WIRTSCHAFTS
UNIVERSITÄT
WIEN VIENNA
UNIVERSITY OF
ECONOMICS
AND BUSINESS

EFMD
EQUIS
ACCREDITED

Bachelor Thesis

Data Finder: Open Data search over multiple sources

Georg Prohaska

Date of Birth: 14.11.1983
Student ID: 0325904

Subject Area: Information Business

Studienkennzahl: 033 561

Supervisors: Dr. Jürgen Umbrich, Dr. Javier D. Fernández

Date of Submission: 27.July 2016

Department of Information Systems and Operations, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

DEPARTMENT FÜR INFORMATIONS-
VERARBEITUNG UND PROZESS-
MANAGEMENT DEPARTMENT
OF INFORMATION SYSTEMS AND
OPERATIONS

Contents

List of Figures

List of Tables

Listings

Abstract

The amount of available Open Data is constantly increasing. Members of the OpenData@WU project are collecting metadata of datasets of over 260 Open Data portals. This thesis addresses the problem of providing a search interface over this metadata. It investigates the technique of faceted search for providing exploratory browsing to the user. In the course of this work a prototype was developed that provides such an interface. Details on how it was created are the main part of this paper.

Zusammenfassung

Jeden Tag steigt die Menge an verfügbaren offenen Daten (Open Data). Die Mitglieder des OpenData@WU Projektes sammeln Metadaten über Datensätze von über 260 Open Data Portalen. Diese Arbeit beschäftigt sich damit, wie ein Suchinterface für diese Metadaten aussehen soll. Wir untersuchen die Methode der Facettensuchen, um dem User eine Suche zur Verfügung zu stellen, bei der er den Suchraum möglichst gut erkunden kann. Im Zuge dieser Arbeit wurde ein Prototyp entwickelt der zeigt, wie solch ein Interface aussehen könnte. Die Details über Aufbau und Ausführung dieses Prototypen machen den Hauptteil dieser Arbeit aus.

Acknowledgements

I want to use this space to thank my supervisors Dr. Jürgen Umbrich and Dr. Javier David Fernandez Garcia. They always had time for me, answered all my questions and gave me very helpful feedback and suggestions.

I also would like to thank my family for their moral support, especially my wife Gabriele who is always there for me and helped me keep my motivation up throughout this study.

1 Introduction

In recent years the concept of Open Data has become more and more relevant. Every day an increasing amount of data is made available for the public to use. There are various providers of Open Data: A lot of different governments and international institutions, such as the EU, have been publishing their data about numerous subjects including geography, transport, environment, economics and many more. There are also private organizations and other institutions, such as universities, that contribute further to the amount of Open Data that is available for everyone to use. Usually each of these parties manages its own portal to provide access to their individual datasets. Often there are even multiple portals within a country managing different regions or themes of Open Government Data. In Austria, for instance, there are sox websites that provide information mostly related to a distinct region within the country [21]. It can be concluded that there is a lot of information available to be discovered and that the possibilities for the end user are growing rapidly.
It should be clear that the high number of access points for Open Data represents a major barrier for a lot of users. First of all, one has to find the portal that provides access to the desired information. That is not an easy task, especially when the users goals are somewhat unspecific. There are data portal catalogs available on the web, for example Dataportals[1], however they only provide lists of names and a short description which is often not enough to infer the true nature of the underlying datasets. Websites, such as Opendatasoft[2], provide a means of finding data portals based on geographical regions, yet they do not facilitate a search over multiple portals.
Another fact that makes searching or browsing the different sources of Open Data difficult is usability. Although there are software packages, such as CKAN[3] or SOCRATA[4], that are widely used among developers of data portals, interfaces of websites differ quite a bit in complexity, usability and design. Therefore, the end user has to invest quite some time to get familiar with every new portal he encounters, which can be frustrating regarding the big number of different sites that exist at the moment.
Designing a search interface for Open Data that provides good functionality, while remaining relatively easy to use is a particularly challenging task. Research shows that end users often take a browsing or exploratory approach rather than looking up specific information [20]. However, if one is seeking

[1] http://dataportals.org/
[2] http://www.opendatasoft.com/
[3] http://ckan.org/
[4] http://socrata.com/

something explicit then the interface still has to provide some means to reach that goal. In general, the functionality necessary to facilitate proper navigation through the available information grows with the number of datasets a portal manages. Hence, different techniques for filtering and visualizing information are required. Many Open Data portals attempt to improved their search interface. There are many different strategies currently in use. Some try a more visual approach, like Offenes Jena[5], others, like APILeibzig[6] only provide APIs which clearly targets a more skilled user group. It should be clear that there are alot of ways to tackle the problem of searching of Open Data, however not final solution has emerged yet.

1.1 Research Question and Approach

The goal of this thesis is to create an Open Data portal that is able to provide a search over multiple other websites. These are restricted to the six Austrian web pages that were mentioned in the introduction because of the narrow time frame. However, the aim is to achieve enough scalability to render further extension of the project possible. The search engine is based on a database which was provided by the Institute for Information Business of the Vienna University of Economics and Business. This database includes RDF triples [3] in the form of the Data Catalog Vocabulary [6] which is a format recommended by the World Wide Web Consortium (W3C). Our approach makes use of this metadata and considers the use of faceted search as a means to allow the user to narrow down results by selecting different filters, such as region, file format, publisher, subject, et cetera. In addition, we support keyword search, which is a standard feature among similar websites. The output consists of meta data for each dataset and a reference to the matching portal page.
The research question for this thesis can be posed as follows:

"How should an interface that provides a faceted search over multiple Open Data sources be designed?"

This thesis was completed in the following four phases:

1. Study of the state of the art
2. Design of the solution
3. Implementation

[5] `http://www.offenes-jena.de/`
[6] `http://www.apileipzig.de/`

4. Evaluation of the solution

At the beginning, a thorough study of the state of the art was necessary in order to get familiar with the topic of Open Data and faceted search. This included comparing many different Open Data portal interfaces, researching filtering and visualization techniques and of course getting familiar with the nature of the relevant datasets. In the second phase the solution had to be designed. This meant deciding on filtering and visualization techniques. Here, the decision to adopt the technique of faceted search was cemented. Furthermore, the appearance of the user interface and the facets themselves were selected at that stage. The third phase consisted of implementation. Here, it was initially necessary to get familiar with the relevant APIs. The next step was to design the architecture for the prototype, and, finally, to write the actual code. The fourth and last phase of this thesis consisted of evaluation of the achieved solution. This included testing, in the course of which the current limitations were discovered. The final step was to formulate the written paper at hand.

2 Theoretical Foundations

At first we are going to discuss the theoretical foundations that are necessary to tackle our problem. We are going to explain what Open Data is exactly and what kinds of Open Data are distinguished. Furthermore, we are going to take a look at the RDF data model [3], which is closely related to Open Data, and, finally, we are going briefly discuss part about faceted search.

2.1 Open Data

Open Data refers to *"A piece of content or data is open if anyone is free to use, reuse, and redistribute it – subject only, at most, to the requirement to attribute and share-alike."* [2] The motivation behind this idea is that Open Data will facilitate innovation, creativity and new businesses, which will consequently lead to economic growth. The amount of data that is available is growing at an ever increasing rate. The Open Data Barometer of 2015 [10] in Figure 1 shows that especially countries of the western world have high capacity for Open Data, which means that they all have established policies for this subject.

However, the availability of data alone does not contribute much to economic growth. Creative individuals or businesses that take advantage of the possibilities of Open Data are needed to achieve that goal. This is a process

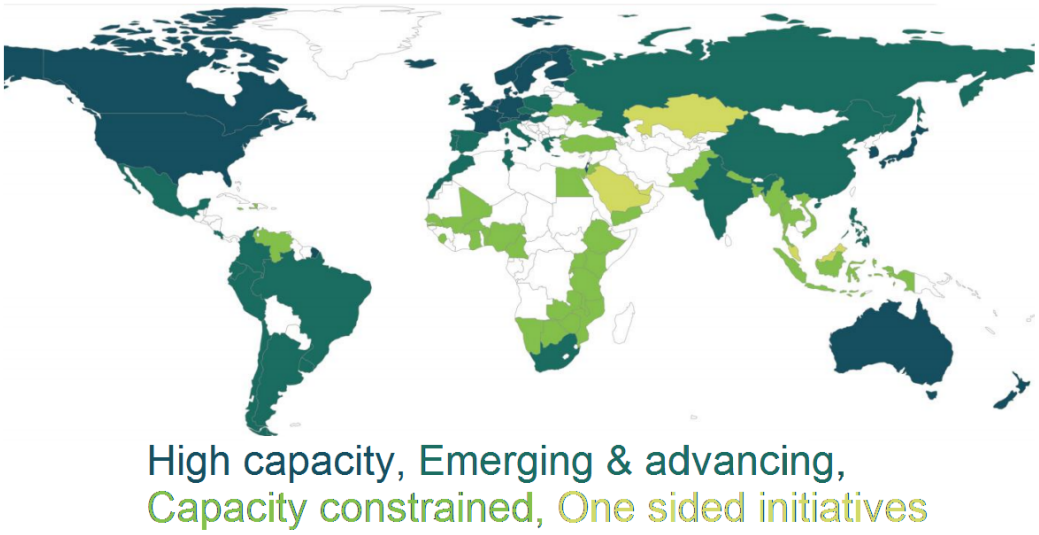

Figure 1: Country clusters based on Open Data Barometer Readiness and Impact questions. Reprinted from [10]

.

that takes some time. Nevertheless, some believe that the number of individuals and businesses that are already engaged in Open Data has reached the critical mass necessary to trigger a set-change in business attitudes towards this matter. Goods and services are being enhanced by the use of Open Data. Furthermore, more and more new businesses, like Duedil, i3 Education Services, and more emerge that base their entire business model around finding new ways to make use of the available data [11]. Big players like Google are also starting to open up their own data in order to increase customer satisfaction and to facilitate easier supplier management. In the future Open Data might have a significant impact on our society. Yet, this process is still in its infancy and alot of work remains to be done in order to reach the full potential of the available possibilities.

