we use the FOAF[19] vocabulary and for the type property we use the standard RDF vocabulary. By using already defined external vocabularies we avoid to define the used properties and we contribute to building a standard for describing special types of entities. The birth date of George Orwell is a typed literal combined with the data type URI `http://www.w3.org/2001/XMLSchema#date`. Thus, we use the XML Schema data types[20] for defining dates. The values of the name properties are plain literals combined with a “@en”, which is the English language tag. Illustrating the triples above as graph model, shown in Figure 3, can denote the structure of a Web of Data.

An important aspect for this study is the concept of *Intermediate Nodes*. Those are necessary, because information are to complex that one can describe them with a simple subject-predicate-object statement. Intermediate nodes are pseudo resources, which represent aggregate concepts. In the DBpedia knowledge base intermediate nodes are merely artifacts from the extraction. They does not have an underlying Wikipedia article. The following *Sesame Street* example will illustrate this matter. In the Unites States, the television series Sesame Street is running on the *Public Broadcasting Service* (PBS). In Germany, the series is running on the network *Norddeutscher Rundfunk* (NDR). Sesame Street is the subject that is described. Its property can named as “is running on network”. But the property value, the object, is not only a simple resource, it is specified by the country. Therefore an intermediate node is used as object, which aggregates the network and the country. The n-triple notation for the example is:

```
<res/Sesame_Street> <ont/network> <res/Sesame_Street__network_1> .
<res/Sesame_Street__network_1> <ont/country> <res/United_States> .
<res/Sesame_Street__network_1> <ont/organisation> <res/Public_Broadcasting_Service> .
<res/Sesame_Street> <ont/network> <res/Sesame_Street__network_2> .
<res/Sesame_Street__network_2> <ont/country> <res/Germany> .
<res/Sesame_Street__network_2> <ont/organisation> <res/Norddeutscher_Rundfunk> .
```

[19] The Friend of a Friend project creates a vocabulary for describing people. For more information see `http://www.foaf-project.org/` (retrieved 14/02/2011).

[20] For further information, see `http://www.w3.org/TR/xmlschema-2/` (retrieved 14/02/2011).

Figure 3: Extended RDF Graph

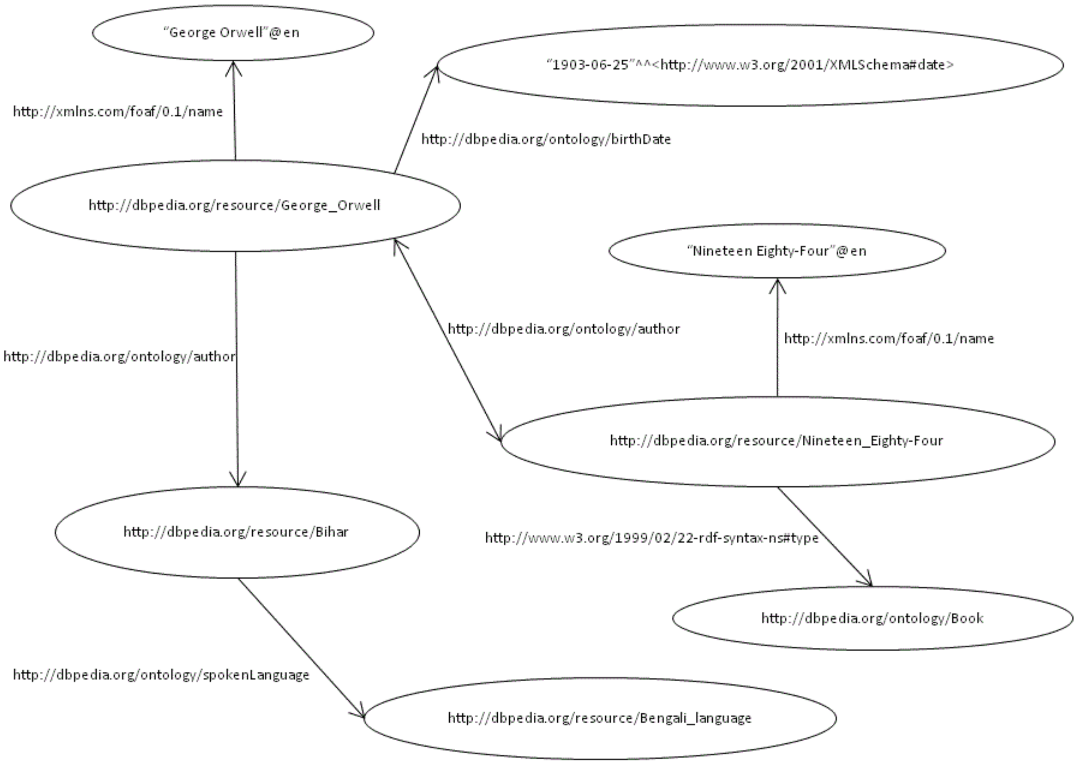
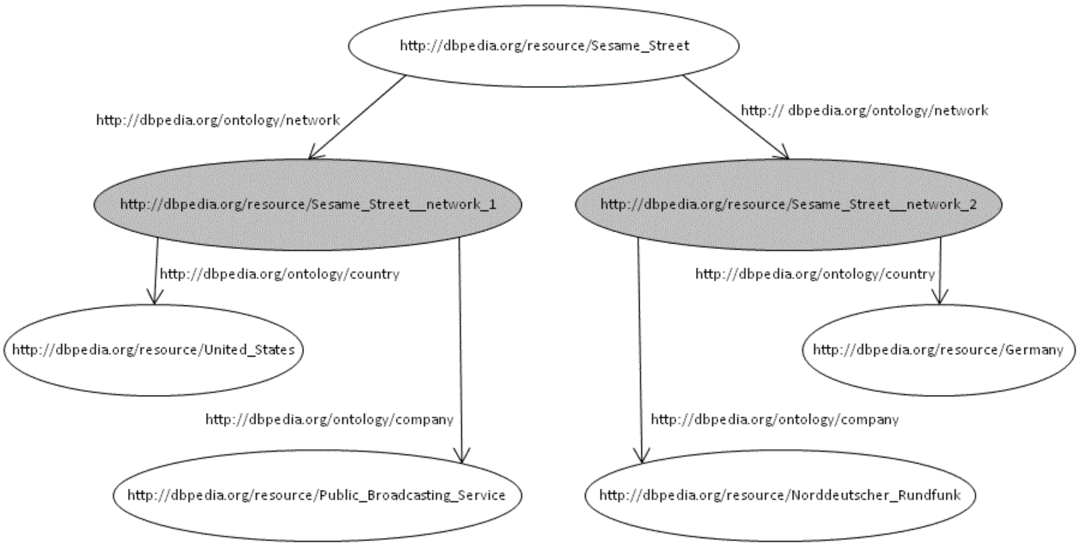

"George Orwell"@en
"1903-06-25"^^<http://www.w3.org/2001/XMLSchema#date>
http://xmlns.com/foaf/0.1/name
http://dbpedia.org/ontology/birthDate
http://dbpedia.org/resource/George_Orwell
"Nineteen Eighty-Four"@en
http://dbpedia.org/ontology/author
http://xmlns.com/foaf/0.1/name
http://dbpedia.org/ontology/author
http://dbpedia.org/resource/Nineteen_Eighty-Four
http://dbpedia.org/resource/Bihar
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://dbpedia.org/ontology/Book
http://dbpedia.org/ontology/spokenLanguage
http://dbpedia.org/resource/Bengali_language

Figure 4: Graph with Intermediate Nodes


http://dbpedia.org/resource/Sesame_Street
http://dbpedia.org/ontology/network
http:// dbpedia.org/ontology/network
http://dbpedia.org/resource/Sesame_Street__network_1
http://dbpedia.org/resource/Sesame_Street__network_2
http://dbpedia.org/ontology/country
http://dbpedia.org/ontology/country
http://dbpedia.org/resource/United_States
http://dbpedia.org/resource/Germany
http://dbpedia.org/ontology/company
http://dbpedia.org/ontology/company
http://dbpedia.org/resource/Public_Broadcasting_Service
http://dbpedia.org/resource/Norddeutscher_Rundfunk

Here, the URI `http://dbpedia.org/resource/Sesame_Street__network_1` identifies[21] the intermediate node that aggregates on which TV network, in the US, Sesame Street is running. The graph model is shown in Figure 4.

# 3. Quality Assessment Framework

The DBpedia Quality Assessment Framework (QAF) is the first approach to an integrated quality evaluation of the DBpedia knowledge base. Due to the source of the knowledge base, the information quality of Wikipedia is highly responsible for DBpedia's. This work takes Wikipedia's information quality as given and focuses on analysing DBpedia as a representation of it. The QAF simply compares two data sets. The first is a best-case data set, or *Gold Standard*, manually created from a Wikipedia dump. It contains all RDF triples of an Wikipedia article that can be extracted from it in the best-case. The gold standard is introduced in 3.4. The second data set is DBpedia's ontology-based extraction result. It covers the facts from Wikipedia and is the part of the knowledge base that is analysed by this study. DBpedia has other important extractors, for instance the abstract extractor or the generic extractor. But the article abstracts are very well in the meanwhile and the generic extractor uses the same parsers like the ontology-based

[21] Intermediate nodes do not require a "universal" identifier, because there is no need to refer to them directly from outside a particular graph [MMM04]. Intermediate nodes with identifiers that are not universal, like just numbers, are called *Blank Nodes*. For further information, see [MMM04] and `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#dfn-blank-node` (retrieved 16/02/2011).

extractor do. Therefore, we only focus on the ontology-based extraction results.

The difficulties that DBpedia's extraction framework faces are described in Section 3.1. A short introduction to IQ and its relevant aspects for this study are given in Section 3.2. In Section 3.3 the used Quality Assessment Metrics are introduced.

## 3.1. Problem definition

Since it never was the aim of Wikipedia to hold structured information that can be queried like a database, all the information on a Wikipedia page are saved in the markup that belongs to the page[22]. The DBpedia extraction has to struggle with the unclean markup and the problem about getting structured data from an unstructured source. The entry point to structured data are the infoboxes used on many Wikipedia pages, which contain properties for the most common domains. Defined mappings sort the infobox properties to the DBpedia Ontology properties and give rules how to extract them. The DBpedia community is able to edit the mappings. This extraction process is prone to errors that can arise from simple mistakes by converting a unit into another, to mayor errors with wrong designed mappings. During the development of the DBpedia Extraction Framework, it's quality was evaluated with simple test scripts and manually inspection of the resulting triples. After each new DBpedia release, many feedback e-mails arrive from the community about hints to errors or strange looking triples. Therefore, it was time to put some effort to a general overview of the knowledge base quality. It would be good to know how much the framework extracts from the whole Wikipedia and what can be done to increase the quality and the amount of data. Furthermore, there is a need for a benchmark to check whether bug-fixes or mayor updates really improve the quality.

## 3.2. Information Quality & relevant Quality Dimensions

First it is to say, that data quality and information quality are often used synonymously in literature. As there is a distinction between data and information, one could insist upon a distinction between the quality concepts. Data are raw, unanalysed facts, and information is useful knowledge that can be derived from it [Bod05]. Since the DBpedia knowledge base holds RDF triples that just embody the semantic between entities, this study will use data interchangeably with information.

Joseph Juran, who was an esteemed author about quality management, defines quality really short as "fitness for use" [Jur74]. This definition has stayed alive ever since and it implies two important aspects [Biz07]:

[22] There are, rarely used templates that integrate data from other pages. See below at 3.4.10.

- First, the quality of information is task-dependent. A bit of information that is useful for one task might be useless for another.
- Second, information quality is subjective. Two users with different demands to quality, may assess the quality of the same piece of information differently although they work on the same task.

Consequentially, different quality aspects are given. In literature these aspects are named dimensions. Richard Wang and Diane Strong introduce a broad catalogue of dimensions of data quality that are important to information users [WS96]. They define a data quality dimension as *"a set of data quality attributes that represent a single aspect or construct of data quality"* [WS96, p.6]. Since their introduction, this catalogue, sometimes extended or modified, is used in related literature [BWPT98, Wan98, Biz07]. This study sticks to the summarised catalogue from Christian Bizer [Biz07]. The quality dimensions are grouped in four categories: Intrinsic Dimensions, Contextual Dimensions, Representational Dimensions and Accessibility Dimensions.

It is to consider that many quality dimensions of DBpedia's knowledge base are bound to the information quality of Wikipedia. There are discussions[23] and researches about the quality of Wikipedia [JL10]. Related to *Condorcet's Jury Theorem*[24] and the *Wisdom of the Crowd* effect[25], a positive relation between the number of authors and edits of an article and its quality can be identified[Sur04, AMP06].

The following sections give definitions from literature for each quality dimension and notes an Wikipedia related example for it. In the end of each dimension group the relevant aspects for this study are pointed out.

### 3.2.1. Intrinsic Dimensions

Intrinsic Dimensions are task-independent. They cover whether the data describe the real world correctly and whether it is consistent in itself [Biz07].

**Accuracy** is the degree of correctness and precision with which data is representing its true state of the real world [WW96, Biz07]. It is an objective dimension, so it can be relevant only for facts. In Wikipedia, facts are mostly summarised in an article's infobox. The accuracy of Wikipedia's facts is checked by the Wikipedia community.

[23] A famous example is about a flawed study of the science journal Nature [Gil05, Enc06, Nat06].

[24] "Condorcet's jury theorem is a political science theorem about the relative probability of a given group of individuals arriving at a correct decision." - Wikipedia: Condorcet's jury theorem (19/01/2011).

[25] "The wisdom of the crowd refers to the process of taking into account the collective opinion of a group of individuals rather than a single expert to answer a question." Wikipedia: Wisdom of the crowd (19/01/2011).

**Timeliness** is the degree to which information are up-to-date [PLW02]. Here, an exception is made: It could also be relevant for the information users task. For example, if someone wants to decide which stocks he wants to buy, he should not count on the finance values of a company's Wikipedia article. But if he wants to study European rivers, information on Wikipedia should be up-to-date. The timeliness of an Wikipedia article depends mostly on the popularity of its topic and consequentially on the involvement of editors.

**Consistency** means that a set of information does not contain a contradiction [MSV$^+$02]. Due to the reason of Wikipedia's multiple authors, information can be inconsistent. A big leak of consistency can be found between the different language versions of the same article. For instance, the population of Germany can deviate between the German and the English Wikipedia article.

**Objectivity** "is the extent to which information is unbiased, unprejudiced, and impartial" [PLW02, p.212]. It should be obvious that Wikipedia cannot acclaim real objectivity for its articles, because of multiple authors and their subjective opinions. But just because of different opinions, the articles cannot be convicted as subjective per se.

Accuracy and objectivity of the knowledge base data are given by Wikipedia's content. But keeping the goal of the DBpedia Project to represent Wikipedia's information in mind, there is a accuracy aspect that should be considered. It is to measure whether the information is extracted correctly. That means, even if a fact is wrong it has to be adopted in correct way. This is the precision of the extraction framework.

Timeliness for DBpedia matters in the aspect of how often a new DBpedia version should be released. At any time in the future DBpedia will be updated live. Thus, this study do not discuss the aspect of timeliness.

The consistency dimension matters especially by cross language comparison of a Wikipedia article. Here, we stick only to the English version, because at this moment it is the only one with a relevant amount of defined mappings. The inconsistency between the language versions will be topic in another study.

### 3.2.2. Contextual Dimensions

Contextual Dimensions are task related. They cover the individual demands that are required for the data user's task.

**Relevancy** "is the extent to which information is applicable and helpful for the task at hand" [PLW02, p.212]. Coming back to a user studying European rivers. For a study about pollution, Wikipedia does not contain the relevant information. But it is relevant for information about the lengths or tributaries of rivers.

**Completeness** “is the extent to which data is not missing and is of sufficient breadth and depth for the task at hand”[PLW02, p.212]. Pipino et al. [PLW02] note three different aspects of completeness. First, *schema completeness*, which is the degree to which attributes and entities are not missing from the schema. Second, *column completeness*, which can defined as a function of missing values in a column of a table. This is related to Edgar Codd’s column integrity constraint, which disallows the occurrence of missing values in a specified column of a relational database [Cod90]. Third, *population completeness*, which refers to the ratio of entities in a information system to the complete population [Biz07]. Related to the example about rivers, it is necessary that Wikipedia contains all European rivers (population completeness). All infoboxes of these rivers must contain the length property (schema completeness) and each length property must contain a numeric value (column completeness). Only if all these completeness aspects are fulfilled, it becomes possible to make a statement about the longest river of Europe.

**Amount of Data** “is the extent to which the volume of data is appropriate for the task at hand” [PLW02, p.212]. The quantity of used data should not be to less nor to much. It must hold enough information to get good results, but should not keep to much details that are not necessary for the users task.

**Understandability** “is the extent to which the data is easily comprehended”[PLW02, p.212]. It is not only task-dependent but also user-dependent. The producer of the data might easily understand it, but the user could fail the intended meaning of the data [JBM08]. Therefore understandability also depends on a good documentation of data. A user of Wikipedia has not only the facts of the infobox, but also the text of the article that puts the facts into a comprehensive context. Referred to the distinction between data and information that is mentioned above: a Wikipedia article contains both.

**Believability** “is the extend to which data is regarded as true and credible” [PLW02, p.212]. Information in Wikipedia should be believed with care, because of the anonymity of its producers and the possibility of vandalism. For the used example of European rivers Wikipedia might be trustworthy enough: Because of the references in the articles, the irrelevance for manipulating the length of a river and the simplicity of detecting manipulation by comparing official information.

**Verifiability** “is the degree and ease with which the information can be checked for correctness” [Nau02, Biz07, p.18]. As long as Wikipedia pages have good references, the content is manually verifiable.

The DBpedia knowledge base provides information from many different domains as linked data. A user has to evaluate for himself whether our knowledge base is relevant for his special task. To increase the knowledge base’s relevancy for many different tasks, it should contain the maximum of knowledge that we can extract from Wikipedia.

The maximum information completeness of the knowledge base is given through the completeness of Wikipedia. If not all rivers in Germany have a Wikipedia page, the knowledge base cannot hold all German rivers. For this study, the completeness of Wikipedia will be the benchmark for DBpedia’s completeness.

To control the amount of data, the knowledge base is split in different data sets[26]. From these, the user is able to chose the information detail level for his task at hand.

The understandability of the knowledge base is given by the principles of Linked Data[27]. Therefore the understandability is not an aspect of this study.

The dimension of believability depends on Wikipedia too. It is a sensitive issue and for instance discussed in [Che05].

### 3.2.3. Representational Dimensions

The representational dimensions cover the importance of representing data within information systems for its quality.

**Representational Consistency** “is the extent to which information is represented in the same format” [PLW02, p.212]. Wikipedia’s content is represented as plain text, rather as HTML or Wikipedia markup language.

**Representational Conciseness** “is the extent to which information is compactly represented” [PLW02, p.212]. Referred to a Wikipedia article, it holds the broadly phrased text and the infoboxes with simple facts.

**Interpretability** “is the extent to which information is in appropriate languages, symbols, and units, and the definitions are clear” [PLW02, p.212]. A good example for interpretability are the cross-language links between Wikipedia articles.

The DBpedia knowledge base is a representation of Wikipedia. With providing Wikipedia content as Linked Data, one could argue, it reduces the representational consistency,

[26] Data set overview of the DBpedia 3.5.1 release:
`http://wiki.dbpedia.org/Downloads351/` (retrieved 03/11/2010).

[27] A general overview of the concept of Linked Data is given in [BHBL09].

because it builds a new representation type. But that would be a tough argumentation. Actually, the DBpedia knowledge base does not influence Wikipedia's consistency as long as Wikipedia and DBpedia hold the same information. In a fictitious case where all musicians are stored only in DBpedia and not in Wikipedia anymore, the consistency would be reduced.

The Interpretability of the knowledge base is given by the Linked Data principles and the RDF Syntax. All items are identified by URIs and are supposed to be self-descriptive. All literals are typed or have language tags respectively. Whether ontology properties and classes are described adequate should be part of a later study. Furthermore, the extraction framework converts all units into the International System of Units[28] to make them easily comparable. Therefore, the representational dimensions are not directly analyzed in this study, but of course, converting errors are caught in the precision evaluation.

### 3.2.4. Accessibility Dimensions

The accessibility dimensions highlight the importance of accessibility of data for its quality. Bizer [Biz07] mentions the following two dimensions:

**Accessibility** "is the extent to which information is available, or easily and quickly retrievable" [PLW02, p.212]. For example, Wikipedia's information are simply accessible over the Web.

**Response Time** is the delay between the request by a user and the response from the System. It depends on different factors, the complexity of the request, the network traffic, server workload, etc. [Biz07].