2.1.1 Open Data Portals

Open Data is usually distributed over public websites which we are calling Open Data portals. Thousands of distinct portals are currently available from all over the world. There are many different approaches on how to present the data to the user. A lot of portals provide interfaces that only allow searching over the metadata of the datasets, while the actual data is stored in downloadable files. There are numerous examples for this approach, one would be the Open Data Portal of the Vienna University of Economics and

Business[7]. Other websites, such as Worldatlas[8], provide a more integrated type of interface that also utilizes some visualization techniques. A different method of granting access to Open Data is to provide just a SPARQL [19] endpoint. This enables experienced users to formulate complex queries in order to answer a big variety of questions. A prominent example for this is DBpedia[9]. This is by no means a complete list of the different kinds of interfaces that are currently in use. It should be clear now that there are numerous ways of solving the problem of search over Open Data and each solution has its individual pros and cons.

2.2 Open Government Data

Open Govenment Data (OGD) is that part of the worldwide Open Data that is provided by different governments. The goal is to increase transparency of government and administrative structures and to encourage citizens to take part in the democratic process. This notion became increasingly popular around the world since the G8 leaders signed an Open Data Charter in 2013 where they agreed to *"Establish an expectation that all government data be published openly by default, ..."* [1]. This lead many countries to launch Open Data intitiatives publishing many different kinds of data. However, some critizise that many countries have yet to open up significant core data about government spending, company registers or public sector contracts, which would contribute alot to transparency and consequently help reduce corruption [10]. While, in many regions around the world, there is still a lot of improvement possible in this matter, the trend seems to go into the right direction. OGD is published mostly in formatted files. TSV, XML, JSON, CSV are just some of the many different open formats that are currently used to store the data. Evidently, this represents a considerable barrier for the end user. A standard format has not yet been established. Furthermore, inconsistencies or poor description of the datasets is very common in OGD[13], which also renders searching the available data hard for the end user. These issues are most likely not going to be solved by the providers of the data in the near future, since, from a technical standpoint, government institutions tend to improve rather slow. However, since OGD is in fact open, third parties have the opportunity to reach improvements or rather to create completely new solutions using OGD.

[7] `http://data.wu.ac.at/`
[8] `http://www.worldatlas.com/`
[9] `http://wiki.dbpedia.org/`

2.3 Linked Open Data

Linked Open Data (LOD) is another form of Open Data. It refers to data that is organized in such a way that its meaning can be interpreted by a computer and that it can be linked to other external data sets [22]. This concept was introduced by Tim Berners Lee in 2006 in a Web architecture note [4]. The Resource Description Framework (RDF) [3], which is a data model for describing relationships between resources, has become the de facto standard for creating LOD. In recent years the lines between the Semantic Web, which is a term that has been utilized frequently in the past, and LOD have become blurred [22]. The ultimate goal is to create the so called Web of Data that can be read and interpreted by a machine. This is also commonly called "The Linked Open Data Cloud"[10] which has been growing rapidly in recent years. Figure 2 shows a graph of the current state of the LOD cloud. Each circle represents a dataset and the radius depends on the number of triples that source provides. DBPedia for example has about 3.000.000.000 triples. The arrows indicate the existence of at least 50 links between two datasets [7]. It should be clear now that there is already a massive amount of information out there and that the potential for innovative applications is particularly high. However, because of the complexity and in-homogeneity of the LOD cloud even a basic task such as searching poses major difficulties for software developers. Hence, different research topics have emerged aside, such as semantic search on LOD, integrating large number of linked data sources, mining the web of linked data, quality evaluation of linked data, et cetera. As a result, we are definitely going to see more and more diverse and interesting applications being developed in the near future.

2.4 RDF

The Resource Description Framework (RDF) [3] is a model to express logical statements about resources. A resource is something that is unique and that one wants to make a statement about. It can be anything from a webpage to a physical or abstract entity. Resources are identified by so called Uniform Resource Identifiers (URI) which are unique strings of characters. Since a lot of resources are in fact websites URIs are often denoted in the form of Uniform Resource Locators (URL). However, URIs do not have to necessarily be reachable on the web. In the latest recommendation of the RDF-model the International Resource Identifier (IRI) has been introduced, which is a generalization of the URI. It allows non-ASCII characters to be used in the

[10] `http://linkeddata.org/`

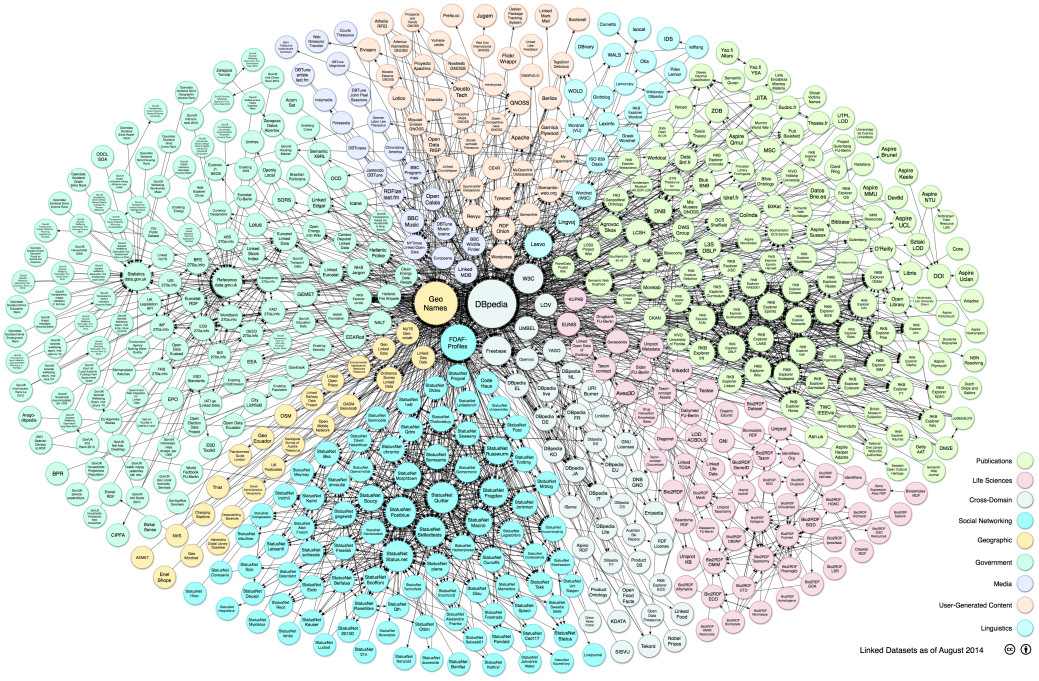

Figure 2: The Linked Open Data cloud

character string.
In the RDF-model each statement consists of three parts: subject, predicate and object. The subject – a resource – is described using the predicate and the object, which can be resources, just values expressed by literals or even blank nodes. Blank nodes represent something that is not specified concretely. They are used like simple variables in algebra [3]. The predicate characterizes the relationship between the subject and the object. These three units are called a RDF-triple. The RDF-model can be mathematically understood as a labeled direct graph, where the directed arcs point from subjects to objects via predicate labels with meaning. The following code shows a simple example for three RDF triples:

Listing 1: An RDF example in RDF/Turtle notation

```
:dataset1
        a dcat:Dataset;
        dct:title "Imaginary dataset";
        dcat:keyword "water" .
```

This defines "dataset1" as a dataset under the DCAT Vocabulary [6]. The second triple states that the title of this specific dataset is "Imaginary dataset". The quotations imply that the object is a literal. The last triple indicates

that the keyword for this dataset is "water", again a literal.
RDF was originally developed by the World Wide Web Consortium as a standard to describe metadata. However, it has been widely employed for building the Semantic Web i.e. the LOD. There are various common serialization formats for RDF, namely N-Triples, N-Quads, JSON-LD, N3, RDF/XML and Turtle (see main RDF formats in [3]). They each have different advantages, for example Turtle is often used by developers of the Semantic Web since its syntax is very easily readable for a human.
By now, a lot of different RDF-vocabularies have been developed that define various predicates and objects in order to provide a base line for formulating statements. For instance, the very commonly used FOAF vocabulary[11] specifies characteristics of people and social groups, such as name, age or title.

2.4.1 SPARQL

To query RDF-data the SPARQL language [19] has become the most commonly used standard. Its syntax and structure resembles SQL. SPARQL has been officially recommended by the World Wide Web Consortium in 2008. It is based on graph pattern matching. A SPARQL query normally consists of one or more sets of triple patterns. These patterns stand for the RDF subject, predicate and object. The components of each triple may be replaced with a variable instead of an IRI or literal. These patterns may also be joined in order to create more complex queries. Consequently, these sets are matched against the RDF data. The results of SPARQL queries can be result sets or RDF graphs. SPARQL also supports aggregation, subqueries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph [19].

2.4.2 DCAT

For this project we needed a way to describe the metadata of all the datasets coming from different Open Data portals. For this purpose the Data Catalog Vocabulary (DCAT) [6] was a good choice, since it covers everything from title over keywords to a description of the catalog i.e. the portal. Figure 3 shows the structure of the DCAT vocabulary. It essentially consists of three main classes:

1. dcat:Catalog
2. dcat:Dataset

[11] `http://www.foaf-project.org/`

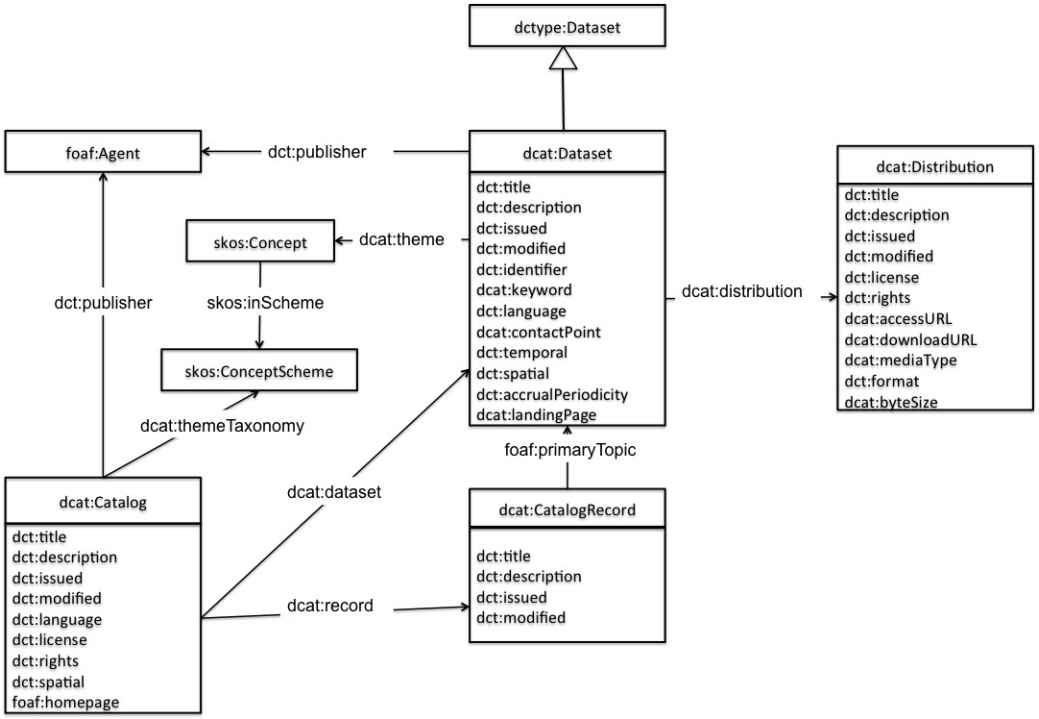

Figure 3: Structure of the DCAT vocabulary [7]

3. dcat:Distribution

A Catalog contains one or many datasets, in our case it translates to one of the Open Data portals. In DCAT a Dataset is defined as a *"collection of data, published or curated by a single agent, and available for access or download in one or more formats"*[6], so it does not necessarily have to be a downloadable file. However, in our project all datasets have that property. Hence, each has a related distribution which represents the file itself. Another RDF-vocabulary that is employed for describing datasets is the VoID vocabulary[12]. It was also created for expressing metadata about datasets, however, it focuses mainly on RDF-datasets and most of the files managed in this project are not actual RDF-files.

2.5 Faceted search

Faceted search is a technique that narrows down search results by using so called facets. Facets are filters for different attributes of the content[15]. For example genre or release date could be two facets for a database of music

[12] `https://www.w3.org/TR/void/`

titles. Facets can have numerical values, such as release date or price range, which are entered by the user. Generally, all facets can be selected by the user and each one constrains matching sets in the search result. The dimension of the content that each facet describes should be orthogonal in order to provide efficient browsing through the data[8]. Faceted search facilitates an easy way of navigating through large content sets, since it provides help to the user for understanding the structure of the available data[14]. This is the main advantage of faceted search; it allows exploratory browsing, since the user can add and remove each facet separately and thereby navigate effectively through the content. Furthermore, if the facets are chosen well this technique is relatively intuitive and easy to use. As of now, faceted search is being used by many different online platforms such as Amazon or Ebay which have very large product ranges for their customers to explore.

3 Related work

We are now going to discuss related work done by different researchers on our topic. We are going to find out that various solutions for faceted search over RDF-data have been created already. A selection of them are going to be examined later in this chapter.

At this time, there is a lot of research being conducted on the topic of OGD and LOD. For example in 2015 a paper was published that investigates the process of data discovery of OGD by novice users [20]. In their study the authors created two different search interfaces: An exploratory data search interface and a baseline interface. The exploratory interface displays the results of a search via a visualization in the form of a colored graph that shows how the found datasets are interconnected. The baseline search interface on the other hand displays the results just as textual lists. Both interfaces work on the same search engine which manages numerical datasets of multiple portals for OGD across Canada. The authors provided four increasingly difficult data search tasks to the participants of the study and invited them to try to solve them using both interfaces. The 16 participants of the study were relatively young, well-educated and mostly familiar with online searching. To evaluate the results both observation and a questionnaire techniques were used. As expected the exploratory data search interface was rated higher on average by the participants in regard to usability. Yet, no significant increase in effectiveness or user satisfaction could be measured by the researchers. This was a relatively small study, however it indicates that colorful visualization techniques do not necessarily increase the effectiveness

of a search interface.

Upon researching OGD and LOD one comes across the topic of faceted search quite frequently. The reason for this is that searching Open Data is still a rather significant problem that has not yet been solved satisfyingly. Faceted search provides a good option to cover some of the exploratory features an Open Data search interface needs. That is why it has been used by multiple research groups to create different prototypes of search interfaces. We will now examine some of them.

3.1 Flamenco

One of the earlier implementations of a faceted search interface was presented by Yee et al. in 2003 [14]. The researchers tried to find an alternative to a keyword-based search interface which was still very common at that time. The search-space consisted of 35,000 fine arts images from the Fine Arts Museum of San Francisco. A limited amount of metadata like artist name, type of media and dates was available, however the only accessible content-based metadata was textual descriptions of the images. Yee et al. created an algorithm that semi-automatically mapped these descriptions to a number of metadata categories they found suitable. This was done by using WordNet [9] which is a large lexical database of English. All the words in the textual descriptions were compared to their higher-level category labels in WordNet and the most frequent categories of each description was stored. Some of the resulting categories had to be altered or discarded manually, yet the authors claim that their algorithm worked surprisingly well. These categories are subsequently used to create facets which allow exploratory browsing of the search-space. The design goals for Flamenco were to focus on usability and to provide an interface that is suited for both searching and browsing. Figure 4 shows a screenshot of the implementation. The available facets are grouped at the left side of the screen each with a number showing how many datasets will remain after applying it to the search. Active facets are displayed at the top and can be removed at any time by clicking on the corresponding 'X'.
Upon completion of the prototype the authors conducted a usability study. They compared the Flamenco interface to a baseline interface that provided only a keyword search. The 32 participants of the study were mostly art history students and people who took art courses. All of them were familiar with and frequent users of other common search interfaces. The evaluation was based on server logs, behavioral logs and paper surveys. The results of the study showed that most participants experienced greater search satisfaction and success with the faceted interface than with the baseline. Furthermore,

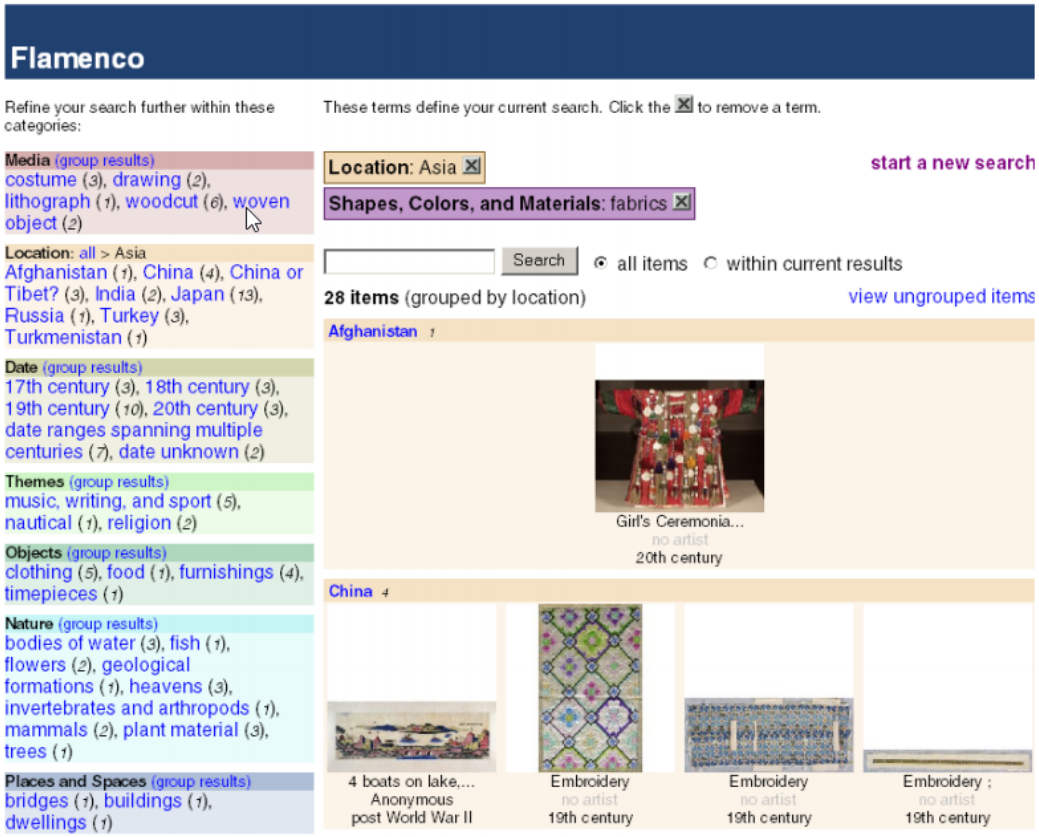

Figure 4: User interface of Flamenco

the faceted interface was perceived to be more useful and flexible than the baseline. 97% of the participants claimed that the faceted interface helped them to get familiar with the provided collection of images. The success rate for finding a desired data set was also significantly higher for the faceted interface.
Yee et al. created a very well designed interface for faceted search and additionally provided a usability study that clearly indicates the advantages of such an approach.