The DBpedia knowledge base is on-line accessible over a public SPARQL query endpoint[29]. Alternatively the whole data set can be download and queried locally. Another access is the DBpedia Linked Data Interface. All DBpedia URIs are dereferenceable. That gives the option to browse the DBpedia data set with Semantic Web browsers like DISCO[30] or Marbles[31]. The accessibility of information is one of the mayor success factors of the Web [Biz07]. With DBpedia as an open web project, its accessibility should be given, and is therefore not relevant for the evaluation in this study.

[28] `http://www.bipm.org/en/CGPM/db/11/12/` (retrieved 05/11/2010).
[29] `http://dbpedia.org/sparql` (retrieved 05/11/2010).
[30] `http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/` (retrieved 05/11/2010).
[31] `http://wiki.dbpedia.org/Marbles/` (retrieved 05/11/2010).

### 3.2.5. Summary

Information Quality is often defined as fitness for use of information. This definition implies that quality is subjective and task-related. It has many different dimensions which has to be considered when assessing it.

The aim of this work is to evaluate the quality of the DBpedia knowledge base. Because the knowledge base is used from multiple users for different tasks, it should fit to a wide frame of possible use cases. This study concentrates on completeness of the ontology-based extraction result compared to Wikipedia and the precision of the extraction framework, because the DBpedia project has no direct influence on the information quality of Wikipedia and therefore to the intrinsic dimensions.

## 3.3. Quality Assessment Metrics

*Information Quality Assessment* is the process of evaluating information to be fit for use. Therefore, relevant quality dimensions have to be measured and resulting scores should be compared with the quality requirements of the information user. The procedure of measuring an information quality dimensions is named *Assessment Metric* [Biz07]. In Section 3.2 the relevant quality dimensions for this study were introduced. The assessment metrics are described next.

This study uses *Content-Based Metrics*. Content-based means that the content of information is analysed [Biz07], in contrary to *Context-Based Metrics* where meta-data about information build the basis for evaluation. That can be data about the circumstances in which the information are created or the author and date of information [Biz07]. And in contrary to *Rating-Based Metrics*, where the evaluation relays on explicit ratings about information itself, its source or provider [Biz07]. The DBpedia mapping-based extraction result is the information that shall be evaluated. Therefore the information content is compared with the best-case extraction result. There are three evaluation steps in which the triples of the data sets are compared.

First, the *Triple Equality Comparison* checks for each triple in the gold standard, whether the whole triple can be found in the knowledge base. Section 4.4.3 goes into details about the process of this comparison.

Second, the *Predicate Neutral Comparison* checks for each triple in the gold standard that has no match in the first step, whether there is a triple in the knowledge base that has the same subject and object as the gold standard triple. This comparison can find triples with wrong predicates or imprecise mappings in which the gold standard does not concur to it. Such triples are count for completeness, but would decrease the understandability. Section 4.4.4 goes into details about the process of this comparison.

Third, the *Object Similarity Comparison* checks whether there is a triple in the knowledge base that has a similar object to the gold standard triple. Then it calculates whether the deviation from the gold triple could be accepted as accurate. With this method it is possible to find wrong extracted values and rounding differences. The wrong extracted triples are the quality indicator for the precision of the extraction framework. Section 4.4.5 goes into details about the process of this comparison.

The QAF calculates the completeness and the precision of the DBpedia knowledge base. Frequently in analysing studies, the precision metric is used together with a metric named *recall.* The completeness could be interpreted as recall, but because of the completeness aspect of Information Quality, this study takes the completeness term. The used completeness and precision metrics in this study are related to the assessment metrics assembled by Pepino et al. [PLW02].

The number of triples, which have a mapping and are accepted as *present* by one of the steps mentioned above, divided by all mapped triples, build one indicator for completeness. The formula of this triple-based indicator $C_T$ is:

$$\frac{t_p}{t_p + t_m} = C_T$$

In which $t_p$ is the number of triples that are marked as present by one of the comparison steps. Furthermore $t_m$ is the number of triples that are marked as missing in all comparison steps but have a mapping defined. Another completeness indicator is derived from the *Snippets.* Those are parts of an infobox that must be extracted in union to get the correct information. For instance two properties that should be merged to one. Details about Snippets are explained in Section 3.4. This Snippet-based indicator is needed because a different amount of triples can arise from one Snippet. For instance, it is possible that one gold triple arises from one Snippet[32] or many gold triples arise from one Snippet[33]. This could "distort" a triple-based completeness indicator, especially for complex Pattern Categories in which the amount of arising triples from one Snippet differ in a high extent. Therefore, some categories are better measured with a Snippet-based indicator. Here, all Snippets that are completely extracted are counted. The formula of the Snippet-based indicator $C_s$ is:

$$\frac{s_p}{s_p + s_m} = C_S$$

In which $s_p$ is the number of Snippets whose triples are all marked as present and $s_m$ is

[32] For instance Plain Properties, see 3.4.1.

[33] For example One-Property Tables, see 3.4.3.

the number of Snippets whose triples are not marked as present completely. The following example will illustrate the effect of the two indicators: assuming a Pattern Category that only holds two Snippets, here named S1 and S2. From S1 arise 10 triples and from S2 arise 90 triples. For both Snippets a mapping is defined, but only S1 is extracted. The triple-based indicator suggests a completeness of $\frac{10}{10+90} = 0.1 \Leftrightarrow 10\%$. The Snippet-based indicator denotes $\frac{1}{1+1} = 0.5 \Leftrightarrow 50\%$. Of course, this is an extreme example, realistic the amount of triples do not differ in such an extent. The Snippet-based indicator allows additional conclusions. For example, we can deduce the ratio of written mappings which work as they should do. Maybe this is more tangible as a ratio of extracted gold triples.

The precision of the extraction framework is measured by dividing the number of wrong extracted triples $t_w$ by the number of extracted triples. The formula of the precision $P$ is:

$$1 - \frac{t_w}{t_w + t_p} = P$$

For a fine-grained evaluation, the node and data types of the triples are recorded. If, for instance, the ratio of extracted triples with an integer-typed literal as object is extremely low, we could follow that the specific integer parser must contain errors. Due to the recorded Pattern Categories and data types, a huge number of quality scores are produced, that allows cross-over comparison for checking the results for consistency.

## 3.4. Gold Standard and Pattern Categories

The *Gold Standard* of an Wikipedia article is its best-case extraction result. It is necessary to create the data set manually, because it would be pointless to compare two automatically created data sets, when one just wants to find out about errors occurring during the automatic process that produces the data.

The gold triples of an article are also influenced by a process of weighing up between usability and complexity of the triples. For example, intervals are build with two ontology properties, start and end value, instead of one ontology property with an intermediate node, which would contain the two values. Here, an additional ontology property is accepted to avoid the intermediate node.

Generally, all the information from an infobox that describes the entity should be extracted in the best-case. A property that only sets the color of an infobox is not characteristic for the entity. But if the color schema of the infobox assign the entity to a class, the color property is relevant.

Entity-nodes are preferred over literal-nodes. Therefore, if it is possible to recognise an Wikipedia article that represents a string value of an infobox property, the article's entity URI should be preferred. This means the gold triple's object should be an entity instead of a literal. But there should be a technical possibility that the framework could recognise the correct entity and the URI representing it. Some clues about entity recognition are given in section 5.2.3.

There are two sources from which the extraction framework is able to get information. First, the Wikipedia article itself, more precise, its HTML. Second, the Wikipedia article's Wikipedia markup, from which the HTML is rendered. The extraction and the mappings are based on the Wikipedia markup. But during the rendering process, some values from the infobox property markup are transformed in such a way that they are more usable for the user. For example, abbreviations that are converted into Wikipedia links or design templates used in the markup, which produce small images after rendering. Considering this aspect, the gold triples could not be created from the markup only. Therefore, the gold standard building fall back to the HTML too. Thus, the gold triple can hold the linked entity instead of a literal with its abbreviation.

Since the usage of infoboxes is really open, the editors are free in the composition of property values. Therefore, it is not possible to cover all occurrences of values. If an editor has bend the infobox standard (how to use one of its properties) too much, it cannot be assumed that the framework is able to extract the properties original meaning. In such a case, no gold triple would be created.

**Pattern Categories** The design of triples and the kind of mapping that is needed for its extraction depends on the pattern of its underlying infobox properties. Each pattern has a different problematic nature by extracting it. For the different property-patterns, the *Pattern Categories* has been invented. One can say that Pattern Categories are fragments of the whole extraction result on which one want to look in particular.

For this study, the infoboxes of the sample articles are cut into single *Snippets*. A Snippet is the smallest part of infobox-related information that is needed to extract its gold triples. This could be just one property or a group of key-value pairs that have to be extracted in union to get the correct information[34]. Another explanation would be that all triples, which are extracted from a Snippet, fall back to the same mapping. By sorting Snippets into Pattern Categories, it is possible to assign errors to a single extractor or parser, which has to handle the explicit pattern. Uncovered patterns can be recognised and new parsers or mappings invented. The criteria for the Pattern Categories are open. For instance, a Snippet could belong to a Pattern Category, depending on its composition or its content. This seems a little bit inconsistent, but it gives the possibility to design the criteria for a Pattern Category task related[35].

[34] For an example, see 3.4.4.

[35] In this study the Coordinates category is content-related. The Number-Unit category is both, content-

For the sample of this study, 11 Pattern Categories are recognized. They are generated to cover all kinds of compositions used in Snippets. To check further development, the categories will help showing the progress in the specific fields. In the following sections they are introduced and examples for Snippets and their gold standard is given.

### 3.4.1. Plain Property

This category holds the simple cases where only one key-value pair is given that just produces one triple. For instance, the key-value pair

```
| name = Elvis Presley
```

has the gold standard triple

```
<resource/Elvis_Presley> <foaf/name> "Elvis Presley"@en.
```

This is the uncomplicated case, and it is to expect that completeness for this category would be high.

### 3.4.2. Lists

Lists contain more than one item that should be extracted in the best-case and they only need one key-value pair for their composition. All items in a List have the same domain, meaning that each item belongs to an equal ontology property. The single items are separated by commas or <br>-tags for example. The last item can be added after an *and* or *&* sign. A List example taken from the *Infobox musical artist* used at the Elvis page looks like this:

```
| associated_acts = [[The Blue Moon Boys]], [[The Jordanaires]], [[The Imperials]]
```

The gold standard for this List contains the following three triples:

```
<resource/Elvis_Presley> <ontology/associatedBand> <resource/The_Blue_Moon_Boys>
<resource/Elvis_Presley> <ontology/associatedBand> <resource/The_Jordanaires>
<resource/Elvis_Presley> <ontology/associatedBand> <resource/The_Imperials>
```

### 3.4.3. One-Property Tables

are tables that only need one infobox property for their composition. Tables contain more than one item that should be extracted in the best-case. As Lists, the items are separated by comma or <br>-tags. If there is more than one row, a row separator is needed, which mostly is the <br>-tag. In contrary to Lists, items can have different domains.

and design-related.

If an infobox property commonly has the same one-row table, its mapping can assign the single items to different ontology properties. To do so, it is necessary that the domains always remain in the same order. If the table has multiple rows, an intermediate node is necessary for each row. Therefore, this Pattern Category could lead to many intermediate node triples. The next two examples will clarify this matter. A key-value pair that does not result in an intermediate node triple is the several times used born or died property[36]:

```
| born = {{birth date|1935|1|8}}<br /><small>[[Tupelo, Mississippi]]<br />United States</small>
```

This is a table with one row and three columns, which headers could be named as Date, Town and Country. The order of items in the born property is mostly the same. Therefore, it is possible to assign the items to corresponding ontology properties. The first item is the birth date, the second and third are the birth town and country. The gold triples are:

```
<resource/Elvis_Presley> <ontology/birthDate> "1935-01-08"^^<XMLSchema#date>
<resource/Elvis_Presley> <ontology/birthPlace> <resource/Tupelo%2C_Mississippi>
<resource/Elvis_Presley> <ontology/birthPlace> <resource/United_States>
```

The second example produces intermediate nodes from an One-Property Table. This property can be found at the Cowboy Bebop page[37].

```
| licensor = {{flagicon|Japan}} [[Bandai Visual]]<br />{{flagicon|United States}} {{flagicon|Canada}}
[[Bandai Entertainment]]
```

This is a table with two rows and the columns Country and Licensor. A particular difficulty in this example are the two country items in the second row. The best-case statements are:

```
<resource/Cowboy_Bebop> <ontology/licensor> <resource/Cowboy_Bebop__licensor_1>
<resource/Cowboy_Bebop__licensor_1> <ontology/country> <resource/Japan>
<resource/Cowboy_Bebop__licensor_1> <ontology/company> <resource/Bandai_Visual>
<resource/Cowboy_Bebop> <ontology/licensor> <resource/Cowboy_Bebop__licensor_2>
<resource/Cowboy_Bebop__licensor_2> <ontology/country> <resource/United States>
<resource/Cowboy_Bebop__licensor_2> <ontology/country> <resource/Canada>
<resource/Cowboy_Bebop__licensor_2> <ontology/company> <resource/Bandai_Entertainment>
```

The sentence “Elvis Presley was born in the United States”, is representable as one triple, because there are only one subject, one predicate and one object. The sentence “In Japan, the licensor of the anime Cowboy Bebop is the company Bandai Visual” is not representable in one triple, because of the licensor’s specific restriction to a country. Therefore intermediate nodes are required.

[36] Example taken from http://en.wikipedia.org/wiki/Elvis_Presley/ (retrieved 26/10/2010).

[37] http://en.wikipedia.org/wiki/Cowboy_Bebop/ (retrieved 26/10/2010).

### 3.4.4. Multi-Property Tables

are tables that need more than one infobox property for their composition. For each column in the table a single infobox property is used. If there are more than one row in these tables, the property values are ordered lists, or for each row a new consecutively numbered set of properties is used. In the last case, for each value a single property is given. For these Pattern Category special mappings are required that join the key-value pairs.

The following example is taken from the Wikipedia page of John Charles[38]:

```
| years = 1948&ndash;1957<br>1957&ndash;1962
| clubs = [[Leeds United A.F.C.|Leeds United]]<br>[[Juventus F.C.|Juventus]]
| caps(goals) = 297 (150)<br>150 {{0}}(93)
```

Here, the Snippet contains three key-value pairs, because they are all needed to extract the whole table. Every column has a property which contains a List, or in case of the *caps(goals)*-property a One-Property Table. The gold triples are:

```
<resource/John_Charles> <ontology/seniorCareer> <resource/John_Charles__seniorCareer_1>
<resource/John_Charles__seniorCareer_1> <ontology/team> <resource/Leeds_United_A.F.C.>
<resource/John_Charles__seniorCareer_1> <ontology/activeYearsStartYear> "1948"^^<XMLSchema#gYear>
<resource/John_Charles__seniorCareer_1> <ontology/activeYearsEndYear> "1957"^^<XMLSchema#gYear>
<resource/John_Charles__seniorCareer_1> <ontology/appearances> "297"^^<XMLSchema#integer>
<resource/John_Charles__seniorCareer_1> <ontology/goals> "150"^^<XMLSchema#integer>
<resource/John_Charles> <ontology/seniorCareer> <resource/John_Charles__seniorCareer_2>
<resource/John_Charles__seniorCareer_2> <ontology/team> <resource/Juventus_F.C.>
<resource/John_Charles__seniorCareer_2> <ontology/activeYearsStartYear> "1957"^^<XMLSchema#gYear>
<resource/John_Charles__seniorCareer_2> <ontology/activeYearsEndYear> "1962"^^<XMLSchema#gYear>
<resource/John_Charles__seniorCareer_2> <ontology/appearances> "150"^^<XMLSchema#integer>
<resource/John_Charles__seniorCareer_2> <ontology/goals> "93"^^<XMLSchema#integer>
```

The next example is taken from Bobby Charlton's article[39]. This table contains two Snippets which can extracted separately. Each row is a single Snippet that belongs to the Pattern Category Multi-Property Table. Both rows are listed here so that one can recognise the table. Each row of the table has its own *years*, *clubs* and *caps* property.

```
| years1 = 1954–1973 | clubs1 = [[Manchester United F.C.|Manchester United]] | caps1 = 606 | goals1 = 199
| years2 = 1973–1975 | clubs2 = [[Preston North End F.C.|Preston North End]] | caps2 = 38 | goals2 = 8
```

The resulting gold triples have the same structure like above, but each set of intermediate nodes could extracted separately. The first set of gold triples is:

```
<resource/Bobby_Charlton> <ontology/seniorCareer> <resource/Bobby_Charlton__seniorCareer_1>
<resource/Bobby_Charlton__seniorCareer_1> <ontology/team> <resource/Manchester_United_F.C.>
<resource/Bobby_Charlton__seniorCareer_1> <ontology/activeYearsStartYear> "1954"^^<XMLSchema#gYear>
```

[38] http://en.wikipedia.org/wiki/John_Charles/ (retrieved 05/11/2010).

[39] http://en.wikipedia.org/wiki/Bobby_Charlton/ (retrieved 19/11/2010).

```
<resource/Bobby_Charlton__seniorCareer_1> <ontology/activeYearsEndYear> "1973"^^<XMLSchema#gYear>
<resource/Bobby_Charlton__seniorCareer_1> <ontology/appearances> "606"^^<XMLSchema#integer>
<resource/Bobby_Charlton__seniorCareer_1> <ontology/goals> "199"^^<XMLSchema#integer>
```

The second set of gold triples is:

```
<resource/Bobby_Charlton> <ontology/seniorCareer> <resource/Bobby_Charlton__seniorCareer_2>
<resource/Bobby_Charlton__seniorCareer_2> <ontology/team> <resource/Preston_North_End_F.C.>
<resource/Bobby_Charlton__seniorCareer_2> <ontology/activeYearsStartYear> "1973"^^<XMLSchema#gYear>
<resource/Bobby_Charlton__seniorCareer_2> <ontology/activeYearsEndYear> "1975"^^<XMLSchema#gYear>
<resource/Bobby_Charlton__seniorCareer_2> <ontology/appearances> "38"^^<XMLSchema#integer>
<resource/Bobby_Charlton__seniorCareer_2> <ontology/goals> "8"^^<XMLSchema#integer>
```

### 3.4.5. Coordinates

Geographical coordinates are information about a location of an entity. They are given in latitude and longitude. The Coordinates-Pattern is content-based. There are different compositions of coordinate properties, but from all should result three coordinate triples. Due to the same gold triple format and the fact that there is a Geocoordinate Parser in the extraction framework that handles all kinds of coordinate related properties, the different coordinate compositions are sorted in the same Pattern Category. The three patterns given below are coordinates of the South African city Cape Town. In the first and second row the coordinates are given in a sexagesimal[40] notation. Each degree is divided into 60 minutes and each minute into 60 seconds. Together with the information about the hemisphere, latitude and longitude also can expressed as decimal values, like shown in the third row.

```
| latd=33 | latm=55 | lats=31 | latNS=S | longd=18 | longm=25 | longs=26 | longEW=E
| coordinates = {{coord|33|55|31|S|18|25|26|E}}
| latitude=-33.925278 | longitude=18.42389
```

All three key-value pairs result in the following gold triples.

```
<resource/Cape_Town> <georss/point> "-33.925278 18.42389"@en
<resource/Cape_Town> <geo/wgs84_pos#long> "18.42389"^^<XMLSchema#float>
<resource/Cape_Town> <geo/wgs84_pos#lat> "-33.925278"^^<XMLSchema#float>
```

### 3.4.6. Number-Units

All property values that contain a number and a unit, belong to the Number-Units-Pattern. Therefore this category is also content-based. This is motivated by the extraction framework's unit parser, which handles the amount of different unit types and their conversion process. Examples for key-value pairs that belong to this category are:

[40] Sexagesimal means a numeral system with sixty as its base.

```
| frequency = 730 [[kilohertz|kHz]]
| period = 2 years
| height = 324 m
| length = {{convert|4079|mm|in|1|abbr=on}}
| budget = [[United States dollar|$]]49,000,000 (estimated)
```

The arising gold triples are:

```
<resource/Entity> <ontology/frequency> "730000.0"^^<XMLSchema#double>
<resource/Entity> <ontology/period> "6.311385E07"^^<XMLSchema#double>
<resource/Entity> <ontology/height> "324.0"^^<XMLSchema#double>
<resource/Entity> <ontology/length> "4.079"^^<XMLSchema#double>
<resource/Entity> <ontology/budget> "4.9E7"^^<datatype/usDollar>
```

The framework converts the numbers to the standard units of the International System of Units[41]. Currencies cannot be converted into other currencies, because of the fluctuating exchange rates.