3.2 Faceted Wikipedia Search

In 2010 Hahn et al. presented a prototype that provided a faceted search interface for the English edition of Wikipedia [16]. The goal was to enable the user to answer complex questions, like "Which rivers flow into the Rhine and are longer than 50 kilometers?". Wikipedia is quite a big search-space and the number of different facets is very high so the researchers had to solve

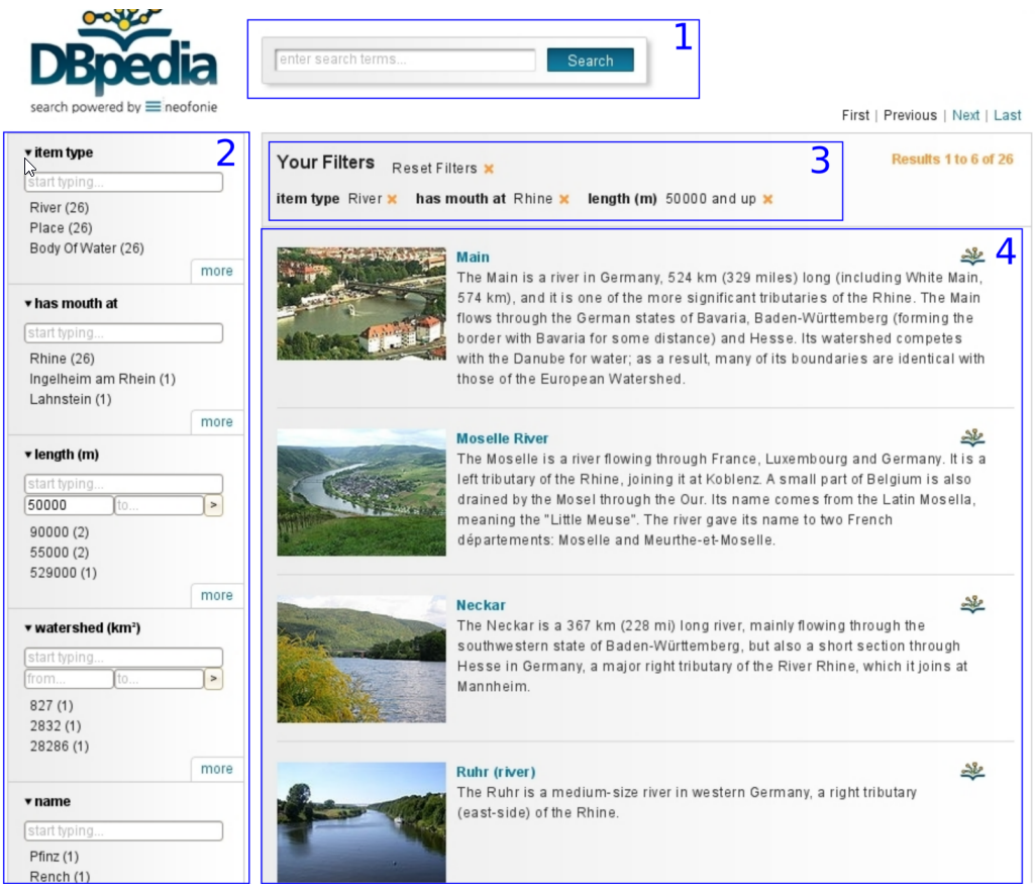

Figure 5: User interface of Faceted Wikipedia Search

the problem of selecting facets that are presumably interesting to the user. Figure 5 shows a screenshot of the interface of the prototype. The user can either start a query by entering a search term into the field at the top or by selecting a facet on the left side of the screen. At the start of a session the most generic facets, such as “item type”, “location” and “year-of-appearance” are displayed. The available facet values can be searched using the corresponding text field.

Faceted Wikipedia Search relies on the DBPedia knowledge base[13]as a search-space. DBPedia is a community project that extracts structured information from Wikipedia and makes it available to the public. Most of the extracted data comes from the infoboxes of Wikipedia which display an article’s most relevant facts. The DBPedia knowledge base is organized as a RDF triple store which contains over 3 billion triples describing over 4,5 million things

[13]`http://wiki.dbpedia.org/`

in 125 different languages. For Faceted Wikipedia Search only the English triples were used.
In Faceted Wikipedia Search, each document is ordered to an item type, which the facets are then assigned to. When an "item type" is selected or a search term is entered the corresponding result set is shown at the center of the interface. Additionally, the most frequent facets of the result set are displayed. The facet values are presented to the user as a list. This list is ordered by the number of documents that correspond to the respected value. The researchers used a commercial search engine as a base for their prototype, so they did not have to focus too much on the issue of response time for large search-spaces. Faceted Wikipedia Search is another nice example how faceted search can be used to navigate user friendly through large amounts of data by providing structured information about the content.

3.3 tFacet

tFacet is an example for a somewhat different approach of implementing a faceted search. It was presented by Brunk and Heim in 2011 [5]. They tried to provide a solution that acts as an intermediary between the user and any SPARQL endpoint that provides access to an RDF store. Their motivation was to enable inexperienced SPARQL users to formulate complex queries to the LOD. tFacet uses a directory tree representation for the facet hierarchy. The authors believe that this is an interface most users are familiar with since it is used in almost all common operating systems. Figure 6 shows how the interface of tFacet looks like. In this case the chosen SPARQL endpoint belonged to the DBPedia data store. To start a user has to select one base class from the available datasets, for instance "film" or "Eurovision song contest". Resulting sets are displayed at the top of the screen (A). On the bottom left side (B) the user can select any number of corresponding facets from a directory tree. At the bottom right (C) one can select the desired values for the selected facets which are then consequently applied and displayed in the result set. Furthermore, the user can chose which facet values are to be displayed in the results tab (D).
Facets are automatically created from the underlying RDF data. For each predicate a facet is constructed. If the predicate leads to a literal then a leaf in the directory tree is reached, this is indicated by a document symbol. If a predicate leads to another object a hierarchy is created – indicated by a folder symbol in the directory tree.
tFacet offers an interesting approach to faceted search. Brunk and Heim tried to make the Semantic Web more accessible for inexperienced users by reusing well-known interaction concepts such as the directory tree representation.

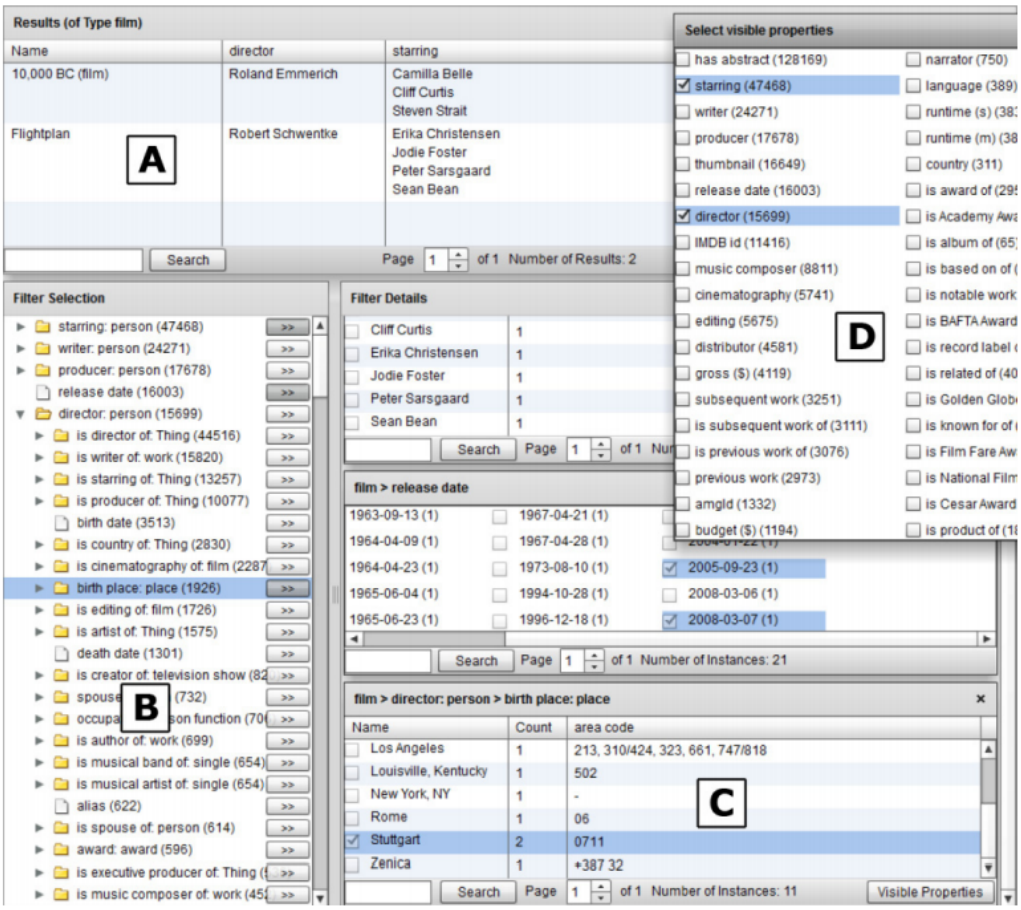

Figure 6: User interface of tFacet

4 Data Finder: Goals and Methods

In this chapter we will present the goals and methods we used in order to create a prototype for search across multiple Open Data portals. The current working title of the application is *Data Finder*. It is a web application that allows the user to perform a search across six different Austrian Open Data portals. The technique of faceted search is utilized to provide an easy way of navigating through the available data.

4.1 Goals

The goal for *Data Finder* was to create an interface that facilitates search across multiple Open Data portals. There are currently six Austrian portals:

1. *Offene Daten Österreichs* - A universal portal for Austria commissioned

Table 1: Statistics of Austrian Open Data portals

URL	Datasets	Files	Publishers
http://data.gv.at/	2026	8358	36
http://www.opendataportal.at/	357	926	31
http://data.salzburgerland.com/	6	-	1
http://data.ktn.gv.at/	102	163	1
http://data.graz.gv.at/	153	409	1
http://data.wu.ac.at/	115	117	2

by the "Cooperation OGD Österreich"[14].

2. *Opendataportal* - Another universal portal for Austria created in a cooperation between "Wikimedia Österreich"[15], "Knowledge Foundation Österreich"[16] and "Cooperation OGD Österreich".

3. *Salzburgerland Data Hub* - Portal for the state of Salzburg. It provides access to its data via a SPARQL endpoint. Since DCAT data for this type of interface was not provided this portal was left out of the prototype.

4. *OGD Kärnten* - Portal for the state of Kärnten.

5. *OGD Graz* - Portal for the city of Graz.

6. *OpenData@WU* - Portal for the Vienna University of Economics and Business.

Table 1 shows a list of the six Austrian portals that are online at this time together with the number of datasets, the number of distinct downloadable files and the number of publishers each portal manages. The Salzburgerland Data Hub provides all of its data in the form of RDF triples, which can be accessed via a SPARQL endpoint. Therefore, there are no directly downloadable files available at that website.

The prototype was made with the intention of being able to provide scalability for a bigger number of portals. This goal was only partly reached due to current speed limitations of the data transfer from the server to the client, for more details the reader is invited to review the chapters “Limitations” and “Future work”.

[14] http://data.gv.at/hintergrund-infos/cooperation-ogd-oesterreich/
[15] https://www.wikimedia.at/
[16] http://www.okfn.at/

Another goal was to create a faceted search interface in order to provide exploratory browsing to the user. In combination with a text-based keyword search this should enable the user to have a satisfying search experience.

4.2 Data

The data for *Data Finder* was originally mined from CKAN metadata of the different Open Data portals. This data includes links to the datasets, title, description, date of issue, date of last modification, keywords, name of publisher, license, formats, and links to downloadable files. Consequently, this data was converted to RDF using the DCAT vocabulary. This process was conducted by the OpenData@WU project and the resulting triples were contributed for this project. The provided files were in the JSON-LD format which is an RDF-format as mentioned previously, however, they had to be converted to the N3 format for implementation reasons. The following code shows an example dataset in the N3 format.

Listing 2: An example dataset in the N3 format

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://example.org/dataset1> a dcat:Dataset;
    dcterms:description "A list of all friends of Georg";
    dcterms:identifier "133337";
    dcterms:issued "2015-01-26T13:23:57";
    dcterms:modified "2015-02-09T09:56:37";
    dcterms:publisher <http://example.org/MyFriends>;
    dcterms:title "Friends of Georg";
    dcat:contactPoint [ a vcard:Organization;
              vcard:fn "Georg and Friends";
              vcard:hasEmail "Georg@example.org" ];
    dcat:distribution <http://example.org/files >;
    dcat:keyword "Georg", "Friends" .

<http://example.org/files> a dcat:Distribution;
    dcterms:description "A list of Names";
    dcterms:format "CSV";
```