### 3.4.7. Intervals

A property with a value that contains two items building an interval together, belong to this category. The items are separated in different ways, but mostly by a minus-sign or the word *to.* If an infobox property belongs to this category, always two triples are extracted from it. These are the triple with the start value and the triple with the end value. Intervals are difficult to extract because of the different domains that are expressed in Intervals. If a time Interval runs to the present, only the start point is set. Examples are:

```
| length = 34 m - 115 m
| activeYears = 1954–1973
| recorded = April to November 1982
| activeYears = 2007 - present
```

The resulting gold triples are:

```
<resource/Entity> <ontology/lengthIntervalStart> "34.0"^^<XMLSchema#double>
<resource/Entity> <ontology/lengthIntervalEnd> "115.0"^^<XMLSchema#double>
<resource/Entity> <ontology/activeYearsStartYear> "1954"^^<XMLSchema#gYear>
<resource/Entity> <ontology/activeYearsEndYear> "1973"^^<XMLSchema#gYear>
<resource/Entity> <ontology/recordedStartDate> "1982-04"^^<XMLSchema#gYearMonth>
<resource/Entity> <ontology/recordedEndDate> "1982-11"^^<XMLSchema#gYearMonth>
<resource/Entity> <ontology/activeYearsStartYear> "2007"^^<XMLSchema#gYear>
```

[41] "International System of Units (SI), French Système Internationale d'Unités, international decimal system of weights and measures derived from and extending the metric system of units. Adopted by the 11th General Conference on Weights and Measures in 1960, it is abbreviated SI in all languages." - Encyclopædia Britannica
http://www.britannica.com/EBchecked/topic/291305/International-System-of-Units-SI# (retrieved 15/02/2011).

### 3.4.8. Open Properties

Open Properties are made up of two key-value pairs, which together build another key-value pair. One contains the key, and the other contains the value. Because of the variable key of the built key-value pair, it is hard to order these properties to the DBpedia ontology. The problem is solved by the ontology properties *openProperty*, *key* and *value*. As an example we look in the infobox at Cape Town's Wikipedia article[42]:

```
| leader_title = [[Mayor of Cape Town|Mayor]]
| leader_name = [[Dan Plato]]
```

A set of best-case intermediate node triples is built:

```
<resource/Cape_Town> <ontology/openProperty> <resource/Cape_Town__openProperty_1>
<resource/Cape_Town__openProperty_1> <ontology/key> <resource/Mayor_of_Cape_Town>
<resource/Cape_Town__openProperty_1> <ontology/value> <resource/Dan_Plato>
```

In the case that only strings are available, the triples would be:

```
<resource/Cape_Town> <ontology/openProperty> <resource/Cape_Town__openProperty_1>
<resource/Cape_Town__openProperty_1> <ontology/key> "Mayor"@en
<resource/Cape_Town__openProperty_1> <ontology/value> "Dan Plato"@en
```

Another possibility for the gold triples is given if the key is restricted to any issue. Then it becomes possible to map the infobox property to an ontology property and no intermediate node is required. The information of the property's key-name is already satisfying. From leader_title and leader_name it is already known that these properties describe the leader of the settlement, which is described in the Wikipedia article. Therefore, the gold triples would be the following:

```
<resource/Cape_Town> <ontology/leaderTitle> <resource/Mayor_of_Cape_Town>
<resource/Cape_Town> <ontology/leader> <resource/Dan_Plato>
```

Here, an intermediate node is avoided, but the connection between the Mayor-, and the Dan_Plato node is only indirectly given via the Cape_Town node and by the name of the ontology properties. It could be argued that with the extraction of Dan Plato's Wikipedia article, the connection could be made. Indeed, Dan Plato's article contains the Infobox Politician, which has the property | office=32nd [[Mayor of Cape Town]]. In this case, it would be enough to extract the leader_name property of the settlement Cape Town, and the office property of Dan Plato. The following two triples would be extracted:

```
<resource/Dan_Plato> <ontology/office> <resource/Mayor_of_Cape_Town>
<resource/Cape_Town> <ontology/leader> <resource/Dan_Plato>
```

[42] http://en.wikipedia.org/wiki/Cape_town/ (retrieved 24/10/2010).

Now it is possible to build a graph: Cape Town has the leader Dan Plato who is a Mayor. If the values of the two initial properties are Wikilinks, the connection can be made directly. In a case of stings, this is not possible, because literal nodes are always the end of a graph. Therefore we had to weigh up between these two possibilities of creating the gold triples. Depending on this decision, we expect a different mapping for the two properties.

### 3.4.9. Open Property Tables

The difference to the Open Property category is an additional hierarchy level. That is represented due to an intermediate node working as a kind of table-frame. The following example can be found at Cape Town's Wikipedia article too:

```
| demographics_type1 = Racial makeup
| demographics1_title1 = [[Coloured]]
| demographics1_info1 = 44.0%
| demographics1_title2 = [[Black African]]
| demographics1_info2 = 34.9%
```

The ontology property *openPropertyTable* contains the table. This table has a header property to identify the subject and row properties, which hold the table its columns as key and value properties. The best-case triples would be:

```
<res/Cape_Town> <ont/openPropertyTable> <res/Cape_Town__openPropertyTable_1>
<res/Cape_Town__openPropertyTable_1> <ont/header> "Racial makeup"@en
<res/Cape_Town__openPropertyTable_1> <ont/row> <res/Cape_Town__openPropertyTable_1__row_1>
<res/Cape_Town__openPropertyTable_1__row_1> <ont/key> <res/Coloured>
<res/Cape_Town__openPropertyTable_1__row_1> <ont/value> "0.44"^^<XMLSchema#double>
<res/Cape_Town__openPropertyTable_1> <ont/row> <res/Cape_Town__openPropertyTable_1__row_2>
<res/Cape_Town__openPropertyTable_1__row_2> <ont/key> <res/Black_African>
<res/Cape_Town__openPropertyTable_1__row_2> <ont/value> "0.349"^^<XMLSchema#double>
```

### 3.4.10. Internal Templates

Some infobox property values go back to infobox internal templates. For example, this internal templates can convert abbreviations to Wikipedia links instead. If the property value from the MediaWiki markup is transformed during the rendering to something more precisely, the infobox property belongs to the this Pattern Category. One example is found on the Wikipedia page of Frankfurt am Main[43]:

```
| Regierungsbezirk = Darmstadt
```

[43] http://en.wikipedia.org/wiki/Frankfurt_am_Main/ (retrieved 14/10/2010).

The value of the property is just a string. But during the rendering, this string is converted to the Wikipedia link [[Darmstadt_(region)|Darmstadt]]. The new link is much more precise than the value of the infobox property. The following gold triple results:

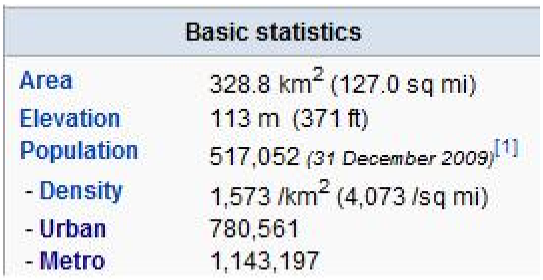
```
<resource/Frankfurt_am_Main> <ontology/administrativeDistrict> <resource/Darmstadt_%27region%28>
```

A more complex case are infoboxes that integrate data from other infoboxes. One example can be found at the Wikipedia article of the German city of Dresden[44]. The infobox property *Einwohner* gives the population number of the city.

```
| Einwohner = 512234
```

The rendered Wikipedia page shows a different population number:

Figure 5: Infobox part of Dresden's Wikipedia article



| Basic statistics | |
|---|---|
| Area | 328.8 km² (127.0 sq mi) |
| Elevation | 113 m (371 ft) |
| Population | 517,052 (31 December 2009)[1] |
| - Density | 1,573 /km² (4,073 /sq mi) |
| - Urban | 780,561 |
| - Metro | 1,143,197 |

This irritating aspect belongs to the design of the *Infobox German location*. It gets its population data from the *Data retrieval template*[45] *Population Germany*[46], without paying regard to the own population property anymore.

A different case of the use of data retrieval templates is found in the *Infobox settlement*[47]. For Swedish municipalities there are also data templates that contain the population data. But here they are observable used as templates in the population property.

```
| population_total = {{Population Swedish municipality|municipality=Lund Municipality}}
| population_as_of = {{Population Swedish municipality|TXT=date}}
```

Of course, the gold triples should contain the correct values, which are:

```
<resource/Lund_Municipality> <ontology/populationTotal> "108947"^^<XMLSchema#double>
<resource/Lund_Municipality> <ontology/populationTotalAsOf> "2010-06-30"^^<XMLSchema#date>
```

[44] http://en.wikipedia.org/wiki/Dresden/ (retrieved 18/10/2010).

[45] A data retrieval template extracts data from data templates. Data templates contain independently retrievable data items. See:
- http://en.wikipedia.org/wiki/Category:Data_retrieval_templates/ (retrieved 18/10/2010)
- http://en.wikipedia.org/wiki/Category:Data_templates/(retrieved 18/10/2010)

[46] http://en.wikipedia.org/wiki/Template:Population_Germany/ (retrieved 18/10/2010).

[47] http://en.wikipedia.org/wiki/Template:Infobox_settlement/ (retrieved 18/10/2010).

### 3.4.11. Merged Properties

If a Snippet contains two infobox properties that are supposed to be merged to one, they belong to this category. The most common case is the height of persons that are split in feet and inches. Here taken from the page of the cricketer Danish Kaneria[48].

```
| heightft = 6
| heightinch = 1
```

These two properties result in one gold triple with a value converted to meter:

```
<resource/Danish_Kaneria> <ontology/height> "1.854"^^<XMLSchema#double>
```

# 4. Experiment

The Quality Assessment Framework (QAF) compares the DBpedia 3.5.1[49] knowledge base[50] with a manually created gold standard for a sample of Wikipedia articles. How the sample is build, and what it includes is explained in section 4.1. The concrete gold standard used in the sample was already introduced in section 3.4. The QAF implementation and the evaluation process are introduced in Sections 4.2 and 4.4.

## 4.1. Sample Structure

Since it is impractical to analyse the whole data set of Wikipedia, this study is based on a sample of 75 Wikipedia articles[51]. In the first step the most used infoboxes in Wikipedia were taken. In the second step, for each of these infoboxes the first 500 Wikipedia articles that appear when using the "*what links here*"-link were inspected. This link is given on every Wikipedia page, also on the pages of infoboxes[52]. It shows a list of Wikipedia pages that link to the current page. From this 500 Wikipedia articles, the one whose infobox has the most filled properties was chosen for the sample. This method assured a broad coverage of 75 popular topics. Table 5 in the Appendix contains the list of the most used infoboxes and the articles in the sample.

To write the sample data set, a user interface was implemented that allows a user to iterate over the sample articles and their Snippets. For each Snippet, the following information was added to the sample data set:

[48] `http://en.wikipedia.org/wiki/Danish_Kaneria/` (retrieved 14/10/2010).

[49] `http://wiki.dbpedia.org/Downloads351#ontologyinfoboxproperties` (retrieved 20/11/2010).

[50] Since only the strict ontology-based extraction is used, it's only one part of the knowledge base.

[51] Based one the Wikipedia dumps generated on 24th March 2010.

[52] In figure 1, this link is marked as B. For instance the list of pages that link to the infobox musical artist:`http://en.wikipedia.org/w/index.php?title=Special:WhatLinksHere/Template:Infobox_musical_artist&limit=500` (retrieved 05/11/2010).

- The name of the infobox.
- The URI of the Wikipedia page.
- The key-value sets of the infobox property that builds a pattern.
- The *Pattern Category* for the key-value set.
- The gold standard, which are all RDF triples that could be extracted from the key-value set in the *best-case*.
- A comment field, with additional information.

The data is stored in XML files. There is one file which contains a list of the sample articles and for each article a file which contains its Snippets.

The 75 Wikipedia articles are cut into 1466 Snippets, for which 3215 gold triples were created. For 54 % of the Snippets, complete mappings are defined. Table 6 shows the distribution of Snippets to Pattern Categories.

To get more detailed results and the possibility of more precisely conclusions, the gold triples are categorised too. They are first split up by the Pattern Category and second by their type of subject and object. Generally, the subject of a triple can be an entity or an intermediate node. Thus, there are these two categories for the triples subject. The objects are divided into literals and resources first. Second, the literals are divided into their XML Schema data types[53], and the resources in turn into entities, intermediate nodes and URLs. Table 7 shows the distribution of the gold triples in the sample by their node types and their data types.

## 4.2. Implementation

The Quality Assessment Framework (QAF) is implemented in Java. The *Jena Semantic Web Framework*[54] (Jena) is used to handle the RDF triples. Jena is an open source Java framework for building Semantic Web applications. The following section describes the main classes and methods.

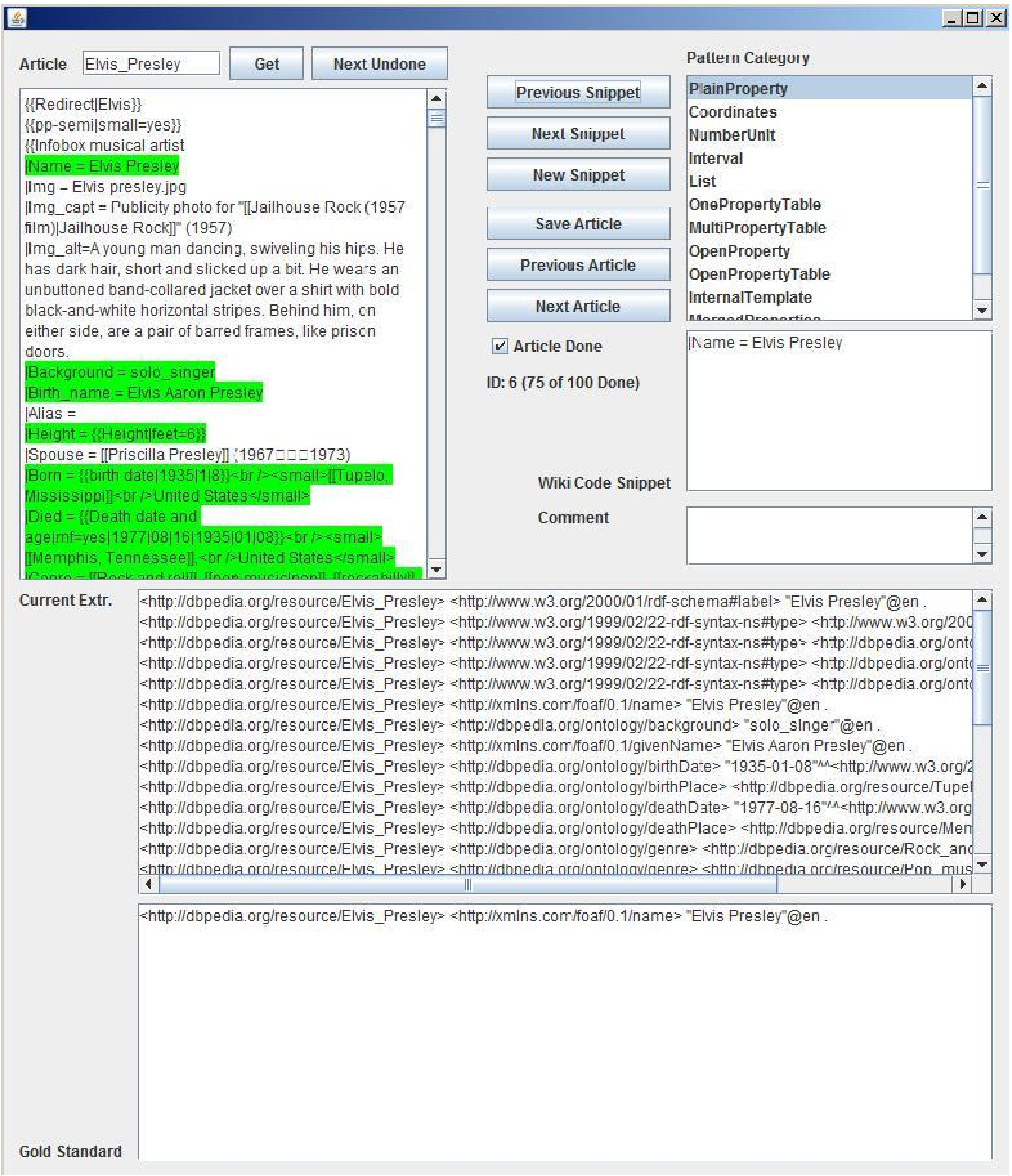
The framework has a user interface to write and edit the gold standard. It allows a user to iterate over the sample articles and shows the article's Wikipedia markup. For help by creating the gold standard, the current extraction result is displayed. For each article one can iterate over its Snippets and categorises them. Figure 6 shows the interface. On the top left, one can see the field with the Wikipedia article name and the articles

[53] XML Schema Part 2: Datatypes Second Edition, see `http://www.w3.org/TR/xmlschema-2/`.

[54] For more information about Jena, see `http://jena.sourceforge.net/` (retrieved 17/02/2011).

Wikipedia code below it. Lines of Wikipedia code that are already used in a Snippet are marked green. On the top right, the Pattern Category can be selected. Below the list of Pattern Categories is the field for the snippet of Wikipedia code, from which the gold triples are build. The text field on the bottom contains the gold triples of the chosen Snippet, and above are the articles current extraction result. The gold standard is saved in the *“...\DBpediaQAF\articles\data\”*-folder.

Figure 6: User Interface for Editing the Gold Standard



Article: Elvis_Presley

```
{{Redirect|Elvis}}
{{pp-semi|small=yes}}
{{Infobox musical artist
|Name = Elvis Presley
|Img = Elvis presley.jpg
|Img_capt = Publicity photo for "[[Jailhouse Rock (1957
film)|Jailhouse Rock]]" (1957)
|Img_alt=A young man dancing, swiveling his hips. He
has dark hair, short and slicked up a bit. He wears an
unbuttoned band-collared jacket over a shirt with bold
black-and-white horizontal stripes. Behind him, on
either side, are a pair of barred frames, like prison
doors.
|Background = solo_singer
|Birth_name = Elvis Aaron Presley
|Alias =
|Height = {{Height|feet=6}}
|Spouse = [[Priscilla Presley]] (1967□□□1973)
|Born = {{birth date|1935|1|8}}<br /><small>[[Tupelo,
Mississippi]]<br />United States</small>
|Died = {{Death date and
age|mf=yes|1977|08|16|1935|01|08}}<br /><small>
[[Memphis, Tennessee]],<br />United States</small>
|Genre = [[Rock and roll]], [[pop music|pop]], [[rockabilly]]
```

☑ Article Done

ID: 6 (75 of 100 Done)

Wiki Code Snippet: |Name = Elvis Presley

Current Extr.

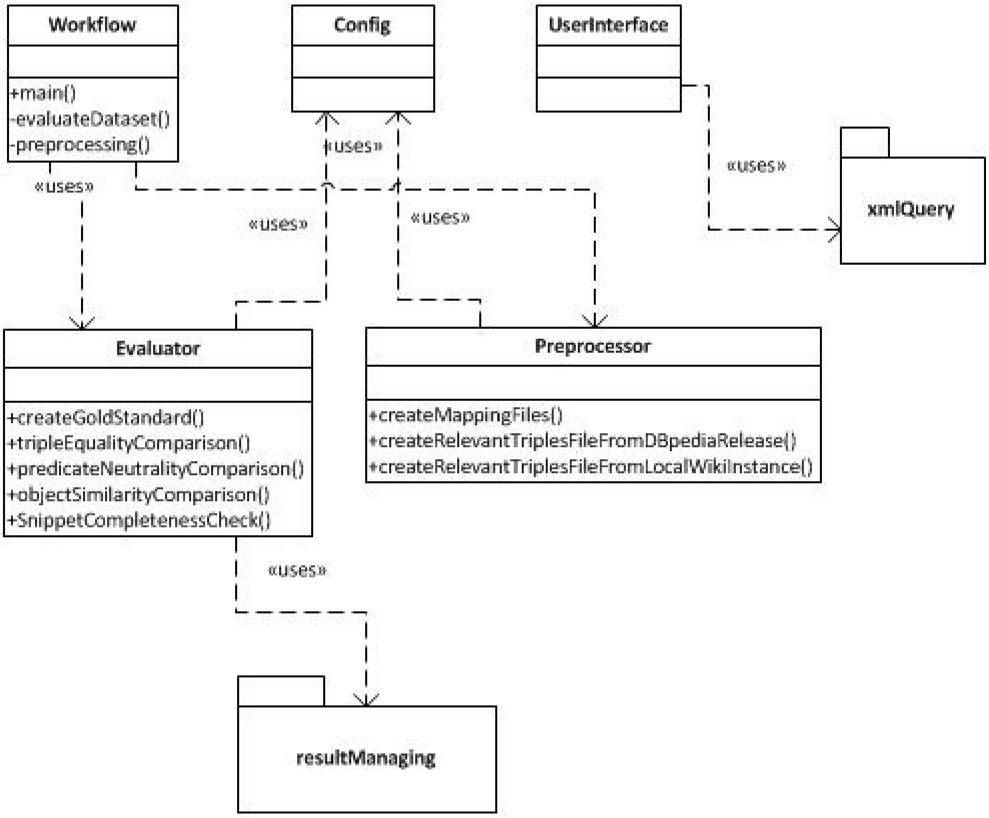
```
<http://dbpedia.org/resource/Elvis_Presley> <http://www.w3.org/2000/01/rdf-schema#label> "Elvis Presley"@en .
<http://dbpedia.org/resource/Elvis_Presley> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/200
<http://dbpedia.org/resource/Elvis_Presley> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ont
<http://dbpedia.org/resource/Elvis_Presley> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ont
<http://dbpedia.org/resource/Elvis_Presley> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ont
<http://dbpedia.org/resource/Elvis_Presley> <http://xmlns.com/foaf/0.1/name> "Elvis Presley"@en .
<http://dbpedia.org/resource/Elvis_Presley> <http://dbpedia.org/ontology/background> "solo_singer"@en .
<http://dbpedia.org/resource/Elvis_Presley> <http://xmlns.com/foaf/0.1/givenName> "Elvis Aaron Presley"@en .
<http://dbpedia.org/resource/Elvis_Presley> <http://dbpedia.org/ontology/birthDate> "1935-01-08"^^<http://www.w3.org/2
<http://dbpedia.org/resource/Elvis_Presley> <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Tupel
<http://dbpedia.org/resource/Elvis_Presley> <http://dbpedia.org/ontology/deathDate> "1977-08-16"^^<http://www.w3.org
<http://dbpedia.org/resource/Elvis_Presley> <http://dbpedia.org/ontology/deathPlace> <http://dbpedia.org/resource/Mem
<http://dbpedia.org/resource/Elvis_Presley> <http://dbpedia.org/ontology/genre> <http://dbpedia.org/resource/Rock_and
<http://dbpedia.org/resource/Elvis_Presley> <http://dbpedia.org/ontology/genre> <http://dbpedia.org/resource/Pop_mus
```

Gold Standard

```
<http://dbpedia.org/resource/Elvis_Presley> <http://xmlns.com/foaf/0.1/name> "Elvis Presley"@en .
```

The main class of the framework is named *Workflow*. From here, the preprocessing and the evaluation can be handled. It is located in the *dbPediaQAF*-package, which also contains the *UserInterface* class, the *Preprocessor* class, the *Evaluator* class, and the *Config* class. The class diagram of this package can be found in Figure 7. The *Config* is a static class which holds framework specifications like path information to files. The *UserInterface* only uses the *xmlQuery*-package to get data from the gold standard files.

Figure 7: Classes in the dbPediaQAF package


Workflow
+main()
-evaluateDataset()
-preprocessing()
Config
UserInterface
«uses»
«uses»
«uses»
«uses»
«uses»
xmlQuery
Evaluator
+createGoldStandard()
+tripleEqualityComparison()
+predicateNeutralityComparison()
+objectSimilarityComparison()
+SnippetCompletenessCheck()
Preprocessor
+createMappingFiles()
+createRelevantTriplesFileFromDBpediaRelease()
+createRelevantTriplesFileFromLocalWikiInstance()
«uses»
resultManaging

The *Evaluator* handles triples and Snippets to the *resultManaging*-package. There they are stored in the *ResultSet* class that has sub result sets. Depending on the comparison method in which a triple or Snippet is analysed, the *Evaluator* passes them to the different sub sets, which are *CompletenessHandswitch* objects. As the name of the *CompletenessHandswitch* class infers, triples and Snippets are manually split between "present" and "missing" by the *Evaluator* class. Section 4.4 covers details about the operations in the *Evaluator* and the *Preprocessor* class. From the *CompletenessHandswitch* objects, the triples and Snippets are passed down to *Switch* classes. Here, they are automatically split depending on their Pattern Category and their node and data type. All switches implement the *Switch* interface that demand for a *get()*, an *add()*, and a *remove()* method for triples. Additional to the switches, the *resultManaging*-package contains the *DataSet* class. It holds the gold standard, all mapping information of infoboxes, and some *HashMaps*[55], which map triples to infoboxes, to Snippets and to Pattern Categories. The *DataSet* initialisation is the first step of the evaluation process and described below in Section 4.4.2. The class diagram of the *resultManaging*-package is illustrated in Figure 8.

The *xmlQuery*-package contains four classes that are responsible for the process of loading and updating XML data. The Java classes Marshaller[56] and Unmarshaller[57] are used for that task. The *ArticleItemCol* class build the root item of the *articles.xml* file. It contains the list of Wikipedia articles that are in the sample. The information whether an article is marked as "done" in the user interface and the infobox template used in the article are also saved in this file. *ArticleItem* objects represents the XML child nodes of the *ArticleItemCol* root. The *Article* class build the root items of the single article XML files in the *DBpediaQAF\articles\data* folder. Here, each article with a gold standard defined has its own XML file. This file contains all Snippets of an article. The *Snippet* objects represent this XML child nodes of the *Article* root object. The class diagram of the *xmlQuery*-package is illustrated in Figure 9.

The *util*-package holds a Java class for exporting the evaluation result to an Excel file, and a class to calculate the similarity of strings. It also contains *Enum Types*[58] of the Pattern Categories and the XML Schema data types used for the evaluation.

[55] A HashMap is a Java object that maps keys to values. See Java API, `http://download.oracle.com/javase/1.4.2/docs/api/java/util/HashMap.html` (retrieved 18/02/2011).

[56] "The Marshaller class is responsible for governing the process of serializing Java content trees back into XML data." - Java API, `http://download.oracle.com/javase/6/docs/api/javax/xml/bind/Marshaller.html` (retrieved 18/02/2011).

[57] "The Unmarshaller class governs the process of deserialising XML data into newly created Java content trees, optionally validating the XML data as it is unmarshalled." - Java API, `http://download.oracle.com/javase/6/docs/api/javax/xml/bind/Unmarshaller.html` (retrieved 18/02/2011).

[58] "An enum type is a type whose fields consist of a fixed set of constants." - `http://download.oracle.com/javase/tutorial/java/javaOO/enum.html` (retrieved 18/02/2011).

Figure 8: Classes in the dbPediaQAF.resultManaging package

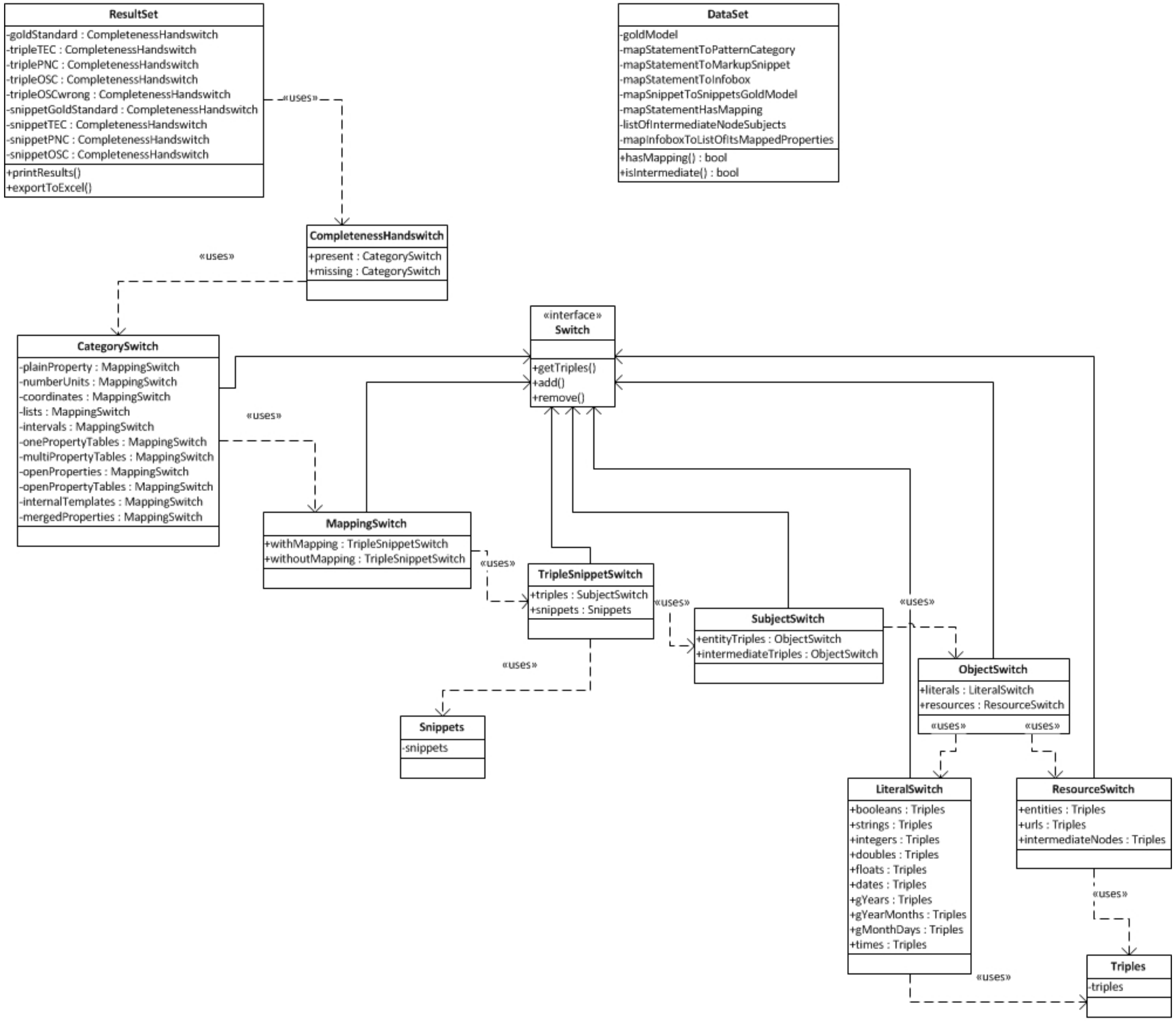
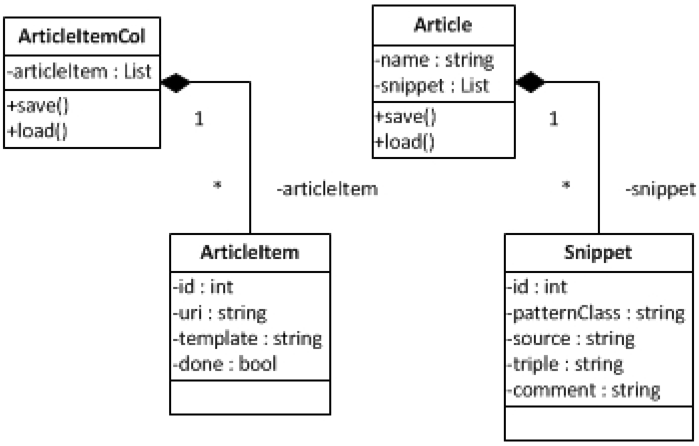

ResultSet
-goldStandard : CompletenessHandswitch
-tripleTEC : CompletenessHandswitch
-triplePNC : CompletenessHandswitch
-tripleOSC : CompletenessHandswitch
-tripleOSCwrong : CompletenessHandswitch
-snippetGoldStandard : CompletenessHandswitch
-snippetTEC : CompletenessHandswitch
-snippetPNC : CompletenessHandswitch
-snippetOSC : CompletenessHandswitch
+printResults()
+exportToExcel()
«uses»
DataSet
-goldModel
-mapStatementToPatternCategory
-mapStatementToMarkupSnippet
-mapStatementToInfobox
-mapSnippetToSnippetsGoldModel
-mapStatementHasMapping
-listOfIntermediateNodeSubjects
-mapInfoboxToListOfItsMappedProperties
+hasMapping() : bool
+isIntermediate() : bool
CompletenessHandswitch
+present : CategorySwitch
+missing : CategorySwitch
«uses»
«interface»
Switch
+getTriples()
+add()
+remove()
CategorySwitch
-plainProperty : MappingSwitch
-numberUnits : MappingSwitch
-coordinates : MappingSwitch
-lists : MappingSwitch
-intervals : MappingSwitch
-onePropertyTables : MappingSwitch
-multiPropertyTables : MappingSwitch
-openProperties : MappingSwitch
-openPropertyTables : MappingSwitch
-internalTemplates : MappingSwitch
-mergedProperties : MappingSwitch
«uses»
MappingSwitch
+withMapping : TripleSnippetSwitch
+withoutMapping : TripleSnippetSwitch
«uses»
TripleSnippetSwitch
+triples : SubjectSwitch
+snippets : Snippets
«uses»
SubjectSwitch
+entityTriples : ObjectSwitch
+intermediateTriples : ObjectSwitch
«uses»
«uses»
ObjectSwitch
+literals : LiteralSwitch
+resources : ResourceSwitch
«uses»
«uses»
Snippets
-snippets
LiteralSwitch
+booleans : Triples
+strings : Triples
+integers : Triples
+doubles : Triples
+floats : Triples
+dates : Triples
+gYears : Triples
+gYearMonths : Triples
+gMonthDays : Triples
+times : Triples
ResourceSwitch
+entities : Triples
+urls : Triples
+intermediateNodes : Triples
«uses»
«uses»
Triples
-triples

Figure 9: Class Diagram of the xmlQuery-package



## 4.3. User Instructions

The gold standard of this study is based on the Wikipedia dump from the 24th of March 2010. It is necessary to hold this dump in a local MediaWiki instance to extract triples from it with different versions of the DBpedia extraction framework. Only this way, meaningful conclusions about changes of the extraction quality are possible. Furthermore, one has to start the extraction server of a local DBpedia version. But before, one has to edit the *SimplePropertyMapping.scala* and the *Extraction.scala* file. The first can be found in the folder DBpedia\trunk\extraction\core\src\main\scala\org\dbpedia\extraction\-mappings\. Here, in the *writeUnitValue*-method, the lines that writes the specific properties must be deleted or commented out to avoid that they appear in the extraction result. The specific properties are additional triples, which would distort the analysis result. The *Extraction.scala* file is located in DBpedia\trunk\extraction\server\src\main\scala\org\-dbpedia\extraction\server\resources\. In the extract method, the URL to the MediaWiki instance must be changed from the live Wikipedia to a local instance. To start the extraction server, one has to execute the *Server.scala* file that is located one folder under the *Extraction.scala* file.

Before the actual evaluation, the file with the triples that should compared with the gold standard and the favoured mapping version files, must be created. That is done by the *Preprocessor*, it creates a new file, which contains all triples that are related to the sample Wikipedia articles. This file is saved at the location set by the *relevantDBpediaTriplesPath*-attribute of the *Config* class. The QAF provides two methods to create this relevant triples file. First, if one has a DBpedia dump file, the *Preprocessor*'s *createRelevantTriplesFileFromDBpediaRelease*-method must be executed. The path to the Dbpedia dump is set by the *dbpediaReleasePath*-attribute of the *Config* class. Second, the relevant triples file can be created from the local MediaWiki instance and the local DBpedia version. These is done by the *Preprocessor*'s *createRelevantTriplesFileFromLocalWikiInstance*-method. The *Preprocessor* creates the mapping files also. The method named *createMappingFiles* collects the mappings of the sample articles that were up-to-date at the date defined in the *Config* class. Because the DBpedia extraction framework always takes the newest mappings for the extraction, the collected mapping versions are only important for the QAF by marking triples and Snippets as mapped during the evaluation. Thus, it actually is not possible to isolate the two effects - improvement of the extraction framework or improvement of the mappings - that lead to better quality.

The user interface can be started by running the *main*-method of the *UserInterface* class. To run the framework on has to execute the *Workflow*'s *main*-method. It contains the two lines:

```
workflow.preprocessing();
workflow.evaluateDataset();
```

If the preprocessing is done by another run before, one should comment out the first line, because there is no need to run the preprocessing a second time. For instance, if one only had changed a Pattern Category or a triple in the gold standard, there is no need for the preprocessing again. The second line starts the evaluation. The *Config* holds some parameters for the output. One can only print the result to the console or export it to an Excel file. This *evaluationResult.xls* file can found in the DBpediaQAF's result folder. Excel do not recalculate its functions automatically after the result file was updated. A easy trick is to replace all "=" with another "=", that will do the recalculation.

## 4.4. Evaluation Process

In the evaluation process the gold triples are compared with the knowledge base triples from the DBpedia 3.5.1 release.

### 4.4.1. Preprocessing

There are a few steps that have to be done before the actual evaluation. These steps are out-sourced in a preprocessor class, because it is not necessary to repeat these steps before additional evaluation runs. The relevant mapping versions of the sample articles are extracted, and DBpedia's ontology based extraction result is cleaned from triples that do not belong to the sample articles.

For each infobox mapping that is used in the sample, the relevant markup version is extracted. The Preprocessor searches in the mapping history for the version that was active at a specific date. That date is defined in the Setup class. The mapping markup is saved in a XML file for later checks whether a property has a mapping. The DBpedia ontology based extraction result contains 1.4 GB of information. It is not necessary and it would take a long time to iterate over all these triples. Therefore, all triples that have a subject URI related to a sample article are copied to an extra file. The new file has a size of only 2.6 MB, and it is easily iterable. In a last preprocessing step, some DBpedia featured triples are deleted. In case of birth and death dates, the extraction framework creates additional year properties. They are not added in every case, only for a few infoboxes. To avoid distortion of results they are deleted.

### 4.4.2. Dataset Initialization

The first evaluation step is the initialization of the data set. Here, all XML files are loaded into memory. The gold standard triples are loaded into a Jena Model[59]. Dependencies of data objects are stored in hash maps. The gold triples are mapped to their Pattern Category, to their source Wikipedia markup and infobox from which they arise. The mapped properties for each infobox are identified and for each gold triple it is checked whether it arises from a mapped property or not. At last, a list is created in which all triples, that have an intermediate node as subject are included. All these information is needed for the four evaluation steps described next.

### 4.4.3. Triple Equality Comparison (TEC)

The TEC checks for each triple in the gold standard whether the whole triple can be found in the knowledge base. The gold triples and the knowledge base triples are loaded into Jena models and are compared by the Jena Model function *contains*[60]. The TEC is searching directly for the equivalent triple. Subject, predicate and object of both triples must be perfectly the same. In the following, triples that have an accurate match are named present and triples that do not have one are called missing. If a triple matches to another, it means that it could possibly be the one equivalent triple in the other

[59] `http://jena.sourceforge.net/javadoc/com/hp/hpl/jena/rdf/model/Model.html` (retrieved 28/01/2011).

[60] `http://jena.sourceforge.net/javadoc/com/hp/hpl/jena/rdf/model/Model.html#contains\%28com.hp.hpl.jena.rdf.model.Statement\%29` (retrieved 28/01/2011).

model. In the TEC it is clear that two matching triples are there equivalents, because that is what it is asking for. But in the *Predicate Neutral Comparison* (PNC) and the *Object Similarity Comparison* (OSC), one gold triple can have more than one matching knowledge base triple. Because one of the triples items are allowed to deviate. Here, the matching knowledge base triples has to be checked for the one triple that could be meant to be the gold triple.

### 4.4.4. Predicate Neutral Comparison (PNC)

The PNC checks each gold triple, that is missing by the TEC, whether there is a matching triple in the knowledge base. The PNC has less strict test criteria as the TEC. During the PNC, a knowledge base triple matches a gold triple if it has the same subject and object as the gold triple. Thus, the predicate can deviate. The PNC can find triples with wrong predicates or imprecise mappings in which the gold standard does not concur to it.

In the first step, all triples that are marked as present in the TEC are removed from the knowledge base model. This is necessary because otherwise these knowledge base triples could rematch to different gold triples during PNC or OSC. In the next step it is iterated over the missing gold triples, which have mappings defined. It is necessary that only mapped gold triples are considered. Otherwise, it would be possible that triples without a mapping could marked as present. This would distort the result. If one of these missing gold triples has matching knowledge base triples, it is iterated over them in an inner loop. If one of those matching knowledge base triples has not count for another gold triple before, the current missing gold triple is marked as present. This metric seems to be critical because it could mark a gold triple as present that actually has no belonging knowledge base triple. An fictitious example will show this and make clear that it does not matter for the evaluation. One imagine a person who is the producer, director and author of a film. The relevant Infobox part can look like this:

```
| director = [[John Doe]]
| writer = [[John Doe]]
| producer = [[John Doe]]
```

That film would have the following three gold triples:

```
<resource/Film> <ontology/director> <resource/John_Doe>.
<resource/Film> <ontology/writer> <resource/John_Doe>.
<resource/Film> <ontology/producer> <resource/John_Doe>.
```

Furthermore, we assume that the director-property has no mapping, and thus is not extracted. The producer-property is mapped to the ontology property creator. This would be a deviation from the gold standard. It could result from outdated mappings for instance. The writer-property is mapped to an ontology property, but cannot be extracted cause of any unknown problems. Therefore, the DBpedia framework can only extract one triple from the Infobox:

<resource/Film> <ontology/creator> <resource/John_Doe>.