```
23      dcterms:issued "2015-01-26T14:23:59";
24      dcterms:license <http://www.opendefinition.org/
25                                          licenses/cc-by>;
26      dcterms:title "Friends";
27      dcat:accessURL <http://example.org/files/
28                                          friends.csv> .
29
30  <http://example.org/MyFriends> a foaf:Organization;
31      foaf:name "Friends of Georg Prohaska" .
32
33  <http://www.opendefinition.org/licenses/cc-by>
34      rdfs:label "Creative Commons Attribution";
35      dcterms:identifier "cc-by" .
```

The first 5 lines introduce shortcuts in order to make the code easier to read. At line 7 the dataset is defined as a DCAT dataset. Thereafter, description, identifier, date of issue and date of last modification are speficied in the form of literals. At line 12 the publisher of the dataset is set to an example URI. The next few rows define a contact point for the dataset, which is an organization with a name and an email address. Thereafter, the distribution is set to an example URI and two keywords are defined. From line 20 to line 28 this distribution is secified. A distribution stands for one downloadable file in one format. It can have a description, a format, a date of issue, a license, a title and an access URL, which is the direct link to the file. Lines 30 and 31 specify the publisher that was used before. It is an organization and has a name. The last three lines specify the CC-BY license that was used in the distribution. It has a label and an identifier.

4.3 Faceted classification

This part will explain how and why we chose the available facets for *Data Finder*. In 1998 Louise Spiteri drew up a set of principles for the creating faceted classifications [18]:

1. Differentiation: "when dividing an entity into its component parts, it is important to use characteristics of division (i.e., facets) that will distinguish clearly among these component parts"

2. Relevance: "when choosing facets by which to divide entities, it is important to make sure that the facets reflect the purpose, subject, and scope of the classification system"

3. Ascertainability: "it is important to choose facets that are definite and can be ascertained"
4. Permanence: facets should "represent permanent qualities of the item being divided"
5. Homogeneity: "facets must be homogeneous"
6. Mutual Exclusivity: facets must be "mutually exclusive," "each facet must represent only one characteristic of division"
7. Fundamental Categories: "there exist no categories that are fundamental to all subjects, and ... categories should be derived based upon the nature of the subject being classified"

The DCAT vocabulary offers some fields that correspond to most of these principles. We felt that the following predicates served these guidelines the most:

- "dct:publisher": This field contains the name of the organization that published the corresponding dataset. We think that this facet surely meets the requirements since it should be relevant for any user to know where the data he wants to use comes from.
- "dct:issued", "dct:modified": These two fields contain the date at which the corresponding dataset was issued respectively modified. The relevance of these facets is inherent; they represent a very important aspect of the data.
- "dct:format": This field contains the information which file formats of the corresponding dataset are available. This facets might not meet the relevance requirement for every user, however all the other principles are clearly served.

There is one more predicate that is commonly used as a facet by a lot of Open Data portals, namely the "dct:license" predicate. However, we decided against the implementation of the license as a facet, since for the current number of managed datasets this field has the same value over 99% of the time. Apparently, virtually all datasets of Austrian portals fall under the Creative Commons BY license. Therefore, a facet for this predicate would have very limited use. However, for future work the license of a dataset should definitely be considered as a facet, since there are a variety of licenses currently in use around the world and it represents an important aspect of the data.

In addition to this direct conversion of DCAT predicates to facets we introduced two more types of facets:

- "Region": This is a facet that classifies a dataset as belonging to a geographical region namely a city or a state. For this prototype we did not take into account which country a dataset could be associated to since all of our datasets would be classified as Austrian, however this would be an aspect for future work. The "Region" facet is assigned by matching the "dct:title", "dct:description" and "dcat:keyword" fields of all datasets to a list of cities and states. This list was drawn up manually since the number of potential regions for the datasets at hand is not very big. Using this technique 57% of the datasets could be assigned to a region.

- "Theme": This is actually a group of facets that characterize datasets based on their context. We decided to chose 9 themes that correspond very well to the categorization that was used on most of the data portals *Data Finder* currently manages. Our categories are: Finance, People, Environment, Education, Health, Economy, Art and Culture, Politics and Law, Geography. This selection of themes was done arbitrarily, however for future work the use of algorithms like Castanet [17] which automatically generate faceted metadata from textual descriptions would definitely be a good option to consider. However, since this was out of scope for this thesis, we assigned each dataset to one or more of these facets by using a manual mapping of keywords to the preselected themes.

A list of all regions and keyword - themes matchings is provided in the appendix of this thesis.

4.4 Adjusting Facets

Orent et al. proposed three metrics to measure the "navigation quality" of a facet [8]. The first one is balance, which refers to the balance of the navigation tree that a facet represents. That means that a facet is balanced when the number of corresponding results is approximately equal for all facet values. Orent et al. posted the following formula to compute the balance of a facet:

$n_s(o_i)$ is the distribution of datasets over the facet values, μ is the mean and N_s is the total number of datasets.
The second metric is the cardinality of a facet which is nothing more than the number of different facet values that are available. Preferably this number

$$balance(p) = 1 - \frac{\sum_{i=1}^{n} \mid n_s(o_i) - \mu \mid}{(n-1)\mu + (N_s - \mu)}$$

Figure 7: Formula for the balance of a facet

should not be too high in order to avoid overwhelming the user with too many choices. Oren et al. use the formula in Figure 8 to compute the metric. The optimal number of facet values can be regulated through the μ

$$card(p) = \begin{cases} 0 & \text{if } n_o(p) \leq 1 \\ \exp^{-\frac{(n_o(p)-\mu)^2}{2\sigma^2}} & \text{otherwise} \end{cases}$$

Figure 8: Formula for the cardinality of a facet

and σ parameters.
The last metric is the frequency of a facet. The more datasets a facet covers the more efficient it is in dividing the search-space. A facet that occurs only in a handful of datasets is not going to be very useful for the user. This is reflected in the frequency metric. It is posted as follows:

$$freq(p) = \frac{n_s(p)}{n_s}$$

Figure 9: Formula for the frequency of a facet

Here $n_s(p)$ is the number of datasets that the facet covers and n_s is the number of total datasets. All three of these metrics are normalized to [0..1] so they can be easily combined into a score for navigational quality.

In order to reach good scores we had to introduce groups of values for the publisher and the format facet. These groups were created arbitrarily with regard to usability and navigational quality. For the publisher facet we decided to divide all possible values into three categories:

1. Regional institutions: These are institutions that are responsible for a specific region, for instance the city of Vienna.

2. State institutions: These are institutions that are state-run, for instance the Department of Finance.

3. Private institutions: These are private institutions, for instance Greenpeace.

The second facet that needed grouping of values was the format facet. We introduced the following five groups:

1. Geo-information: These are formats that contain any kind of geographical or spacial information like coordinates et cetera. A common example would be the Shapefile (SHP) format.

2. Picture: These formats are used to store graphical information, in most cases pictures; Portable Network Graphics (PNG) or JPEG would be examples for this group.

3. Structured: This group stands for formats that contain structured information. Examples would be JSON, HTML, or XML

4. Text: Here all formats are grouped that contain written text. PDF, TXT and DOC are some of the most common formats of this type.

5. Other: This is the group for all remaining formats. Some of them are compression type formats like ZIP, others are more exotic and rare.

These were the groupings for the present version of *Data Finder*. They were tailored to suit the current datasets of the six Austrian portals. Therefore, they are probably going to need some adjustment when the managed data increases. A complete list of the groupings can be found in section A of the appendix of this thesis.
For the “issued” and “modified” facets we decided to allow only the year as a facet value, since this provided the highest navigational quality scores for the current data. However, in future work this will be subject to change, since with a bigger number of datasets smaller time spans would make a lot of sense. Furthermore, if the underlying database is updated regularly options like “last week” or “last month” should be considered.
With the grouping of facet values we reached satisfying navigational quality scores for the facets. Table 2 shows the scores in detail.

Using the grouping of values we achieved relatively good scores for most of the facets. The only facet that stands out is "Theme". This facet currently has a rather low navigational quality score, however we still kept it in, since this facet has very good potential for future work.

Table 2: Navigational quality scores of facets

Facet	Balance	Cardinality	Frequency	Total score
Publisher	0,37	0,98	0,99	0,78
Issued	0,66	1	1	0,89
Modified	0,46	0,98	1	0,81
Region	0,58	0,72	0,57	0,63
Theme	0,43	0,8	0,5	0,17
Format	0,49	1	0,99	0,82

5 Data Finder: Implementation

Data Finder was developed as a web application that can be used by any browser that supports Javascript and HTML 5.0. Most of the testing so far was done using the Google Chrome browser, however a brief check showed that the application works on many common browsers without producing any major bugs.
In this chapter we are going to discuss the current architecture of *Data Finder*. Firstly, we are going to cover which tools and packages we utilized. Secondly, we are going to dive into the concrete architecture and code of *Data Finder*. Thirdly, the present state of the user interface is going to be shown. Lastly, the limitations of the current version are going to be explained.

5.1 Tools

Data Finder was created using PHP 5.5.9 for the server side implementation. This choice was based on personal preference and experience of the author. PHP provides all necessary requirements, the main one, which was being able to place SPARQL queries and retrieve the results, was achieved by the use of an external library called sparqllib[17]. We used the Openlink Virtuoso server[18] to store the data in the form of RDF-triples. This was again based on personal preference as well as recommendation of the thesis supervisors. In order to be able to properly import the data into the RDF-store a conversion from JSON-LD to N3 was necessary. The JSON-LD format allows the use of Internationalized Resource Identifiers (IRI) which is a form of URI that allows the use of almost all characters of the Universal Character Set. However, N3 only supports URIs which can only consist of a subset of ASCII characters. For the triples at hand the only characters that are not allowed in the URI

[17] `http://graphite.ecs.soton.ac.uk/sparqllib/`
[18] `http://virtuoso.openlinksw.com/`

definition were whitespaces. Therefore, the affected RDF-triples had to be converted. This was done through the use of a simple Python script that encoded all whitespaces to fit the definition.
The user interface as well as all of the program logic on the client side is entirely written in Javascript and CSS. For the user interface we rely on two external packages; The first one being Bootstrap[19], which is an open source framework that provides a uniform and pleasant look and feel for most interface elements, such as buttons, labels, tables et cetera. The second library in use is called Boostrap Tree View[20], which is, as the name indicates, build on top of the Bootstrap framework. This is just a small package that provides a solution for the graphical display and selection of the hierarchical structure of the facets.
Data Finder currently works with an Apache webserver which was installed on an Ubuntu workstation. All of the infrastructure was provided by the Institute for Information Business of the Vienna University of Economics and Business.

5.2 Architecture

Now, we are going to discuss the details of the implementation of the current version of *Data Finder*. Figure10 shows a simplified depiction of the architecture of the prototype. At the start of the project the CKAN metadata was collected from the Open Data portals and converted to the RDF vocabulary DCAT. Thereafter, the RDF triples were loaded into the RDF store. The core of the prototype consists of two main parts, namely the user interface and the PHP script. The user interface allows the user to select facets and enter any number of searchterms in order to start his search. This information is then passed to the server where the PHP script formulates a SPARQL query based on the selection of the user and sends that query to the RDF store. The resulting datasets are then encoded as a Javascript array and displayed by the user interface. The support files contain some needed information about facet groupings and naming; we are going to discuss their contents in detail later in this chapter.
Most of the code of *Data Finder* is divided between the two files "index.php" and "results.php". The first one provides most of the client side logic. Here, all of the user interface (except for the results list) is initialized and managed. This entails setting up the layout of the page, handling user input and managing breadcrumb creation. The second file is responsible for maintaining the

[19] `http://getbootstrap.com/`
[20] `https://github.com/jonmiles/bootstrap-treeview`

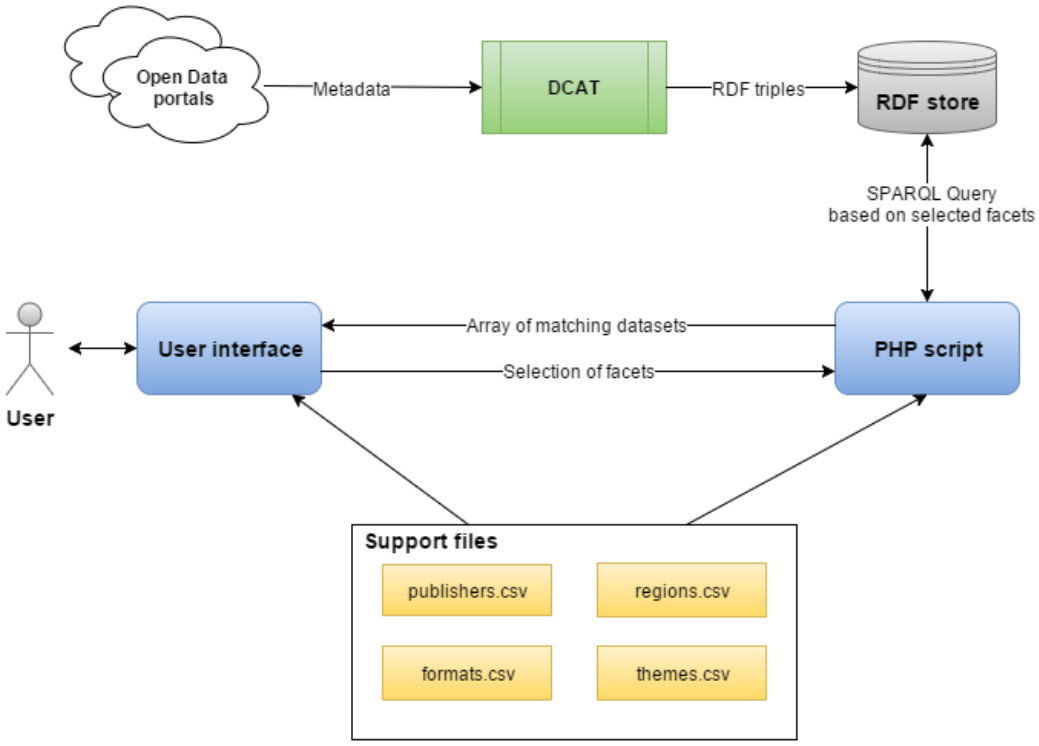

Figure 10: Architecture of *Data Finder*

connection to the RDF store, formulating SPARQL queries and processing and displaying the results. The "results.php" file is actually embedded into "index.php" by the use of an HTML iframe element.
When *Data Finder* is started initially all 2425 datasets are loaded and shown in the results list. As soon as the user selects a facet or enters a searchterm, via the textbox and corresponding button, a process starts that reduces the resulting set of data based on the input that was made. Figure 11 shows how this process works in detail. In the following, we provide details of each process.

1. User input and list of selected facets

When the user clicks on a facet or presses the search-button the first thing that happens is that the selection is added to a list, which is implemented as a list of strings separated by semicolons.

2. Create visual breadcrumb

Once the selection has been added to the list a visual breadcrumb is created in the upper part of the page just above the facet selection and results list.

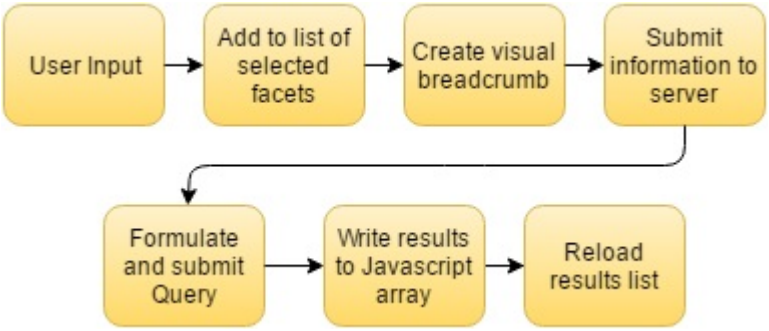

Figure 11: Implemented process of faceted search

Each breadcrumb is represented by a HTML div element that contains the facet type and name as a textual label and a small button that allows the user to remove the facet. The color of each breadcrumb depends on the type of facet that was selected, which is an idea that was inspired by the work of Martin A. Hearst[12]. Breadcrumbs are arranged from left to right based on the chronological order in which they were created. The intention behind this is to give the user an eye-catching visual history of the path he took through the informational space to reach the current results.

3. Submit information to server

After the breadcrumb has been generated the list of selected facets is sent to the server. This happens by submitting the HTML form mentioned above to the server targeting the "results.php" script, which handles communication with the RDF store and visualization of the results.

4. Formulate and submit query

Upon receiving the list of selected facets the PHP script first puts together a list of filters for the SPARQL query that corresponds to the chosen facet values. To fully understand how this works we have to take a look at the support files mentioned above. All of them are in the CSV format and were created manually. The contents of the files can be reviewed in the appendix of this thesis. There are four files:

1. "publishers.csv": This file includes a list of all publishers and which group each one belongs to. The first column contains the name of the publisher and the second column contains a code that represents the group. "ra" stands for regional institution, "si" for state institution and "pi" for private institution.

2. “regions.csv”: This file includes a list of all currently possible regions that a dataset can be assigned to. The first column contains the name of the region in German and the second column contains the name in English.

3. “formats.csv”: Here all the possible formats are listed together with a code for the group it belongs to. The first column contains the format in the same syntax as it is stored in the RDF data, which is the three to four letter file extension in most cases. The second column contains the name of the corresponding group.

4. “themes.csv”: This file contains the matching of keywords to the nine different themes for the datasets we introduced earlier. The first column contains the keywords and the second column contains the matching theme. The keywords are taken from the dcat:keyword field of the RDF data. This list is comprised of just the most frequent keywords that could be easily matched to one of the categories, so it is far from being complete.

When one of the facet groups, for instance, a format group is selected a list of filters is compiled that consists of all the corresponding facet values. All of the filters are combined using the OR operator in order to achieve matches to the full group of values. These are taken from the corresponding support file in this case “formats.csv”.
The following listing shows how a query of *Data Finder* looks like. This query is formulated when the user selects "Stadt Wien" as a publisher, 2016 as the year of issue, and JPEG as the desired format. These facets are achieved via the FILTER statements at line 38, 39 and 40.

Listing 3: An example SPARQL query from *Data Finder*