The TEC would not find equivalent triples. Therefore, all three gold triples are missing. Thereafter the PNC searches for matching triples. The gold triple with the director predicate is first in turn. But, there is no mapping defined for the director property and thus the gold triple is marked as missing. The second gold triple in turn would be the "writer-triple". It has a mapping defined and consequently it is iterated over the matching knowledge base triples. In this example it is only the "creator-triple", which has not counted for a gold triple before. Thus, the "writer-triple" is marked as present and the the PNC goes on with the "producer-triple". For that the "creator-triple" matches too. But it already has counted for the "writer-triple". From there, the "producer-triple" is mistakenly marked as missing. Actually, the "creator-triple" belongs to the "producer-triple". But in fact, this does not matter, because in the end, there is one mapped gold triple marked as present and two gold triples as missing, from which one is mapped. This numbers are correct and that is what counts for the evaluation result. Due to the reason that subject and object must be equivalent between knowledge base and gold standard triples, it is assured that the datatype and intermediate node statistics are not influenced.

### 4.4.5. Object Similarity Comparison (OSC)

The object similarity comparison checks whether there is a triple in the knowledge base which has a similar object to the gold standard triple. Then it calculates whether the deviation from the gold triple could be accepted as accurate. If it does, the triple is marked as present during the OSC. With this method it is possible to find wrong extracted values and rounding differences.

Here, the first step is to remove present triples from the knowledge base model too. Then, it is iterated over the missing gold triples that are not marked as present in the TEC and PNC. If there are knowledge base triples that have the same subject and the same predicate like a missing gold triple, it is iterated over these matching triples in an inner loop. The objects of the matching knowledge base triples are compared with the object of the current missing gold triple. Depending on the type of the object, different methods are used to estimate how similar the objects are. In case of numbers, a deviation of 0.1 % is accepted as accurate. The similarity of other data types is estimated by the *Levenshtein distance* (LD) [Lev66]. That is a metric to measure the difference between two sequences. If the objects are resources, only the resource-name of its representing URI is taken into account, because the other part of the URIs will be the same anyway. The LD counts the minimum number of edits that has to be made, to transfer a string into another. For normalization, the number of edits $LD$, divided by the length $l$ of the longer string, is subtracted from 1.

$$NLD = 1 - \frac{LD}{max(l_A, l_B)}$$

This leads to a scale from 0 to 1, in which 1 means that the two strings are equal. The OSC accepts a similarity of 0.8 as accurate. A value of 0.8 does not seems really similar, but it leads to good results. The following examples will give a feeling for the similarity values. With a look at the triple, extracted from the homepage property of the french city Marseille[61], one can see the slash in the end of the URL.

```
<resource/Marseille> <foaf/0.1/homepage> <http://www.marseille.fr/>
```

In the gold standard, URLs do not have a slash in the end. Whether they should, will not be discussed here. The normalised Levenshtein distance between “http://www.marseille.fr/” and “http://www.marseille.fr” is 0.9583 and the triple is marked as present. The next example is about the military service branch of Bernard Montgomery[62], who was a British Army officer. The infobox property

```
branch = [[File:Flag of the British Army.svg|23px]] [[British Army]]
```

results in the following two triples[63]:

```
<resource/Bernard_Monti> <ontology/militaryBranch> <resource/British_Army>
<resource/Bernard_Monti> <ontology/militaryBranch> <resource/Flag_of_the_British_Army.svg>
```

The first triple is the gold triple, and the second is the wrong extracted actual triple. The calculated similarity is 0.4286 and the triple is correctly marked as wrong extracted[64]. Gold triples are marked as wrong extracted if their object similarity is not accurate, during the OSC.

The OSC has to handle intermediate nodes in a special way, because the rules for creating the intermediate node URIs differ between the extraction and the gold standard.

[61] http://en.wikipedia.org/wiki/Marseille/ (retrieved 26/11/2010).

[62] http://en.wikipedia.org/wiki/Bernard_Montgomery,_1st_Viscount_Montgomery_of_Alamein/ (retrieved 26/11/2010).

[63] The URI representing Bernard Montgomery is shortened, the original is http://dbpedia.org/resource/Bernard_Montgomery,_1st_Viscount_Montgomery_of_Alamein.

[64] In this example the triple objects are resources. As mentioned above, only the resource names are considered for calculation in such a case. Here, this means “British_Army” and “Flag_of_the_British_Army.svg”.

The extraction framework builds the intermediate node URIs out from the entity name from which they arise and the values of the objects that the intermediate node holds. In the gold standard the URIs are build from the entity and the predicate name from which the intermediate nodes arise. For instance, the gold URI and the actual URI of an intermediate node resulting from the engine property of the Volkswagen Kübelwagen:

```
engine = [...] 985 cc ({{convert|23|bhp|kW|lk=on|abbr=on}}) [...]
```
[65]

First, the two extracted triples, in which the URI of the intermediate node is build from the entity name and a part of the property value are:

```
<Volkswagen_K%C3%BCbelwagen> <engine> <Volkswagen_K%C3%BCbelwagen__985_cc____%2F>
<Volkswagen_K%C3%BCbelwagen__985_cc____%2F> <displacement> "9.85E-4"^^<XMLSchema#double>.
```

Second, the gold triples representing the same information are:

```
<Volkswagen_K%C3%BCbelwagen> <engine> <Volkswagen_K%C3%BCbelwagen__engine_1>
<Volkswagen_K%C3%BCbelwagen__engine_1> <displacement> "0.000985"^^<XMLSchema#double>.
```

The gold triples URIs are build from the entity name and the predicate name, followed by a numeration. Due to this different URI naming, triples with an intermediate node as subject, cannot match each other when searching for knowledge base triples, with the same subject and predicate as the gold triple. In our example, this applies to the triples which contain the displacement predicate. If the gold triple subject is an intermediate node, the matching criteria should be more open, in such a way that <Volkswagen_K%C3%BCbelwagen__985_cc____%2F> could match to <Volkswagen_K%C3%BCbelwagen__engine_1>. The following two matching criteria can do this: First, it is necessary that both subjects are intermediate nodes. Second, the subject URI has to contain the entity name part of the intermediate node. In the example that is the string "Volkswagen_K%C3%BCbelwagen".

### 4.4.6. Snippet Completeness Check

The Snippet Completeness Check finds Snippets that are completely extracted. For each Snippet it is checked whether all triples arising from it have a mapping and could be found in the knowledge base. This check is based on the results of the triples comparison. First, it is checked whether all triples from a Snippet are marked as present in the TEC result. If that is the case, the Snippet is marked as present. If not, the Snippet is marked as missing in the TEC step. In a second step it is checked whether all triples from the Snippet are marked as present in the merged TEC and PNC results. In the possible third step, the merged three results of the TEC, PNC and OSC are searched for the

[65] Here, we look only of one part of the value.

Snippet's triples. Due to this method, the Snippet result is split into the three steps too. Like the triples, the Snippets are categorized by their Pattern Category. If the case occurs that a Snippet holds triples that are not extracted at all, the whole Snippet is marked as missing. For an example, we take the Snippet that holds the born property from the Wikipedia article of Elvis Presley. This is a simple Snippet, because it only holds one key-value pair with a single-row table:

```
Born = {{birth date|1935|1|8}}<br /><small>[[Tupelo, Mississippi]]<br />United States</small>
```

The gold triples are:

```
<resource/Elvis_Presley> <ontology/birthDate> "1935-01-08"^^<XMLSchema#date>.
<resource/Elvis_Presley> <ontology/birthPlace> <resource/Tupelo%2C_Mississippi>.
<resource/Elvis_Presley> <ontology/birthPlace> <resource/United_States>.
```

The first two triples are found as present in the TEC result, but the third triple is missing. Therefore the Snippet is marked as missing during the TEC step. Thereafter it is searched for all three triples in the TEC and PNC result. Here too, the third triple cannot be found in the present triples and the Snippet is marked as missing for the second PNC step. In the third step, the present triples from the OSC are added two the present triples from the TEC and PNC. But the third triple cannot be found as present also in this third step, because it is not extracted. Therefore, the Snippet is marked as missing in all three steps. The Snippet Completeness Check just searches for Snippets that are completely extracted. It does not matter that two of the three triples are extracted very well. This aspect is already covered in the triple-based comparisons.

# 5. Results

The QAF exports the results to an excel file. This file holds the triple comparison results, structured by the evaluation steps, the triples Pattern Categories and their XML-Schema data types. The results of the Snippet-based evaluation can be found in a second sheet. It is structured by the three evaluation steps and the Snippets Pattern Categories. Section A.1 gives a overview of the result set structure.

From the 3215 gold triples in the sample, 697 triples were found in the DBpedia dump: 642 triples during the TEC, 32 triples during the PNC and 23 triples during the OSC. Only for 1512 of the 3215 gold triples are mappings defined. Without a mapping it is not possible to extract data. Taking only the gold triples with mappings into account, the framework extracts 46 % of the gold triples. The missing 54 % have to be achieved by an improvement of the extraction framework. In this Section the most percentages are rounded to no decimal place. In the tables found in the Appendix they are given with one decimal place, therefore they can differ.

## 5.1. Results for the Pattern Categories

In this section, the results for the single Pattern Categories are discussed and suggestions for an improvement of the extraction framework and the mapping language are given.

**Plain Property[66]** This category is the one with the highest frequency of occurrence. 61 % of the Snippets and 28 % of the gold triples belong to this category. Here one can see the need to differentiate between triple-based statistics and Snippet-based statistics. Nearly two-thirds of all Snippets are in the Plain Property category, but not even one-third of all triples arise from it, because just one gold triple arises from a Snippet in this category, but the average is 2.2 triples per Snippet. This pattern has a differentiated data type characteristic. No intermediate nodes arise from this category, because of the plain structure of the pattern. 59 % of the triples have a literal, and 41 % have a resource as object. The most frequently data types are entities with 39 %, strings with 29 %, and integers with 10 %.

The completeness of this category is good. The triple-based and the Snippet-based completeness both reach 81 %. The equality of both completeness indicators is based on the one-to-one relation between Snippets and triples. The triple-based completeness is 35 percentage points higher than the average of 46 %, and the Snippet-based completeness is 18 percentage points higher than the average of 63 %. Of course, this goes back to the simplicity of the underlying pattern. The precision for this category is 96.1 %, and most errors occur during the conversation of units to their standard units. Compared with the average precision of 91.1 %, it is is a good result, especially if reconsidering that this category holds 28 % of all triples.

One could expect that the ratio of created mappings for this category is higher than the average, because it is easier to create mappings for the simple Snippets. The ratio of created mappings for Plain Property-Snippets is 58 % that is 11 points higher than the average of 47 %. But there are also Pattern Categories with a higher ratio of defined mappings. For example, Intervals with 71 %[67], and Number-Units and Coordinates with each 67 %.

**Lists[68]** Lists are the second big pattern in the sample. 14 % of the Snippets and 25 % of the gold triples came from Lists. Here too, we see the need to differentiate between triple-based statistics and Snippet-based statistics. The average number of triples per

[66] Detailed category results can be found in Appendix A tables 8 and 10.

[67] The low number of Interval-cases in the sample should be considered for the significance of this percentage.

[68] Detailed category results can be found in Appendix A tables 8 and 11.

List-pattern is 3.9. In relation to this aspect, the List-pattern is the opposite to the Plain Property-pattern. Based on the nature of the List-patterns, this result is not surprising. A List is one key-value pair that holds many single items. Lists have a clear triple structure: No intermediate nodes as subject and the objects are mainly split into 73 % entities and 26 % strings.

The triple-based completeness for Lists is 34 %, 12 percentage points less than the average. The Snippet-based completeness is only 19 %, 44 percentage points less than the average. These numbers definitely show a need to improve the List-extraction, because of the amount of data that is getting lost. The Problem is that many property values are not recognised as Lists. Therefore, only one item is extracted, mostly the first one. This conclusion is supported by the 19 % Snippet completeness. If only one item is extracted from a List-Snippet, the Snippet is marked as missing. The triple completeness is not so far away from the average, because of the 19 % completely extracted Lists that hold 1.7 triples above average. Another task is to add more possible item separators that are used in Lists. These could be slashes, minus signs or asterisks and the final *and* in an enumeration. The precision of the List-Category is 91.5 %, that is just over the average. The loss of precision in this category can be traced back to parsing errors of strings. These are not serious errors, but strings that hold the complete List instead of a single item.

Mappings are already defined for 60 % of the List-Snippets. This is more than the overall average of 47 %. Together with the low completeness indicator, this leads to the conclusion that writing new List mappings would not be efficient at the moment. The prior work should be the improvement of the List extraction.

**One-Property Tables**[69] The One-Property Table pattern occurs only in 3 % of all Snippets, but it holds proportionally much triples. To be precise, 14 % of all triples come from One-Property Tables. On average 9.1 triples arise from one Snippet. One-Property Tables produce many intermediate nodes. Only 39 % of the triples subjects are entities and 63 % of them just connect the intermediate nodes with their source entity[70]. Thus, they have an intermediate node as object. Only 14 % of the triples from this category are not part of an intermediate node construction, that means they have not an intermediate node as subject or object. This triples mainly have entities or *gYear*[71]

[69] Detailed category results can be found in Appendix A tables 8 and 12.

[70] This means the entity build from the Wikipedia article URL. For instance look at the second example in section 3.4.3. The first gold triple, <resource/Cowboy_Bebop> <ontology/licensor> <resource/Cowboy_Bebop__licensor_1>, connects the source entity Cowboy_Bebop with its first licensor intermediate node.

[71] gYear is a XML Schema datatype for year dates.

information as objects. The big part, 61 % of the triples subjects, are intermediate nodes. They too, have mainly entities, 67 %, and gYears, 20 %, as objects. This leads to the conclusion that One-Property Tables are generally used to connect entities with each other or entities with year specifications.

The triple-based completeness is only 5 %, and not one single Snippet is extracted completely. This is not surprising at all, because there are no specialised mappings written for this pattern. For some properties which contain this category pattern, mappings are written for only one item in the table. Therefore, the other items are not even tried to extract. This leads to the 5 % extracted triples. Because of the heuristic that checks triples for mappings, the percentage of 54 % defined mappings for triples is not convincing for this Pattern Category. The heuristic just searches for a mapping written for the relevant infobox property. If there is a mapping, all triples that arise from this property are marked as mapped. Therefore we can only take the restrictive Snippet-based mapping indicator into account, which is 0 %. The precision of 32.5 % is by far the lowest of all categories. This low percentage is caused by the single item picks. These items are extracted well, but compared to the gold standard, they should be part of an intermediate node, which contains all information of the table. It might be questionable, that these extracted single items influence the precision in such a huge way. If we exclude this effect, the precision rises to 92.9 %, with errors during the unit conversion.

To be fair, the already defined mappings are not useless, but they are not complete, they only pick some items. For reappearing standard tables[72], the mapping language should provide new solutions. The tables from which no intermediate nodes arise[73], because each item can be mapped to a ontology property, can already be mapped completely. Here, it is just the need to write more and complete mappings. Another important aspect is, that mapping-authors make the mistake to interpret tables as lists. This leads to a loss of triples, because using a list-mapping, the author considers only one column of a table. For instance, in the second example in section 3.4.3, only the licensors are extracted, without the country relation.

**Multi-Property Tables[74]** Due to the similar pattern structure, the characteristic of this Pattern Category is like the One-Property Tables. Only 6 % of all Snippets belongs to this category, but it holds 19 % of the gold standard triples. On average 6.8 triples arise from one Snippet. Mainly all triples from MulitPropertyTables are intermediate node constructions. Thus means 94 % of the arising triples have an intermediate node as

[72] For instance, see the second example in section 3.4.3.

[73] For instance, see the first example in section 3.4.3.

[74] Detailed category results can be found in Appendix A tables 8 and 14.

subject or object. The 78 % of triples with an intermediate node as subject and without an intermediate node as object, hold the original extracted information. 30 % of this objects are entities, 26 % integers, 20 % gYears, 10 % strings and 8 % doubles. This data type distribution is the result of the many statistic tables, which belongs to the Multi-Property Table pattern.

The completeness result is equivalent to the One-Property Table pattern. The triple completeness is only 9 % and the Snippet completeness is 0 %. MulitPropertyTables are not really extracted right now. Single key-value pairs are caught by the extraction, but the whole relation to other table items got lost. Related to the example in 3.4.4, the clubs, John Charles played for, are extracted. But they aren't linked to the period, and the caps and goals are missing completely. The precision of 88.9 % is not really significant for this Pattern Category, because of the same intermediate node problem described above for the One-Property Tables. Without this effect the precision would be 100 %, which also is not really meaningful considering the low number of cases. Only 8 triples are marked as present, and they are all caught in the Object Similarity Comparison. Therefore, the objects are not equal, they are similar in an adequate way. In general we can say that this category is not extracted.

The mapping language does not provide a comprehensive solution for this category. If there are mappings, they are defined mostly for only one key-value pair of the table. Not for one Snippet a complete mapping is defined, and the percentage of defined mappings for triples is 15 %. Here too, this percentage is to handle with care, due to the same heuristic problem described above for the One-Property Tables. But for Mulit-Property Tables, this effect has not such a big consequence for the percentage, due to two facts: Generally, the table items are split to different properties. Therefore, not all triples arise from the Snippet are marked as mapped, if only one item has a mapping. Especially, since often each table-item has its own property, this effect is irrelevant for such a table.

**Coordinates**[75] About one of five Wikipedia articles have coordinates given. Though only 1.2 % of all Snippets are coordinates and only 1.7 % of all gold triples arise from them, this category is important, because of the usefulness of coordinates. This category has a really clear node structure. From one Snippet arises three triples, one triple with a string as object, which contains a *GeoRSS Point*[76] and two triples with float data types, which hold the latitude and longitude in the *WGS84 Geo Positioning* vocabulary[77].

Coordinates are the best extracted category. For the sample of this study, the triple-

[75] Detailed category results can be found in Appendix A tables 8 and 15.

[76] `http://www.georss.org/` (retrieved 10/01/2011).

[77] `http://www.w3.org/2003/01/geo/wgs84_pos` (retrieved 10/01/2011).

based and the Snippet-based completeness are 100 %. The very same for the precision of the coordinate extraction, it is 100 %. This depends on the amount of work that is already put into the extraction framework's coordinate parser. Here we have to give a special mention to Christian Becker, who invented DBpedia Mobile. It is a location-centric DBpedia client application for mobile devices [BB08]. An amount of geographical information is required for this program and thus he puts special effort to extract them.

Mappings are written for two-thirds of all coordinates in the sample. Therefore, only more mappings must be written to increase the amount of geographic information. The mapping language already provides the needed specifications to create all coordinate mappings for the sample.

**Number-Units**[78] With regard to the triple amount, this category is a small one, just 2.4 % of the gold triples belong to it. But 5.2 % of all Snippets are Number-Units. This is the fourth rank of the most categories. From each Snippet one triple arises and all of them have an entity as subject and a double as object.

Number-Units are relatively good extracted. This is caused by the complex Unit-Parser of the extraction framework. The triple and the Snippet completeness are both 69 %. The equality of both completeness indicators is based on the one-to-one relation between Snippets to triples. The good completeness result is clouded by a low precision value of 85.4 %. The precision suffers from mistakes during the conversion of units to their standard units.

Compared to the average, there are much mappings defined for the category Number-Units. 67 % of all triples and Snippets have mappings defined. This can be explained by the simplicity of writing Number-Unit-mappings.

**Intervals**[79] This Pattern Category is a really small one. It is so small that the resultant percentages might not be representative. Only 31 triples (1.0 %) arise from this pattern and only 16 Interval-Snippets (1.1 %) are in the sample. This category produces no intermediate nodes, and 61 % of the triple-objects are gYears. The other 39 % are numbers and other date formats.

The few Intervals are relatively good extracted, the triple-completeness is 73 %, and the Snippet-completeness is 64 %. The precision is 100 %, but we have to remind the small number of Interval-cases. Another fact that must considerd is that an Interval only belongs to this category, if it is not part of a list or table. A list of Intervals that is not extracted correct will decrease the List completeness and precision.

[78] Detailed category results can be found in Appendix A tables 8 and 16.

[79] Detailed category results can be found in Appendix A tables 8 and 17.

For Intervals are relatively much mappings defined, 71 % of all Interval-triples, and 69 % of all Interval-Snippets are mapped. Mostly, these are the career durations of athletes, and politicians.

**Open Properties**[80] The percentage of triples that arise from this category and the percentage of Snippets that belong to it are both 4 %. Most of the triples (70 %) are part of an intermediate node construction. This is based on the difficulty of this pattern, in which the key of a property's key-value pair is variable. If the key is not restricted to any issue, it is not possible to map the infobox property to a ontology property and an intermediate node is required. The most objects of triples with an intermediate node as subject are strings with 47 % and entities with 34 %. The Open Property-pattern is used for individual properties of an entity that is described by the extracted Wikipedia article. Therefore, these properties are not standardised by an infobox property.

Just 14 triples from this category have a mapping defined. This corresponds to 10 % defined mappings. The triple-completeness, for that small number of cases is 29 %. Only 7 Snippets have complete mappings, which is just 13 % of all Open Property-Snippets. Here too, the Snippet completeness of 43 %, is not really convincing due to the low number of cases. The same applies for the precision of 100 %.

To improve the extraction of this category, the first thing would be to expand the mapping language for this kind of pattern.

**Open Property Tables**[81] This category contains rare special cases of a table design. Only two Snippets with 26 triples belong to this category, which is below 1 % of all triples and Snippets. The pattern is used for tables with individual statistics of an entity. No mappings are written for this category, and therefore one cannot say anything about the completeness and precision.

It would not be efficient to put some effort in this category right now, because of the rare occurrence. But the frequency of the pattern should be observed. If it is used more often in the future, it could be useful to extract it because of the high amount of information that are stored in this kind of tables.

**Internal Templates**[82] The triple and Snippet occurrence from this pattern are both 4 %. Most of the triples are entity triples, only 28 % have an intermediate node as subject. The 72 % triples with an entity as subject have the following objects: 41 %

[80] Detailed category results can be found in Appendix A tables 8 and 18.

[81] Detailed category results can be found in Appendix A tables 8 and 19.

[82] Detailed category results can be found in Appendix A tables 8 and 20.

entities, 23 % URLs, 18 % intermediate nodes, and 17 % strings. The many URLs are noticeable. They come from infobox properties that contain IDs from which links to external databases are generated. For example the *CAS number*[83] of chemicals. The infobox *drugbox*[84] generates a link to the chemical's page of the U.S. National Library of Medicine[85]. From the CAS number property, two gold triples arise, one with the CAS number and one with the link to the chemical's page of the U.S. National Library of Medicine. This leads to the relatively high ratio of URLs.

The completeness of this category is low, just 9 % of the gold triples are extracted and not one Snippet is extracted completely. The reason is that the URLs are not extracted, only the IDs. Another problem is already described in Section 3.4 and the basis of this category. Snippets belong to this category when it is not sufficient to extract the markup, instead the rendered infobox must be taken into account to extract the gold standard. But exactly that feature is not invented and therefore the completeness of this category is not satisfying. The precision of the Internal Template extraction is 83 %. The completeness and precision for this category are not really significant, due to the low number of cases.

Suggestions to improve the Internal Template extraction can be found in section 5.2.6.

**Merged Properties**[86] This pattern is rarely used. Less than 1 % of the gold triples arises from this category, and only 1 % of the Snippets belong to it. Merged Properties have a basic node type structure. No intermediate nodes arise and just two types of data types are present as objects. First, there are 69 % dates and second 31 % doubles. The reason is that single infobox properties for year, month and day are merged to dates. The doubles arise from lenght-values that are specified with two infobox properties. Mostly they are used for the person height (feet and inches are merged to metres) used in infoboxes of American athletes.

Here too, the small number of cases relativises the conclusions for the category. The triple and the snippet-based completeness are 57 %, with a precision of 80 %. The loss of precision arises from merging- or converting-errors, and completeness suffers from wrong mappings. These are mappings which lead to a single extraction of the year, month and day. Instead, the merging mappings must be used to extract the gold triples.

Mappings are written for 54 % of the triples and Snippets. Here too, the wrong

[83] CAS Registry Numbers are unique identifiers for every chemical described in the open scientific literature.

[84] `http://en.wikipedia.org/wiki/Template:Drugbox/` (retrieved 13/01/2011).

[85] Here, the NLM link for aspirin: `http://www.nlm.nih.gov/cgi/mesh/2009/MB_cgi?term=50-78-2&rn=1/` (retrieved 13/01/2011).

[86] Detailed category results can be found in Appendix A tables 8 and 13.

mappings are counted in. For an improvement of the Merged Properties-extraction, the existing mappings should be checked for the use of merging mappings and new mappings should be written.

## 5.2. Tasks for improving the DBpedia framework

This section point out several tasks needed to improve the DBpedia extraction framework. They are mentioned here because they go beyond the simple fixing of errors or writing new mappings. Some of these tasks are mayor extensions, which could take weeks of work. The *Entity Recognition* introduced in Section 5.2.3 and the *Ontology Property Finder* in Section 5.2.4 would necessitate a new DBpedia data set. Because they would use heuristics, which would link entities, additional to the links of Wikipedia. This data would not be a pure representation of Wikipedia anymore.

### 5.2.1. Combined infoboxes

There are infoboxes made up from sub infoboxes. Three different types of these combined infoboxes were found. The first type has nested infoboxes. It starts with a {|-tag followed by the opening infobox. Then sub infoboxes that contain the actual key value pairs follow. The nested infoboxe ends with a closing |}-tag. Examples for these type are the Infobox aircraft[87] or the Infobox ship[88], which is shown below:

```
{| {{Infobox ship begin}}
{{Infobox ship image
| Ship image = [[File:Uss bang.jpg|center|300px|USS Bang]]
| Ship caption = USS Bang
{{Infobox ship career
| Ship country = US
| Ship flag = {{USN flag|1972}}described
| Ship name = USS "Bang" (SS-385)
}}
{{Infobox ship characteristics
| Ship displacement = 1,526 tons surfaced<br>2,391 tons
| Ship length = 311 ft 8 in (95 m)
}} |}
```

The second type uses infobox properties that have sub infoboxes as values. One example is the Chembox[89]:

```
{{Chembox
ImageFile =
```

[87] http://en.wikipedia.org/wiki/Template:Infobox_aircraft_begin (retrieved 13/01/2011).

[88] http://en.wikipedia.org/wiki/Template:Infobox_ship_begin (retrieved 13/01/2011).

[89] http://en.wikipedia.org/wiki/Template:Chembox (retrieved 13/01/2011).

```
| IUPACName =
| PIN =
| OtherNames =
| Section1 = {{Chembox Identifiers | CASNo = | PubChem = | SMILES = }}
| Section2 = {{Chembox Properties | Formula = | MolarMass = }}
| Section3 = {{Chembox Hazards | MainHazards = | FlashPt = | Autoignition = }}
}}
```

The third type is more complex. It is a combination of sub infoboxes where some of them describe different entities. This concerns the Infobox animanga[90]. It starts with a header which is followed by infoboxes to different mediums or other episodes of the same manga. Each medium needs his own subject URI because they cannot be attached to the same entity. The end of the combined infobox is set with a footer infobox.

```
{{Infobox animanga/Header | name = | image = | caption = | genre = }}
{{Infobox animanga/Print | type = manga | title = | author = | illustrator = }}
{{Infobox animanga/Video | type = tv series | title = | director = }}
{{Infobox animanga/Footer}}
```

The extraction should handle such combined infoboxes by taking the sub infoboxes into account. In the current status, the sub infoboxes of the first two types are deleted and most information is lost. For the third type, a possible solution could be to write mappings for all of the sub infoboxes and link them together. Some of the properties must be added to all entities that have its seeds in the combined infobox. For example the *genre* from the header infobox.

Table 1: Combined infoboxes

| Template | Times used | Type |
|---|---|---|
| Infobox Ship Characteristics | 20566 | 1 |
| Chembox | 6498 | 2 |
| Infobox aircraft type | 6384 | 1 |
| Infobox animanga | 3437 | 3 |

The list cannot claim to be complete.

### 5.2.2. Various infoboxes to one entity

This Task is related to the one above. But here there are various distinguishable infoboxes in one Wikipedia article that deliver information to the same entity. The framework can

[90] http://en.wikipedia.org/wiki/Template:Infobox_animanga (retrieved 13/01/2011).

already handle different infoboxes in the same Wikipedia article, but just if that infoboxes describe different entities. A new URI will be created for each infobox on a Wikipedia page. But if the information from the additional infobox belongs to the same entity, the same URI should be used. That is nott a big problem, in fact the only reason this point is mentioned here is the “Weather_box”, formerly known as “Infobox weather” and the “climate chart” template. Actually these templates were not extracted, because they do not have a name property from which a URI could be built up.

Table 2: Templates that deliver additional information to entities

| Template | Times used |
|---|---|
| Weather box | 2471 |
| Climate chart | 711 |

The list cannot claim to be complete.

### 5.2.3. Entity recognition

It is really common to link only the first appearance of an entity in an Wikipedia article. For example, if a person turns up more than once in an infobox, it is commonly linked only the first time to the Wikipedia article describing that person. This leads to many missing statements, because the mapping demands a person-entity for each of the infobox properties in which the person is mentioned. But only the first time this requirement is fulfilled. The second time the person is mentioned, only a string is given. A look at the infobox of the Thunderbirds TV series will clarify it:

```
{{Infobox television
| show_name = Thunderbirds [...]
| creator = [[Gerry Anderson]]<br>[[Sylvia Anderson]]
| writer = Gerry Anderson<br>Sylvia Anderson<br> [...]
| voices = Sylvia Anderson<br> [...]
}}
```

In the best-case the following five triples are extracted:

```
<resource/Thunderbirds_%28TV_series%29> <ontology/creator> <resource/Gerry_Anderson>
<resource/Thunderbirds_%28TV_series%29> <ontology/creator> <resource/Sylvia_Anderson>
<resource/Thunderbirds_%28TV_series%29> <ontology/writer> <resource/Gerry_Anderson>
<resource/Thunderbirds_%28TV_series%29> <ontology/writer> <resource/Sylvia_Anderson>
<resource/Thunderbirds_%28TV_series%29> <ontology/voice> <resource/Sylvia_Anderson>
```

Actually, the last three statements get lost. This is only one example for the need of a named entity recognition. The framework should be able to search for an entity if the mapping requirement is not fulfilled. Nearly in every case there are useful extra information given from the mapping, so that the entity recognizer has already some restrictive information. For the example above, it would be enough to compare the string with the labels of already extracted Wikipedia links. The named entity recognition is also useful to check for the correctness of extracted statements. Another example right from the Thunderbird TV series:

```
| country = [[Television in the United Kingdom|United Kingdom]]
```

This infobox key value pair will cause an mistake. The extracted statement is:

```
<resource/Thunderbirds_%28TV_series%29> <ontology/country> <resource/Television_in_the_United_Kingdom>
```

It is obvious that the resource “Television_in_the_United_Kingdom” is not the series country of origin. With a little help of the Wikipedia link label “United Kingdom” it should be possible to find the right entity.

The most frequently triples are those with an entity as object. 1409 of the 3215 triples are count in this category. But only 23.6 % are extracted and for 43.8 % of the missing triples have mappings defined. By ignoring the triples with an intermediate node as subject, the percentage of extracted triples is rising to 47.5 %. It is astonishing that the percentage of the missing triples that have mappings defined is rising to 51.5 % also. Actually, one would expect that the triples with an entity as subject are extracted much better than the intermediate node triples. So it is, not one intermediate node triple with an entity as object is extracted. But the reason for this result is caused in the missing or incomplete mappings for intermediate nodes and not in the more complex way to extract them. These 51.5 % missing pure entity triples[91] with defined mappings are the potential of a good named entity recognition. Related to the whole extraction, the potential is about 12 % more successful extracted triples with a perfect entity recognition. If there would be working mappings for the intermediate nodes, the potential would rise, because 27 % of all gold triples have an intermediate node as subject.

### 5.2.4. Ontology Property Finder

Key value pairs that are counted among the Pattern Categories Open Property and Open Property Table always produce intermediate node triples. To keep the number of them low, an Ontology Property Finder would be helpful, because some of the free labeled properties that are used in the both corresponding Pattern Categories always exist as ontology properties. For example we look again in the infobox at the Cape Town Wikipedia page[92]:

[91] Subject and object are entities.

[92] http://en.wikipedia.org/wiki/Cape_town (retrieved 24/11/2010).

```
| leader_title = [[Mayor of Cape Town|Mayor]]
| leader_name = [[Dan Plato]]
```

Normally a set of intermediate node statements would be extracted:

```
<resource/Cape_Town> <ontology/openProperty> <resource/Cape_Town__openProperty_1>
<resource/Cape_Town__openProperty_1> <ontology/key> <resource/Mayor_of_Cape_Town>
<resource/Cape_Town__openProperty_1> <ontology/value> <resource/Dan_Plato>
```

But the DBpedia resource Cape_Town belongs to the ontology classes City and PopulatedPlace. The class PopulatedPlace has the ontology property *mayor*[93]. So, with a Ontology Property Finder, it would be possible to create the simpler triple:

```
<resource/Cape_Town> <ontology/mayor> <resource/Dan_Plato>
```

### 5.2.5. Extraction of additional display and design templates

There are several small templates used to display data in a consistent way. Till now, the framework can handle a few of it, like the age templates[94] or the convert templates[95]. But there are further useful templates that are frequently used. The advantage of these templates is that they keep the data structured and that they can link accurate to the relevant entities, because of the template intern validation. Other interesting templates are only for designing list or tables.

Table 3: Display and design templates

| Template | Times used |
|---|---|
| Flagicon | 1753200 |
| Flag | 232783 |
| Flagcountry | 71575 |
| Colorbox | 5929 |
| Collapsible list | 338 |

The list cannot claim to be complete.

### 5.2.6. Rendered page extraction to solve internal templates

A new functionality should be implemented that allows the framework to extract data not only from the MediaWiki markup, but also from the rendered Wikipedia page. This is needed to solve the internal template problem described in 3.4.10. There may be other ways to get the covert information. But an extraction from the rendered Wikipedia page

[93] *http://mappings.dbpedia.org/index.php/OntologyProperty:Mayor* (retrieved 24/11/2010).
[94] http://en.wikipedia.org/wiki/Wikipedia:Age_calculation_templates (retrieved 24/11/2010).
[95] http://en.wikipedia.org/wiki/Template:Convert (retrieved 24/11/2010).

could also help by extracting tables or cryptic strings. It may be easier to extract the rendered content than to adopt the rendering process in the frameworks parsers.

The Problem with these data integration from other templates is that the information written in the Wikipedia code of a page could be outdated since there is no need for an manual update of that page anymore. The Wikipedia page will show the up to date information although the code might be out of date. The framework extracts from the code and not from the rendered page. The infoboxes in Table 4 uses information from other templates.

Table 4: Infoboxes that get information from other Templates

| Template | Times used | Source template |
|---|---|---|
| Infobox German location | 12763 | Population Germany |
| Infobox District DE | 340 | Population Germany |
| Infobox German Regierungsbezirk | 32 | Population Germany |
| Infobox German State | 16 | Population Germany |
| Infobox municipality | ca. 290 | Population Swedish municipality |
| Infobox municipality | ca. 290 | Area Swedish municipality |

The list cannot claim to be complete.

### 5.2.7. Mapping extensions

There are different patterns in which information is filled in Wikipedia infoboxes. The Pattern Categories defined in these work try to differ these patterns by there kind in which they use the key value system of the infoboxes. To extract these Pattern Categories correct, there is a need for special defined mappings that take the style of the patterns into account. Examples are given in 3.4.

## 5.3. Second roll after improvement

DBpedia 3.6 was released at 17th January 2011. Some tasks discovered in this study are implemented in the new version of the extraction framework. For example:

- The List parsing has been improved by adding new item separators[96].
- Flag templates are parsed[97].
- Reappearing strings in Infoboxes, which are linked only once, are extracted now[98].

[96] For the result of the evaluation, see Lists in section 5.1.

[97] Task is given in section 5.2.5.

[98] For further details, see section 5.2.3.

To make the two evaluation results comparable to each other the same Wikipedia dump are extracted with both extraction framework versions. This is necessary, because the Wikipedia dump used for DBpedia 3.6 has changed much from the dump used for DBpedia 3.5.1. Triples extracted from the new dump could not match the gold standard triples from the earlier dump. Nevertheless, there is one point we cannot smooth out. The mappings have changed in the meanwhile. The extraction framework cannot access older mapping versions right now. Only when this is fixed, the Quality Assessment Framework could perfectly assign quality changes to an improvement of the extraction framework, or to improved mappings.

But for this second run, one can account the improved extraction framework for the better evaluation results, because the amount of mappings has not really changed. By running the evaluation with the DBpedia 3.5.1 data set, 1512 mapped triples could be found. In the second run, with the new framework and the current mappings[99], 1532 mapped triples were found. These 20 more mappings will influence the results in a marginal amount only.

The overall triple completeness has increased from 46 % to 61 %, and the Snippet completeness from 64 % to 72 %. The precision increased from 91.2 % to 92.3 %. The biggest part of this improvement fall back to the upgraded List parsing. The triple completeness for Lists increased from 33 % to 76 %. The amount of successful extracted List-triples has more as doubled, from 156 to 366. Almost all of this new extracted List-triples have a resource as object. Thus, the completeness of this triples rise from 33 % to 81 %. The completeness of List-triples with a literal as object remain constant at 38 %. This goes back to the new List parsing that now uses an improved Wikipedia link parser. The extraction of reappearing strings, which are linked only once, also influence this result. The example in Section 5.2.3 can illustrate this relation. Resources in Lists are much better parsed in the new extraction framework, but the parsing of Literals can still improved. The other Pattern Category that highlight the improved extraction framework is the Plain Property category. The completeness increased from 81 % to 84 %, that is not much compared to the improved List-category, but it is the biggest category, therefore these 3 percentages will result in much more nice extracted triples. Table 22 and Table 21 compare the completeness and precision results of the first and second run.

[99] Extraction from the 18th January 2011.

# 6. Conclusion

The DBpedia Quality Assessment Framework (QAF) is the first approach to an integrated quality evaluation of the DBpedia knowledge base. It compares the DBpedia 3.5.1 ontology-based extraction result with a manually created gold standard for a sample of Wikipedia articles. As Wikipedia is the source of DBpedia's knowledge base, the information quality of Wikipedia is highly responsible for DBpedia's. Therefore, this study puts the focus on the completeness of the ontology-based extraction result compared to Wikipedia, and the precision of the extraction framework.

The gold standard contains 3215 RDF triples that can be extracted from the sample Wikipedia articles in the best-case. To get fine-grained results, the gold triples are organised in 11 different Pattern Categories based on the design of the underlying infobox structure. The average amount of triples that arise from these Pattern Categories differ from only one triple arising from a *Plain Property* to 13 triples arising from an *Open Property Table*. For this reason, two indicators are used to explain the completeness of a Pattern Category. First, the triple-based indicator that considers all triples arising from a category. Second, the Snippet-based indicator, which requires that all gold triples arising from one pattern are found in the knowledge base to count the Snippet as complete.

The completeness indicators reveal strong distinctions according to the Pattern Category, the data type of triple objects and the node type of triple subjects. The best extracted Pattern Category are the Coordinates. Its completeness indicators are both 100 % for the sample articles. On the contrary, only 5 % of the triples that arise from *One-Property Tables* are extracted. Triples that have an literal as object are extracted in 69 % of all cases, in contrast, only 43 % of the triples with an resources as object are extracted. The influence of the subject type of a triple is even more critical for the extraction. 53 % of the triples with an entity as subject are extracted, but only 4 %, if the subject is an intermediate node. This results show, that there are some patterns, like the table patterns or the *Open Property* patterns, which are not sufficient covered by the extraction framework. DBpedia's mapping language does not provide a mapping specification for them. Another task is the handling of intermediate nodes in generally. Its URIs should be generated more consistent and maybe it is advisable to give them there own name space. That would help to differentiate them from "real" resources.

The precision indicator seems adequate for the Pattern Categories that are covered by the extraction framework. The overall precision is 91.1 %. The *Plain Property* category has a really high precision of 96.1 %. As expected, the *Number-Unit* pattern has a lower value of 85.4 % that is caused by the conversion of units.

During the creation of the gold standard many tasks are detected that could improve the quality of the extraction framework. These are simple tasks like parsing new templates or mayor extensions like the entity recognition described in Section 5.2.3.

The QAF can act as a benchmark for further DBpedia releases. In Section 5.3 the extraction result of DBpedia 3.5.1 is compared with the extraction framework version of the DBpedia 3.6 release. Here, the improvement of parsing the List-pattern is ascertainable between the two versions. The triple-based completeness indicator for Lists increased from 33 % to 66 % and the overall triple-based indicator increased from 46 % to 61 %.

The small sample of 75 Wikipedia articles and the absence of mappings for a few categories disallow a significant conclusions for all Pattern Categories, especially for *Merged Properties*, *Open Properties* and *Open Property Tables*. For further evaluation, the sample should be enhanced. But nonetheless, the evaluation results show that the Quality Assessment Framework is able to highlight leaks of the extraction framework and is able to point out improvements.

# References

[AMP06] Ofer Arazy, Wayne Morgan, and Raymond Patterson. Wisdom of the crowds: Decentralized knowledge construction in wikipedia. Proceeding of the 16th Workshop on Information Technologies & Systems (WITS '06), December 2006.

[BB08] Christian Becker and Christian Bizer. Dbpedia mobile: A location-enabled linked data browser. 1st Workshop about Linked Data on the Web (LDOW2008), Beijing, China, April 2008.

[BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web & Information Systems*, 5(3):1–22, 2009.

[Biz07] Christian Bizer. *Quality-Driven Information Filtering.* VDM Verlag Dr. Müller, 2007.

[BL06] Tim Berners-Lee. Linked data - design issues. `http://www.w3.org/DesignIssues/LinkedData.html`, July 2006.

[BLK+09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auser, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7:154–165, 2009.

[Bod05] David Boddy. *Management: An introduction.* Prentice Hall, 2005.

[BWPT98] Donald Ballou, Richard Wang, Harold Pazer, and Giri Tayi. Modeling information manufacturing systems to determine information product quality. *Management Science*, 44(4):462–484, 1998.

[Che05] Thomas Chesney. An empirical examination of wikipedia's credibility. *First Monday - http://131.193.153.231/www/issues/issue11_11/chesney/index.html*, 2005.

[Cod90] Edgar Codd. *The Relational Model for Database Management: Version 2.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[Enc06] Encyclopædia Britannica, Inc. Fatally flawed - refuting the recent study on encyclopedic accuracy by the journal nature. `http://corporate.britannica.com/britannica_nature_response.pdf`, March 2006.

[GCM$^+$08] John F. Gantz, Christopher Chute, Alex Manfrediz, Stephen Minton, David Reinsel, Wolfgang Schlichting, and Anna Toncheva. The diverse and exploding digital universe. Technical report, IDC White Paper - sponsored by EMC http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf, March 2008.

[Gil05] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438:900–901, December 2005.

[JBM08] Lei Jiang, Alex Borgida, and John Mylopoulos. Towards a compositional semantic account of data quality attributes. In Qing Li, Stefano Spaccapietra, Eric Yu, and Antoni Olivé, editors, *Conceptual Modeling - ER 2008 27th International Conference on Conceptual Modeling, Barcelona, Spain*, volume 5231 of *Lecture Notes in Computer Science*, pages 55–68. Springer-Verlag Berlin Heidelberg, Oktober 2008.

[JL10] Sara Javanmardi and Cristina Lopes. Statistical measure of quality in wikipedia. 1st Workshop on Social Media Analytics (SOMA '10), Washington, DC, USA., July 2010.

[Jur74] Joseph Juran. *The Quality Control Handbook*. McGraw-Hill, New York, 3rd edition, 1974.

[Lev66] Vladimir Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[Mil08] Mike Miliard. Wikipediots: Who are these devoted, even obsessive contributors to wikipedia? Salt Lake City Weekly, February 2008.

[MMM04] Frank Manola, Eric Miller, and Brian McBride. Rdf primer. W3C Recommendation http://www.w3.org/TR/rdf-primer/, February 2004.

[MSV$^+$02] Massimo Mecella, Monica Scannapieco, Antonino Virgillito, Roberto Baldoni, Tiziana Catarci, and Carlo Batini. The daquincis broker: Querying data and their quality in cooperative information systems. In Stefano Spaccapietra, Sal March, and Karl Aberer, editors, *Journal on Data Semantics I*, volume 2800 of *Lecture Notes in Computer Science*, pages 208–232. Springer-Verlag Berlin Heidelberg, 2002.

[Nat06] Nature Publishing Group. Encyclopaedia britannica and nature: a response. http://www.nature.com/press_releases/Britannica_response.pdf, March 2006.

[Nau02] Felix Naumann. *Quality-Driven Query Answering for Integrated Information Systems.* Springer Verlag, Heidelberg, 2002.

[NR00] Felix Naumann and Claudia Rolker. Assessment methods for information quality criteria. In *In Proceedings of the International Conference on Information Quality*, pages 148 – 162, 2000.

[PLW02] Leo Pipino, Yang Lee, and Richard Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, April 2002.

[Sur04] James Surowiecki. *The wisdom of crowds.* Anchor Books, New York, 2004.

[Tan07] Bill Tancer. Look who's using wikipedia. TIME Magazine - Time Inc. `http://www.time.com/time/business/article/0,8599,1595184,00.html`, March 2007.

[Wan98] Richard Wang. A product perspective on total data quality management. *Communications of the ACM*, 41(2):58–65, February 1998.

[Woo07] Alex Woodson. Wikipedia remains go-to site for online news. Reuters `http://www.reuters.com/article/2007/07/08/us-media-wikipedia-idUSN0819429120070708`, July 2007.

[WS96] Richard Wang and Diane Strong. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–33, 1996.

[WW96] Yair Wand and Richard Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):96–95, November 1996.

# A. Appendix

## A.1. How to read the Excel file

The output excel file holds two sheets with lots of numbers and percentages. The belonging tables could be found in the appendix. This section will give information about the meaning of this numbers.

The sheet with the triple-based result contains six sections. Each section itemises its Pattern Category results, the node and data type statistics of its triples and the number of mappings defined. The first section is the overview of the gold standard triples. Here one can see from which Pattern Categories the gold triples arising and which data types they have. The second section contains the triple-based completeness indicator. It is summed up from the third to fifth section, which are the results of the TEC, PNC and OSC. In all four sections, the completeness is given in percentages. The last section on the table's bottom shows wrong extracted triples.

The sheet with the Snippet-based results has the same sections. An exception is the section with the wrong extracted triples. The Snippets have no data types and therefore are only itemised by the Pattern Categories.

## A.2. Sample Structure

Table 5: Infoboxes and Articles in the Sample

| Rank | Infobox Template | Times used | Wikipedia Article in the sample |
|---|---|---|---|
| 1 | Infobox_settlement | 185391 | Cape_Town |
| 2 | Taxobox | 145820 | Subspecies_of_Canis_lupus |
| 3 | Infobox_album | 92242 | The_Dark_Side_of_the_Moon |
| 4 | Infobox_football_biography | 51306 | John_Charles |
| 5 | Infobox_film | 48768 | Night_of_the_Living_Dead |
| 6 | Infobox_musical_artist | 46947 | Elvis_Presley |
| 7 | Infobox_French_commune | 36789 | Marseille |
| 8 | Infobox_officeholder | 36472 | Clement_Attlee |
| 9 | Infobox_actor | 34702 | John_Wayne |
| 10 | Infobox_single | 32694 | Bohemian_Rhapsody |
| 11 | Infobox_company | 26468 | Apple_Inc. |
| 12 | Infobox_person | 25867 | Charles_Dickens |
| 13 | Infobox_NRHP | 23231 | Manzanar |
| 14 | Infobox_ship_career | 21063 | German_submarine_U-238 |
| 15 | Infobox_book | 20481 | Chapterhouse:_Dune |
| 16 | Infobox_ship_characteristics | 19661 | Amsterdam_%28VOC_ship%29 |
| 17 | Geobox | 18353 | Delaware_River |
| 18 | Infobox_football_biography_2 | 16030 | Bobby_Charlton |
| 19 | Infobox_television | 15654 | Thunderbirds_%28TV_series%29 |
| 20 | Infobox_radio_station | 15616 | WKHX-FM |
| 21 | Infobox_video_game | 15467 | Final_Fantasy_IX |
| 22 | Infobox_UK_place | 14969 | Glasgow |
| 23 | Infobox_Indian_jurisdiction | 14469 | Hyderabad%2C_India |
| 24 | Infobox_military_person | 13482 | Bernard_Montgomery |
| 25 | Infobox_German_location | 12718 | Frankfurt_am_Main |
| 26 | Infobox_planet | 12348 | Uranus |
| 27 | Infobox_school | 12127 | Wah_Yan_College%2C_Hong_Kong |
| 28 | Infobox_football_club | 11492 | Real_Madrid_C.F. |
| 29 | Infobox_MLB_player | 11240 | Babe_Ruth |
| 30 | Infobox_university | 10673 | University_of_Southern_California |
| 31 | Infobox_airport | 9520 | John_F._Kennedy_International_Airport |
| 32 | Infobox_military_unit | 9391 | United_States_Air_Force |
| 33 | Basketballbox | 9239 | 2006_NBA_Finals |
| 34 | Infobox_road | 8913 | New_Jersey_Turnpike |
| 35 | Infobox_writer | 8728 | Enid_Blyton |
| 36 | Infobox_scientist | 8471 | Ernest_Rutherford |
| 37 | Infobox_mountain | 8418 | Mount_Hood |
| 38 | Infobox_lake | 8164 | Lake_Huron |
| 39 | Infobox_river | 8102 | Daugava_River |
| 40 | Infobox_Italian_comune | 8100 | Assisi |

| Rank | Infobox Template | Times used | Wikipedia Article in the sample |
|---|---|---|---|
| 41 | Infobox_military_conflict | 8009 | Battle_of_Monte_Cassino |
| 42 | Infobox_software | 7096 | KOffice |
| 43 | Rugbybox | 7071 | Heineken_Cup_finals |
| 44 | Infobox_Australian_place | 6828 | Sydney |
| 45 | Infobox_Gridiron_football_person | 6318 | Chuck_Cecil |
| 46 | Infobox_ice_hockey_player | 6304 | Peter_Forsberg |
| 47 | Chembox | 6257 | Methanol |
| 48 | Infobox_Aircraft_Type | 6187 | A6M_Zero |
| 49 | Infobox_television_episode | 5439 | And_When_the_Sky_Was_Opened |
| 50 | Infobox_comics_character | 5155 | Superman |
| 51 | Infobox_hurricane_small | 5019 | 1973_Atlantic_hurricane_season |
| 52 | Infobox_station | 4862 | Union_Station_%28Toronto%29 |
| 53 | Infobox_stadium | 4750 | Veterans_Stadium |
| 54 | Infobox_NFLactive | 4543 | Johnny_Unitas |
| 55 | Infobox_disease | 4501 | Autism |
| 56 | Drugbox | 4347 | Amphetamine |
| 57 | Infobox_royalty | 4080 | Stephen_I_of_Hungary |
| 58 | Infobox_Automobile | 3855 | Volkswagen_K%C3%BCbelwagen |
| 59 | Infobox_artist | 3660 | Joan_Mir%C3%B3 |
| 60 | Infobox_weapon | 3656 | T-80 |
| 61 | Infobox_NFLretired | 3623 | Lawrence_Taylor |
| 62 | Infobox_protected_area | 3587 | Masai_Mara |
| 63 | Infobox_college_coach | 3520 | Joe_Paterno |
| 64 | Infobox_character | 3466 | Skuld_%28Oh_My_Goddess%21%29 |
| 65 | Infobox_animanga/Header | 3390 | Astro_Boy |
| 66 | Infobox_Korean_name | 3363 | Yi_Sun-sin* |
| 67 | Infobox_newspaper | 3314 | The_Philadelphia_Inquirer |
| 68 | Infobox_cricketer_biography | 3179 | Danish_Kaneria |
| 69 | Infobox_enzyme | 3166 | Carbonic_anhydrase* |
| 70 | Infobox_language | 3139 | Chinese_language |
| 71 | Infobox_animanga/Print | 3114 | Cowboy_Bebop |
| 72 | Infobox_UK_school | 3107 | Christ%27s_Hospital |
| 73 | Infobox_U.S._county | 3035 | Adams_County%2C_Idaho |
| 74 | Infobox_rugby_league_biography | 3025 | Arthur_Beetson |
| 75 | Infobox_song | 2982 | Stairway_to_Heaven |
| 76 | Infobox_organization | 2846 | Radio_Free_Europe/Radio_Liberty |
| 77 | Infobox_Swiss_town | 2769 | Winterthur |

* The infoboxes of these Wikipedia articles do not contain any relevant properties, thus they are skipped. The numbers given in “times used” as at March 2010.

Table 6 contains the sample's distribution of triples and Snippets to the Pattern Categories. A reading example would be: 801 triples arise from 203 List-Snippets. These are 24.9 % of all triples in the sample. For 480 of the 801 List-triples mappings are defined. This means that 59.9 % of the List-triples arise from mapped properties.

Table 6: Pattern Category Distribution of the Sample

| Pattern Category | Triples | | | | Snippets | | | |
|---|---|---|---|---|---|---|---|---|
| | num. | in % | map. | in % | num. | in % | map. | in % |
| Plain Property | 893 | 27.8 % | 515 | 57.7 % | 893 | 60.9 % | 515 | 57.7 % |
| List | 801 | 24.9 % | 478 | 59.7 % | 203 | 13.8 % | 125 | 61.6 % |
| Multi-Property Table | 619 | 19.3 % | 90 | 14.5 % | 91 | 6.2 % | 8 | 8.8 % |
| One-Property Table | 447 | 13.9 % | 242 | 54.1 % | 49 | 3.3 % | 24 | 49.0 % |
| Open Property | 139 | 4.3 % | 14 | 10.1 % | 55 | 3.7 % | 7 | 12.7 % |
| Internal Template | 116 | 3.6 % | 58 | 50.0 % | 50 | 3.4 % | 29 | 58.0 % |
| Number-Unit | 76 | 2.4 % | 51 | 67.1 % | 76 | 5.2 % | 51 | 67.1 % |
| Coordinate | 54 | 1.7 % | 36 | 66.7 % | 18 | 1.2 % | 12 | 66.7 % |
| Interval | 31 | 1.0 % | 22 | 71.0 % | 16 | 1.1 % | 11 | 68.8 % |
| Open Property Table | 26 | 0.8 % | 0 | 0.0 % | 2 | 0.1 % | 0 | 0.0 % |
| Merged Properties | 13 | 0.4 % | 7 | 53.8 % | 13 | 0.9 % | 7 | 53.8 % |
| Sum | 3215 | 100 % | 1517 | 47.2 % | 1466 | 100 % | 789 | 53.8 % |

Table 7 contains the distribution of the gold triples by their node and data type. A reading example would be: 2339 triples have an entity as subject. 1306 of these triples, or 55.8 %, arise from properties which have a mapping defined. 1005 of the 2339 entity-triples have a literal as object and 1334 entity-triples have a resource as object.

Table 7: Distribution of Gold Triples by Node and Data Type

| Subject | Object | Datatype | Triples | with Mapping | in % |
|---|---|---|---|---|---|
| Entities | | | 2339 | 1301 | 55.6 % |
| | Literals | | 1005 | 501 | 49.9 % |
| | | string | 514 | 223 | 43.4 % |
| | | double | 164 | 97 | 59.1 % |
| | | integer | 100 | 43 | 43.0 % |
| | | gYear | 93 | 55 | 59.1 % |
| | | date | 82 | 52 | 63.4 % |
| | | float | 37 | 25 | 67.6 % |
| | | gYearMonth | 13 | 5 | 38.5 % |
| | | gMonthDay | 1 | 1 | 100.0 % |
| | | time | 1 | 0 | 0.0 % |
| | Resources | | 1334 | 800 | 60.0 % |
| | | entity | 1033 | 691 | 66.9 % |
| | | intermediate | 253 | 77 | 30.4 % |
| | | URL | 48 | 32 | 66.7 % |
| Intermediates | | | 876 | 211 | 27.2 % |
| | Literals | | 495 | 112 | 22.6 % |
| | | gYear | 154 | 72 | 46.8 % |
| | | integer | 145 | 16 | 11.0 % |
| | | string | 103 | 12 | 11.7 % |
| | | double | 48 | 4 | 8.3 % |
| | | date | 29 | 4 | 13.8 % |
| | | float | 8 | 4 | 50.0 % |
| | | gYearMonth | 8 | 0 | 0.0 % |
| | | gMonthDay | 0 | 0 | - |
| | | time | 0 | 0 | - |
| | Resources | | 381 | 99 | 26.0 % |
| | | entity | 376 | 98 | 26.1 % |
| | | intermediate | 4 | 0 | 0.0 % |
| | | URL | 1 | 1 | 100.0 % |
| Sum | | | 3215 | 1512 | 47.2 % |

## A.3. Completeness & Precision Statistics

To avoid confusion by understanding the percentages, it is to say that the percentage in the present column contains the completeness indicator. It is the ratio of present Snippets to mappings, and the best case would be 100 %. The percentage in the missing column contains the ratio of missing Snippets with a mapping defined, and the best case would be 0 %, because then there are no missing triples with mappings defined.

Table 8: Snippet-based Completeness Statistics

| Pattern Category | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|
| | Snippets | Snippets | in % | Snippets | with Mapping | in % |
| Plain Property | 893 | 415 | 80.6 % | 478 | 100 | 20.9 % |
| List | 203 | 24 | 19.2 % | 179 | 101 | 56.4 % |
| Multi-Property Table | 91 | 0 | 0 % | 91 | 8 | 8.8 % |
| Number-Unit | 76 | 35 | 68.6 % | 41 | 16 | 39.0 % |
| Open Property | 55 | 3 | 42.9 % | 52 | 4 | 7.7 % |
| Internal Template | 50 | 0 | 0 % | 50 | 29 | 58.0 % |
| One-Property Table | 49 | 0 | 0 % | 49 | 24 | 49.0 % |
| Coordinate | 18 | 12 | 100 % | 6 | 0 | 0 % |
| Interval | 16 | 7 | 63.6 % | 9 | 4 | 44.4 % |
| Merged Properties | 13 | 4 | 57.1 % | 9 | 3 | 33.3 % |
| Open Property Table | 2 | 0 | - | 2 | 0 | 0 % |
| Sum | 1466 | 500 | 63.3 % | 966 | 289 | 29.9 % |

Table 9: Overall Triple-based Completeness Statistics

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 2339 | 688 | 52.9 % | 1651 | 613 | 37.1 % |
| | Literals | | 1005 | 346 | 69.1 % | 659 | 155 | 23.5 % |
| | | string | 514 | 154 | 69.1 % | 360 | 69 | 19.2 % |
| | | integer | 100 | 37 | 86.0 % | 63 | 6 | 9.5 % |
| | | double | 164 | 62 | 63.9 % | 102 | 35 | 34.3 % |
| | | date | 82 | 45 | 86.5 % | 37 | 7 | 18.9 % |
| | | gYear | 93 | 23 | 41.8 % | 70 | 32 | 45.7 % |
| | | gYearMonth | 13 | 0 | 0 % | 13 | 5 | 38.5 % |
| | | float | 37 | 25 | 100 % | 12 | 0 | 0 % |
| | | gMonthDay | 1 | 0 | 0 % | 1 | 1 | 100 % |
| | | time | 1 | 0 | - | 1 | 0 | 0 % |
| | Resources | | 1334 | 342 | 42.8 % | 992 | 458 | 46.2 % |
| | | entity | 1033 | 328 | 47.5 % | 705 | 363 | 51.5 % |
| | | intermediates | 253 | 0 | 0 % | 253 | 77 | 30.4 % |
| | | URL | 48 | 14 | 43.8 % | 34 | 18 | 52.9 % |
| Intermediate Nodes | | | 876 | 9 | 4.3 % | 867 | 202 | 23.3 % |
| | Literals | | 495 | 9 | 8.0 % | 486 | 103 | 21.2 % |
| | | string | 103 | 4 | 33.3 % | 99 | 8 | 8.1 % |
| | | integer | 145 | 0 | 0 % | 1445 | 16 | 11.0 % |
| | | double | 48 | 1 | 25.0 % | 47 | 3 | 6.4 % |
| | | date | 29 | 0 | 0 % | 29 | 4 | 13.8 % |
| | | gYear | 154 | 0 | 0 % | 154 | 72 | 46.8 % |
| | | gYearMonth | 8 | 0 | - | 8 | 0 | 0 % |
| | | float | 8 | 4 | 100 % | 4 | 0 | 0 % |
| | | gMonthDay | 0 | 0 | - | 0 | 0 | - |
| | | time | 0 | 0 | - | 0 | 0 | - |
| | Resources | | 381 | 0 | 0 | 381 | 99 | 26.0 % |
| | | entity | 376 | 0 | 0 | 376 | 98 | 26.1 % |
| | | intermediates | 4 | 0 | - | 4 | 0 | 0 % |
| | | URL | 1 | 0 | 0 | 1 | 1 | 100 % |
| Sum | | | 3215 | 697 | 46.1 % | 2518 | 815 | 32.4 % |

Table 10: Completeness Statistics of the Pattern Category Plain Property

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 893 | 415 | 80.5 % | 479 | 100 | 20.9 % |
| | Literals | | 523 | 226 | 81.3 % | 297 | 52 | 17.5 % |
| | | string | 256 | 117 | 85.4 % | 139 | 20 | 14.4 % |
| | | integer | 86 | 37 | 90.2 % | 49 | 4 | 8.2 % |
| | | double | 66 | 24 | 64.9 % | 42 | 13 | 31.0 % |
| | | date | 62 | 39 | 90.7 % | 23 | 4 | 17.4 % |
| | | gYear | 41 | 8 | 47.1 % | 33 | 9 | 27.3 % |
| | | gYearMonth | 9 | 0 | 0 % | 9 | 1 | 11.1 % |
| | | float | 1 | 1 | 100 % | 0 | 0 | - |
| | | gMonthDay | 1 | 0 | 0 % | 1 | 1 | 100 % |
| | | time | 1 | 0 | 0 % | 1 | 0 | 0 % |
| | Resources | | 370 | 188 | 79.7 % | 182 | 48 | 26.4 % |
| | | entity | 348 | 175 | 79.8 % | 174 | 44 | 25.3 % |
| | | URL | 22 | 14 | 77.8 % | 8 | 4 | 50.0 % |
| Sum | | | 893 | 415 | 80.5 % | 479 | 100 | 20.9 % |

Table 11: Completeness Statistics of the Pattern Category List

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 801 | 162 | 33.9 % | 639 | 316 | 49.5 % |
| | Literals | | 214 | 21 | 37.5 % | 193 | 35 | 18.1 % |
| | | string | 210 | 20 | 37.0 % | 190 | 34 | 17.9 % |
| | | double | 2 | 1 | 50.0 % | 1 | 1 | 100 % |
| | | date | 2 | 0 | - | 2 | 0 | 0 % |
| | Resources | | 587 | 141 | 33.4 % | 446 | 281 | 63.0 % |
| | | entity | 587 | 141 | 33.4 % | 446 | 281 | 63.0 % |
| Sum | | | 801 | 162 | 33.9 % | 639 | 316 | 49.5 % |

Table 12: Completeness Statistics of the Pattern Category One-Property Table

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 174 | 12 | 10.6 % | 162 | 101 | 62.3 % |
| | Literals | | 35 | 3 | 9.4 % | 32 | 29 | 90.6 % |
| | | gYear | 18 | 0 | 0 % | 18 | 16 | 88.9 % |
| | | string | 9 | 0 | 0 % | 9 | 9 | 100 % |
| | | date | 4 | 3 | 100 % | 1 | 0 | 0 % |
| | | double | 2 | 0 | 0 % | 2 | 2 | 100 % |
| | | gYearMonth | 2 | 0 | 0 % | 2 | 2 | 100 % |
| | Resources | | 139 | 9 | 11.1 % | 130 | 72 | 55.4 % |
| | | intermediate | 110 | 0 | 0 % | 110 | 57 | 51.8 % |
| | | entity | 25 | 9 | 42.9 % | 16 | 12 | 75.0 % |
| | | URL | 4 | 0 | 0 % | 4 | 3 | 75.0 % |
| Intermediate Nodes | | | 273 | 1 | 0.8 % | 272 | 128 | 47.1 % |
| | Literals | | 91 | 1 | 1.6 % | 90 | 61 | 67.8 % |
| | | gYear | 55 | 0 | 0 % | 55 | 51 | 92.7 % |
| | | string | 15 | 0 | 0 % | 15 | 5 | 33.3 % |
| | | gYearMonth | 7 | 0 | - | 7 | 0 | 0 % |
| | | date | 6 | 0 | - | 6 | 0 | 0 % |
| | | double | 4 | 1 | 25.0 % | 3 | 3 | 100 % |
| | | integer | 4 | 0 | 0 % | 4 | 2 | 50.0 % |
| | Resources | | 182 | 0 | 0 % | 182 | 67 | 36.8 % |
| | | entity | 182 | 0 | 0 % | 182 | 67 | 36.8 % |
| Sum | | | 447 | 13 | 5.4 % | 434 | 229 | 52.8 % |

Table 13: Completeness Statistics of the Pattern Category Merged Properties

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 13 | 4 | 57 % | 9 | 3 | 33 % |
| | Literals | | 13 | 4 | 57 % | 9 | 3 | 33 % |
| | | date | 9 | 2 | 50 % | 7 | 2 | 29 % |
| | | double | 4 | 2 | 67 % | 2 | 1 | 50 % |
| Sum | | | 13 | 4 | 57 % | 9 | 3 | 33 % |

Table 14: Completeness Statistics of the Pattern Category Multi-Property Table

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 135 | 0 | 0 % | 135 | 24 | 17.8 % |
| | Literals | | 21 | 0 | 0 % | 21 | 6 | 28.6 % |
| | | gYear | 12 | 0 | 0 % | 12 | 6 | 50.0 % |
| | | double | 6 | 0 | - | 6 | 0 | 0 % |
| | | integer | 2 | 0 | - | 2 | 0 | 0 % |
| | | date | 1 | 0 | - | 1 | 0 | 0 % |
| | Resources | | 114 | 0 | 0 % | 114 | 18 | 15.8 % |
| | | intermediate | 97 | 0 | 0 % | 97 | 12 | 12.4 % |
| | | entity | 17 | 0 | 0 % | 17 | 6 | 35.3 % |
| Intermediate Nodes | | | 484 | 8 | 12.1 % | 476 | 58 | 12.2 % |
| | Literals | | 339 | 8 | 17.8 % | 331 | 37 | 11.2 % |
| | | integer | 128 | 0 | 0 % | 128 | 14 | 10.9 % |
| | | gYear | 96 | 0 | 0 % | 96 | 21 | 21.9 % |
| | | string | 49 | 4 | 67 % | 45 | 2 | 4.4 % |
| | | double | 38 | 0 | - | 38 | 0 | 0 % |
| | | date | 19 | 0 | - | 19 | 0 | 0 % |
| | | float | 8 | 4 | 100 % | 4 | 0 | 0 % |
| | | gYearMonth | 1 | 0 | - | 1 | 0 | 0 % |
| | Resources | | 145 | 0 | 0 % | 145 | 21 | 14.5 % |
| | | entity | 145 | 0 | 0 % | 145 | 21 | 14.5 % |
| Sum | | | 619 | 8 | 8.9 % | 611 | 82 | 13.4 % |

Table 15: Completeness Statistics of the Pattern Category Coordinates

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 54 | 36 | 100 % | 18 | 0 | 0 % |
| | Literals | | 54 | 36 | 100 % | 18 | 0 | 0 % |
| | | float | 36 | 24 | 100 % | 12 | 0 | 0 % |
| | | string | 18 | 12 | 100 % | 6 | 0 | 0 % |
| Sum | | | 54 | 36 | 100 % | 18 | 0 | 0 % |

Table 16: Completeness Statistics of the Pattern Category Number-Units

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 76 | 35 | 68.6 % | 41 | 16 | 39.0 % |
| | Literals | | 76 | 35 | 68.6 % | 41 | 16 | 39.0 % |
| | | double | 76 | 35 | 68.6 % | 41 | 16 | 39.0 % |
| Sum | | | 76 | 35 | 68.6 % | 41 | 16 | 39.0 % |

Table 17: Completeness Statistics of the Pattern Category Intervals

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 31 | 16 | 72.7 % | 15 | 6 | 40.0 % |
| | Literals | | 31 | 16 | 72.7 % | 15 | 6 | 40.0 % |
| | | gYear | 19 | 15 | 93.8 % | 4 | 1 | 25.0 % |
| | | date | 4 | 1 | 50 % | 3 | 1 | 33.3 % |
| | | integer | 4 | 0 | 0 % | 4 | 2 | 50.0 % |
| | | gYearMonth | 2 | 0 | 0 % | 2 | 2 | 100 % |
| | | double | 2 | 0 | - | 2 | 0 | - |
| Sum | | | 31 | 16 | 72.7 % | 15 | 6 | 40.0 % |

Table 18: Completeness Statistics of the Pattern Category Open Property

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 71 | 4 | 33.3 % | 67 | 8 | 11.9 % |
| | Literals | | 20 | 0 | 0 % | 20 | 2 | 10.0 % |
| | | integer | 8 | 0 | - | 8 | 0 | 0 % |
| | | string | 7 | 0 | 0 % | 7 | 1 | 14.3 % |
| | | gYear | 3 | 0 | - | 3 | 0 | 0 % |
| | | double | 2 | 0 | 0 % | 2 | 1 | 50.0 % |
| | Resources | | 51 | 4 | 40.0 % | 47 | 6 | 12.8 % |
| | | intermediate | 29 | 0 | 0 % | 29 | 1 | 3.4 % |
| | | entity | 19 | 4 | 44.4 % | 15 | 5 | 33.3 % |
| | | URL | 3 | 0 | - | 3 | 0 | 0 % |
| Intermediate Nodes | | | 68 | 0 | 0 % | 68 | 2 | 2.9 % |
| | Literals | | 44 | 0 | 0 % | 44 | 1 | 2.3 % |
| | | string | 32 | 0 | 0 % | 32 | 1 | 3.1 % |
| | | integer | 10 | 0 | - | 10 | 0 | 0 % |
| | | double | 2 | 0 | 0 % | 2 | 0 | 0 % |
| | Resources | | 24 | 0 | 0 % | 24 | 1 | 4.2 % |
| | | entity | 23 | 0 | - | 23 | 0 | 0 % |
| | | URL | 1 | 0 | 0 % | 1 | 1 | 100 % |
| Sum | | | 139 | 4 | 28.6 % | 133 | 10 | 7.4 % |

Table 19: Completeness Statistics of the Pattern Category Open Property Table

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 8 | 0 | - | 8 | 0 | 0 % |
| | Literals | | 3 | 0 | - | 3 | 0 | 0 % |
| | | double | 3 | 0 | - | 3 | 0 | 0 % |
| | Resources | | 5 | 0 | - | 5 | 0 | 0 % |
| | | entity | 3 | 0 | - | 3 | 0 | 0 % |
| | | intermediate | 2 | 0 | - | 2 | 0 | 0 % |
| Intermediate Nodes | | | 18 | 0 | - | 18 | 0 | 0 % |
| | Literals | | 6 | 0 | - | 6 | 0 | 0 % |
| | | double | 4 | 0 | - | 4 | 0 | 0 % |
| | | string | 2 | 0 | - | 2 | 0 | 0 % |
| | Resources | | 12 | 0 | - | 12 | 0 | 0 % |
| | | entity | 8 | 0 | - | 8 | 0 | 0 % |
| | | URL | 4 | 0 | - | 4 | 0 | 0 % |
| Sum | | | 26 | 0 | - | 26 | 0 | 0 % |

Table 20: Completeness Statistics of the Pattern Category Internal Templates

| Subject | Object | Data Type | Sum | Present | | Missing | | |
|---|---|---|---|---|---|---|---|---|
| | | | Triples | Triples | in % | Triples | with Mapping | in % |
| Entities | | | 83 | 5 | 11.4 % | 78 | 39 | 50.0 % |
| | Literals | | 15 | 5 | 45.5 % | 10 | 6 | 60.0 % |
| | | string | 14 | 5 | 50.0 % | 9 | 5 | 55.6 % |
| | | double | 1 | 0 | 0 % | 1 | 1 | 100 % |
| | Resources | | 68 | 0 | 0 % | 68 | 33 | 48.5 % |
| | | entity | 34 | 0 | 0 % | 34 | 15 | 44.1 % |
| | | URL | 19 | 0 | 0 % | 19 | 11 | 57.9 % |
| | | intermediate | 15 | 0 | 0 % | 15 | 7 | 46.7 % |
| Intermediate Nodes | | | 33 | 0 | 0 % | 33 | 14 | 42.4 % |
| | Literals | | 15 | 0 | 0 % | 15 | 4 | 26.7 % |
| | | string | 5 | 0 | - | 5 | 0 | 0 % |
| | | date | 4 | 0 | 0 % | 4 | 4 | 100 % |
| | | gYear | 3 | 0 | - | 3 | 0 | 0 % |
| | | integer | 3 | 0 | - | 3 | 0 | 0 % |
| | Resources | | 18 | 0 | 0 % | 18 | 10 | 55.6 % |
| | | entity | 18 | 0 | 0 % | 18 | 0 | 55.6 % |
| Sum | | | 116 | 5 | 8.6 % | 111 | 53 | 47.7 % |

Table 21 contains the precision values of the first run and the second run. The first run is the evaluation of the extraction framework of the DBpedia 3.5.1 release and the second run is evaluation of the extraction framework version of the DBpedia 3.6 release.

Table 21: Comparison of Precision Statistics between the first and second run

| Pattern Category | first run | second run |
|---|---|---|
| Coordinate | 100 % | 100 % |
| Interval | 100 % | 100 % |
| Open Property | 100 % | 100 % |
| List | 91.5 % | 92.9 % |
| Plain Property | 91.1 % | 96.5 % |
| Multi-Property Table | 88.9 % | 88.9 % |
| Number-Unit | 85.4 % | 85.4 % |
| Internal Template | 83.3 % | 75 % |
| Merged Properties | 80.0 % | 80.0 % |
| One-Property Table* | 32.5 % | 38.5 % |
| Open Property Table | - | - |
| Overall | 91.1 % | 92.3 % |

* The extraction framework cannot handle One-Property Tables yet. This is already pointed in a triple-based completeness of only 5.4 %. The precision suffers from wrong extracted intermediate nodes.

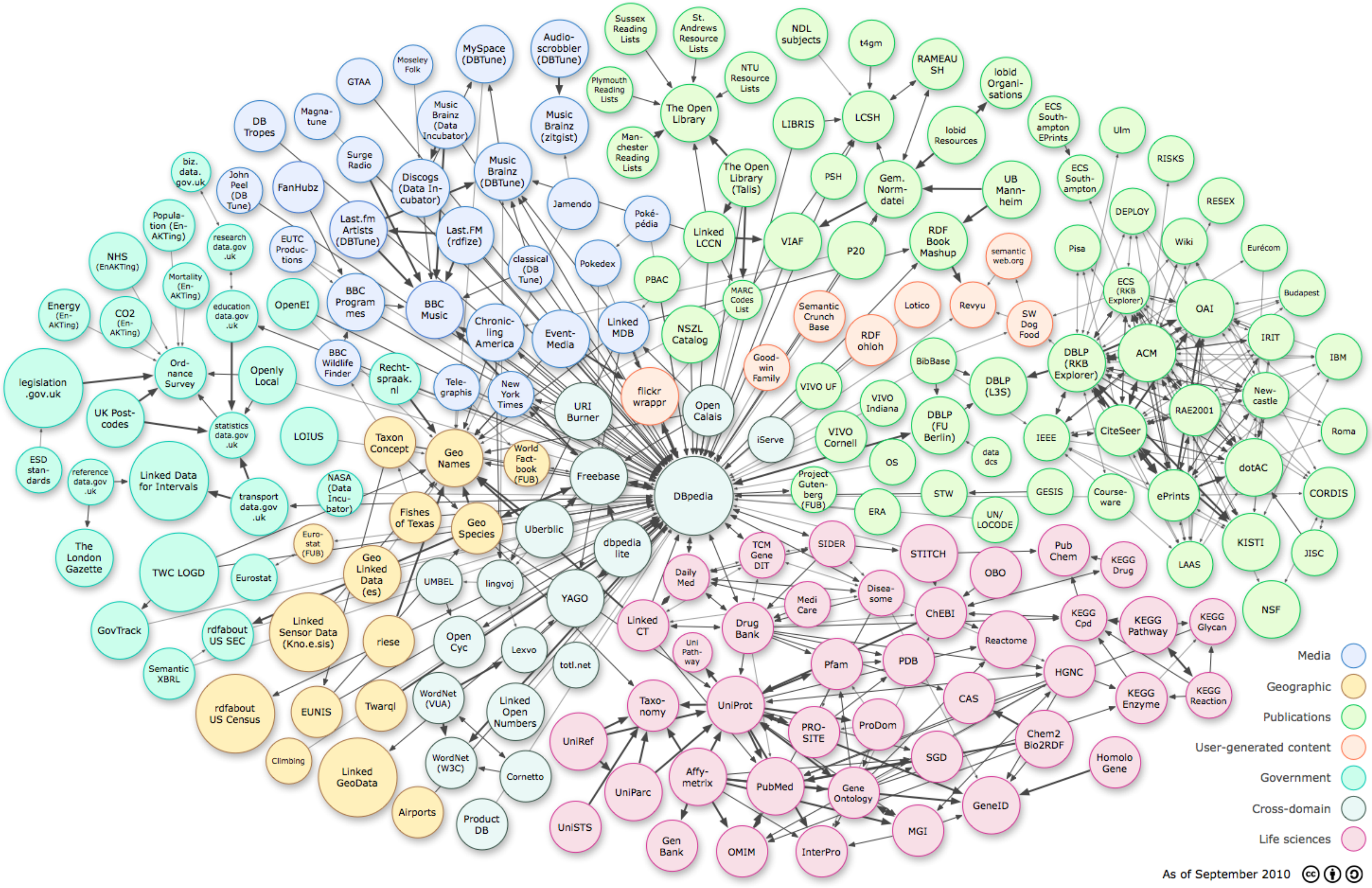
Table 22 contains the completeness values of the first run and the second run. The first run is the evaluation of the extraction framework of the DBpedia 3.5.1 release and the second run is evaluation of the extraction framework version of the DBpedia 3.6 release.

Table 22: Comparison of Completeness Statistics between the first and second run

| Pattern Category | first run | second run |
|---|---|---|
| Coordinate | 100 % | 100 % |
| Interval | 72.7 % | 68.2 % |
| Open Property | 28.6 % | 33.3 % |
| List | 33.9 % | 75.9 % |
| Plain Property | 80.5 % | 83.5 % |
| Multi-Property Table | 8.9 % | 8.9 % |
| Number-Unit | 68.6 % | 67.3 % |
| Internal Template | 8.6 % | 10.2 % |
| Merged Properties | 57.1 % | 57.1 % |
| One-Property Table | 5.4 % | 6.1 % |
| Open Property Table | - | - |
| Overall | 46.1 % | 60.6 |

## A.4. Figures

Figure 10: LOD Cloud



Sussex Reading Lists
St. Andrews Resource Lists
NDL subjects
t4gm
RAMEAU SH
lobid Organisations
Moseley Folk
MySpace (DBTune)
Audioscrobbler (DBTune)
Plymouth Reading Lists
NTU Resource Lists
GTAA
Magnatune
DB Tropes
Music Brainz (Data Incubator)
Music Brainz (zitgist)
The Open Library
LIBRIS
LCSH
ECS Southampton EPrints
lobid Resources
Ulm
Surge Radio
Manchester Reading Lists
biz. data. gov.uk
John Peel (DB Tune)
FanHubz
Discogs (Data Incubator)
Music Brainz (DBTune)
The Open Library (Talis)
PSH
Gem. Normdatei
UB Mannheim
ECS Southampton
RISKS
Jamendo
Pokédia
RESEX
Population (EnAKTing)
Last.fm Artists (DBTune)
Last.FM (rdfize)
EUTC Productions
research data.gov .uk
Linked LCCN
VIAF
P20
RDF Book Mashup
DEPLOY
NHS (EnAKTing)
classical (DB Tune)
Pokedex
semantic web.org
Pisa
Wiki
Eurécom
Mortality (EnAKTing)
BBC Programmes
BBC Music
OpenEI
PBAC
Energy (EnAKTing)
CO2 (EnAKTing)
education data.gov .uk
Chronicling America
Event Media
Linked MDB
MARC Codes List
Semantic Crunch Base
Lotico
Revyu
ECS (RKB Explorer)
Budapest
OAI
NSZL Catalog
SW Dog Food
RDF ohloh
BBC Wildlife Finder
Rechtspraak. nl
Openly Local
Ordnance Survey
legislation .gov.uk
Goodwin Family
BibBase
DBLP (RKB Explorer)
ACM
IRIT
IBM
Telegraphis
New York Times
URI Burner
flickr wrappr
Open Calais
VIVO UF
VIVO Indiana
DBLP (L3S)
Newcastle
UK Postcodes
statistics data.gov .uk
LOIUS
Taxon Concept
Geo Names
World Factbook (FUB)
Freebase
iServe
VIVO Cornell
DBLP (FU Berlin)
IEEE
CiteSeer
RAE2001
Roma
ESD standards
reference data.gov .uk
Linked Data for Intervals
NASA (Data Incubator)
transport data.gov .uk
DBpedia
Project Gutenberg (FUB)
OS
data dcs
dotAC
Fishes of Texas
Geo Species
Uberblic
STW
GESIS
Courseware
ePrints
CORDIS
The London Gazette
Eurostat (FUB)
ERA
UN/ LOCODE
TWC LOGD
Eurostat
Geo Linked Data (es)
UMBEL
lingvoj
dbpedia lite
TCM Gene DIT
SIDER
STITCH
Pub Chem
KEGG Drug
KISTI
JISC
LAAS
Daily Med
Diseasome
OBO
YAGO
GovTrack
rdfabout US SEC
Linked Sensor Data (Kno.e.sis)
riese
Open Cyc
Lexvo
Linked CT
Medi Care
Drug Bank
ChEBI
KEGG Cpd
KEGG Pathway
KEGG Glycan
NSF
Semantic XBRL
totl.net
Uni Pathway
Reactome
Pfam
PDB
HGNC
KEGG Enzyme
KEGG Reaction
rdfabout US Census
EUNIS
Twarql
WordNet (VUA)
Linked Open Numbers
Taxonomy
UniProt
CAS
UniRef
PROSITE
ProDom
Chem2 Bio2RDF
Homolo Gene
SGD
Climbing
Linked GeoData
WordNet (W3C)
Cornetto
Affymetrix
UniParc
PubMed
Gene Ontology
GeneID
Airports
Product DB
UniSTS
Gen Bank
OMIM
InterPro
MGI
Media
Geographic
Publications
User-generated content
Government
Cross-domain
Life sciences
As of September 2010