```
1  SELECT ?dataset, ?title, ?description, ?publisher,
2  concat(day(?issued), ".", month(?issued), ". ",
3    year(?issued)) as ?issued,
4  concat(day(?modified), ".", month(?modified), ". ",
5    year(?modified)) as ?modified, ?format
6  {
7    ?dataset a dcat:Dataset .
8    ?dataset dct:title ?title .
9    ?dataset dct:description ?description .
10   ?dataset dct:publisher ?p .
11   ?p foaf:name ?publisher .
12   ?dataset dct:issued ?issued .
```

```
13    ?dataset dct:modified ?modified .
14
15    {
16    SELECT ?dataset, concat(
17      group_concat(?k ; separator = " / "),
18       " ", ?t, " ", ?de) as ?all,
19      group_concat(?k; separator =" / ") as ?kws
20    WHERE {
21      ?dataset dcat:keyword ?k .
22      ?dataset dct:title ?t .
23      ?dataset dct:description ?de .
24    }GROUP BY ?dataset ?t ?de
25    }
26
27    {
28    SELECT ?dataset,
29      group_concat(distinct ?f ; separator =" / ")
30          as ?format
31    WHERE{
32      ?dataset dcat:distribution ?di .
33      ?di dct:format ?f
34    }
35    }
36
37    FILTER CONTAINS(lcase(?publisher), "stadt wien")
38    FILTER (year(?issued) = 2016)
39    FILTER CONTAINS(lcase(?format), "jpeg")
40  }
```

In general the query is a selection of all necessary RDF predicates for each available dataset. Since one dataset can have multiple distributions (files) and therefore multiple formats, a subquery that uses the *group_concat* function of SPARQL is necessary in order to concatenate all formats into one field. In this way a resulting table is achieved where each line represents one dataset. The complete list of all datasets is reduced by adding filter conditions to the SPARQL query. For instance each searchterm that was input by the user is matched against a concatenation of the three RDF predicates "dcat:keyword", "dct:title" and "dct:description". The SPARQL function *contains* is utilized to achieve this. In order to avoid any issues with capitalization all fields and restrictions are first fully converted to lower case characters. Therefore, the searchterms are case-insensitive. Furthermore, all

strings are converted to the UTF 8 encoding.
Filtering for the region facet works the same way; The selected regions are matched against the same concatenation as the searchterms. The only exception is that the strings to match are taken form the "regions.csv" file in order to achieve hits for both German and English occurrences of the name of the region.
When all necessary filters are generated the query is constructed and sent to the RDF store. To achieve this we used the external library that was mentioned before. This library only needs the query in the form of a string and an URL to a SPARQL endpoint and it returns the resulting table as an array.

5. Write results to Javascript array

The array that is returned by sparqllib is consequently used by the PHP script to generate an analogous Javascript array at the client's side. This means that the client has to load the complete list of resulting datasets into memory. Since all of the data is solely alphanumeric and therefore relatively small this is currently not an issue. However, with a bigger number of datasets sequential loading should be thought about.

6. Reload results list

Finally, the iframe that contains the results list is reloaded based on the new data. The resulting datasets are displayed using the unordered list HTML element, which is visually refined by the bootstrap framework. Pagination is implemented in Javascript in such a way that the unordered list is created dynamically whenever a new page is selected. Now, the user can review the results and continue refining his search if necessary.

If, instead of picking a new one, the user decides to remove an element from the list of breadcrumbs, the process currently works the same way, the only difference being that the facet is removed from the internal list of selected facets and the visual breadcrumb is discarded as well.
This was a summary of how *Data Finder* processes user input and manages faceted search. At this time, when an additional facet is selected a new query is sent to the RDF store and the complete results list is reloaded. Admittedly, this is not necessary, since adding a new restriction to the search can never increase the number of resulting sets, therefore, the client does not require any new information from the server. Nevertheless, this is just a first prototype so there is still much room for improvement, and this is definitely a point where efficiency can be increased relatively easily in the future.

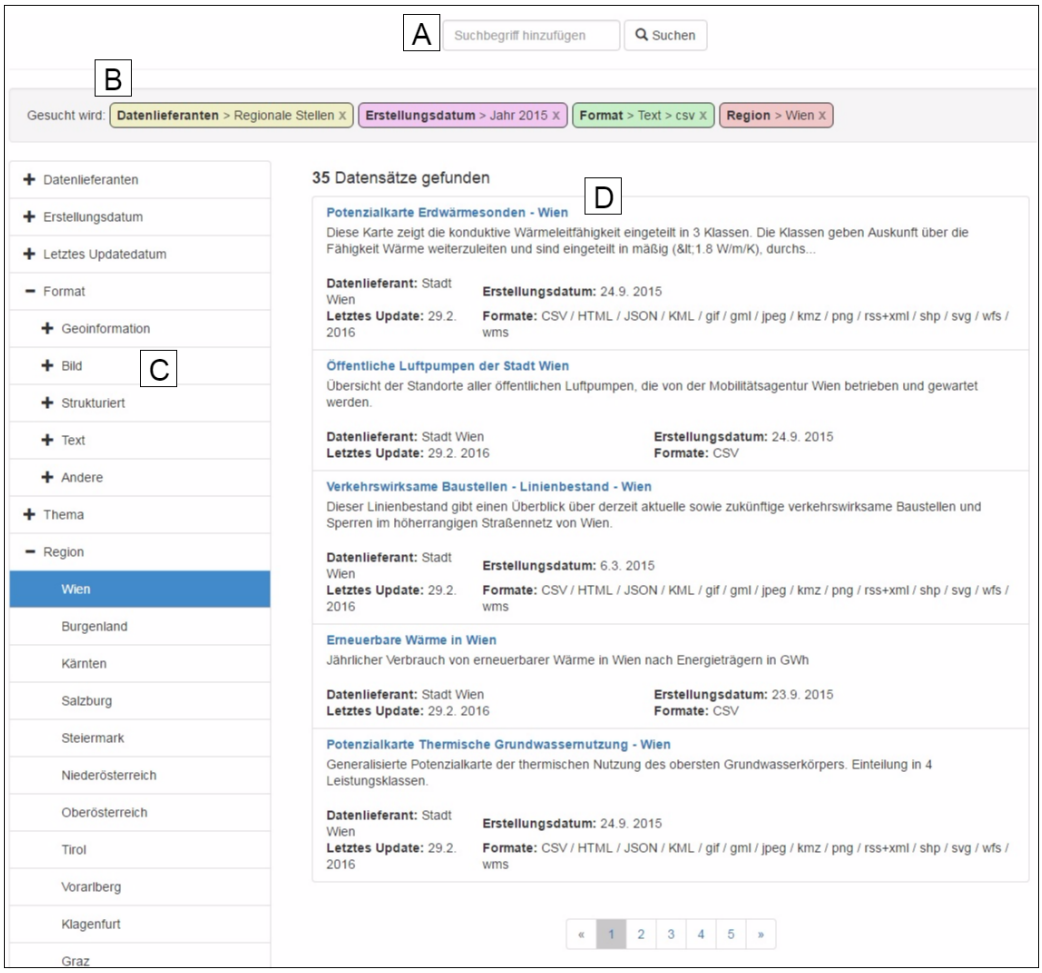

Figure 12: User interface of *Data Finder*

5.3 User interface

We are now going to discuss the User interface of *Data Finder*. The application can be accessed publicly[21]. Figure 12 shows a screenshot of its present state. Since a very high percentage of the datasets at hand are in German we decided to keep the user interface in German as well. At the top of the page (A) we see a textbox where the user can enter a searchterm. As soon as the search button ("Suchen") is pressed the input is checked for unwanted characters and, consequently, a new facet is created.
Beneath the search field the list of breadcrumbs is located (B). We can see that each breadcrumbs label consists of the name of the facet in bold, if

[21] `http://data.wu.ac.at/opendatasearch/`

needed, the facet value group and, lastly, the selected facet value. Coloring is based on the type of facet that is depicted. This was done with the intention to facilitate an effortless memorization of the search history for the user. The visual appearance of the breadcrumbs was inspired by the work of Marti A. Hearst [12].
On the left side of the page we can find the tree view where all the possible facets are displayed (C). The user can select a facet or a facet value group simply by clicking on it in the tree view. We decided to use this method of revealing hierarchy because we only have a maximum of three levels and for this depth the usability of this kind of representation is relatively high. If the number of facets respectively facet values increases significantly in the future a different approach could become necessary.
The iframe containing the results list is displayed at the center of the page (D) and it occupies the biggest part of the available screen space. At its top left corner the number of datasets that match the current selection of facets is shown. The main part of the frame consists of the list of datasets. We elected to make five resulting sets visible simultaneously. In order to review the remaining results the user has to use pagination at the bottom of the frame. Some useful metadata about each set is displayed. In this version six distinct aspects of each dataset are indicated:

1. Title: The title of each dataset is displayed at the top of each list item. Furthermore, this is a link to the actual dataset on the corresponding Open Data portal page.

2. Description: Then the description of the dataset is presented. It was limited to a maximum of 300 characters, since some datasets have very long descriptions and those can cause unwanted distortion of the user interface.

3. Publisher: the name of the publisher.

4. Date of issue

5. Date of last modification

6. Formats: All of the available formats of each dataset is displayed. They are separated by a backslash.

There is some more metadata available that could be of interest to the user. For instance, direct links to the downloadable files together with their size could be provided. Furthermore, the license under which the dataset is distributed could be of interest to some users. A possibility for future work

could be to reduce the amount of metadata that is displayed initially and to introduce a new panel that contains all the details about a specific dataset. This panel would be shown upon clicking on an item of the results list.
This was an explanation of the current state of the user interface of *Data Finder*. It was implemented based on the available screen space of an average desktop PC. In the future a version that is optimized to fit multiple screen sizes, for instance also a mobile device, would be a possibility worth thinking about.

5.4 Limitations

The current version of *Data Finder* is just a prototype that was created in the course of this bachelor thesis. The narrow temporal scope lead to quite a few limitations, which could potentially be addressed in future work.
One of the more significant limitations is the current number of datasets respectively the amount of Open Data portals that are covered by the application. There is a tremendous amount of Open datasets available on the web, and the Institute for Information Business of the Vienna University of Economics and Business could also provide much more metadata in the DCAT format. Covering more portals would definitely increase the usefulness and target group of the application. However, with datasets from non-German-speaking countries the aspect of multiple languages will have to be considered. This facet was left out almost completely in this thesis.
Another limitation is the current speed and efficiency of *Data Finder*. Currently there is a delay of a few seconds whenever a facet is selected. This is due to the complexity of the SPARQL query and the speed of the RDF store. This situation surely has room for improvement, since the current implementation is just one of many possible ways to address the problems at hand. For example, the filtering of the data does not necessarily need to happen at the level of the SPARQL query. It could be faster to load all datasets into memory of the server and do the filtering in the PHP script. Alternatively, the filtering process could also be completely redirected to the clients' side, which would be a fine solution as long as the amount of managed data stays at a manageable scope.
The user interface has also a lot of aspects that could be improved upon, ranging from better visualization of the facets and their impact on the current selection to a more efficient way of displaying the results list.
Clearly, there is a lot of progress to be made, however, the current prototype shows one possible and working solution for the problem of faceted search on RDF data.

6 Conclusions and Future Work

Searching Open Data is becoming an increasingly popular topic amongst researchers since the available data is growing at an ever increasing rate. Therefore, designing a well-rounded search engine respectively interface is going to be essential in order to reach the full potential of Open Data. This paper presented one possible way of solving this problem.
This thesis addresses this topic and proposes the use of faceted search over RDF data describing the content of Open Data portals. In the course of our work we developed a prototype, called *Data Finder*, that provides an interface for a search across six Austrian Open Data portals. We decided to employ the technique of faceted search in order to enable exploratory browsing of the search space, which is an approach that many end users take when navigating through Open Data [20].
It has been shown that there are many ways to implement a faceted search on a given set of items. In the current version of the prototype the filtering of results is happening at the most bottom level, namely the query to the underlying RDF store. This may be subject to change in future versions for efficiency reasons.
Data Finder was designed with the intention to provide a search over multiple portals. Clearly, the current number of covered portals is relatively small compared to the remaining ones available on the web. So, this aspect is one that definitely should be addressed in the future.
Another point is the "Theme" facet, whose current implementation is rather simple. The mere matching of keywords to categories only achieves a small portion of the navigational quality this facet potentially could have. An algorithm similar to Castanet [17] that automatically assigns facet values based on the textual description of each dataset could be used to drastically increase the usefulness of this facet. A tool like Babelnet [22] could be employed to achieve this feature. This is not an easy task, however, we feel like the benefits would be worth the work, since a facet like this could improve the search experience quite drastically, especially for a bigger search space.
The user interface is also to be improved in future versions of *Data Finder*. One feature that is a de facto standard of other faceted search implementations is showing the number of items the results list is going to be reduced to when each facet is selected. This number is usually shown in brackets next to the facet and indicates how restrictive each facet is at the current state of the search. This will contribute to decreasing the occasions where a user encounters an empty results list, which causes feelings of being lost and

[22] `http://babelnet.org/`

should be avoided if possible . Oren et al. [8] even propose adjusting the font-size based on the navigational quality a facet has at each state of the search, which is an approach worth considering.
Overall, this project, and the presented *Data Finder* prototype, constitutes a first step towards improving the search facilities across multiple Open Data portals, an open, interesting and timely challenge which deservers further research and efforts.

References

[1] G8 open data charter and technical annex. URL: `https://www.gov.uk/government/publications/open-data-charter/g8-open-data-charter-and-technical-annex`, 2013. [Online; accessed 06-August-2016].

[2] The open definition. URL: `http://opendefinition.org/`, 2016. [Online; accessed 25-July-2016].

[3] Andrew Wood Antoine Isaac, Pierre-Antoine Champin and Sandro Hawke. Rdf 1.1 primer. URL: `https://www.w3.org/tr/2014/note-rdf11-primer-20140624/`, 2014. [Online; accessed 30-July-2016].

[4] Tim Berners-Lee. Linked data. URL: `https://www.w3.org/designissues/linkeddata.html`, 2006. [Online; accessed 08-August-2016].

[5] Sören Brunk and Philipp Heim. tfacet: Hierarchical faceted exploration of semantic data using well-known interaction concepts. In *Proceedings of the International Workshop on Data-Centric Interactions on the Web (DCI 2011)*, volume 817 of *CEUR-WS.org*, pages 31–36, 2011.

[6] Richard Cyganiak. Data catalog vocabulary. URL: `https://www.w3.org/TR/vocab-dcat/`, 2014. [Online; accessed 25-July-2016].

[7] Richard Cyganiakn. The linking open data cloud diagram. URL: `http://lod-cloud.net/`, 2014. [Online; accessed 25-July-2016].

[8] Renaud Delbru Eyal Oren and Stefan Decker. *Extending Faceted Navigation for RDF Data.* The Semantic Web - ISWC 2006.

[9] Christiane Fellbaum. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[10] World Wide Web Foundation. Open data barometer global report. URL: http://opendatabarometer.org/, 2015. [Online; accessed 25-July-2016].

[11] Richard Hammell. Driving growth, ingenuity and innovation. Technical report, Deloitte LLP, 2012.

[12] Marti A. Hearst. *Design Recommendations for Hierarchical Faceted Search Interfaces.* ACM SIGIR Workshop on Faceted Search, 2006.

[13] Sebastian Neumaier Jürgen Umbrich and Axel Polleres. *Quality assessment & evolution of Open Data portals.* In IEEE International Conference on Open and Big Data, Rome, Italy, August 2015, Rome, Italy, 2015.

[14] Kevin Li Ka-Ping Yee, Kirsten Swearingen and Marti Hearst. *Faceted Metadata for Image Search and Browsing.* SIGCHI Conference on Human Factors in Computing Systems, 2003.

[15] Shiyali Ramamrita Ranganathan. *Elements of library classification.* Bombay: Asia Publishing House, 1962.

[16] Christopher Sahnwaldt Christian Herta Scott Robinson Michaela Bürgle Holger Düwiger Rasmus Hahn, Christian Bizer and Ulrich Scheel. *Business Information Systems*, chapter Faceted Wikipedia Search, pages 1–11. Springer, 2010.

[17] Emilia Soica and Marti A. Hearst. *Demonstration: Using WordNet to Build Hierarchical Facet Categories.* In ACM SIGIR Workshop on Faceted Search, 2006.

[18] Louise Spiteri. A simplified model for facet analysis: Ranganathan 101. *Canadian Journal of Information and Library Science*, 1998.

[19] Andy Seaborne Steve Harris and Eric Prud'hommeaux. Sparql 1.1 query language. URL: https://www.w3.org/TR/sparql11-query/, 2013. [Online; accessed 02-August-2016].

[20] M. Swamiraj and L. Freund. *Facilitating the discovery of open government datasets through an exploratory data search interface.* 2015 Open Data Research Symposium, Ottawa, Canada, 2015.

[21] Jürgen Umbrich and Sebastian Neumaier. Portal list open data portal watch. URL: http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm, 2016. [Online; accessed 25-July-2016].

[22] Liyang Yu. *A Developer's Guide to the Semantic Web*. Springer, 2010.

A Support Files

A.1 List of Regions

Table 3: List of Regions (regions.csv)

Region in German	**Region in English**
Wien	Vienna
Burgenland	Burgenland
Kärnten	Carinthia
Salzburg	Salzburg
Steiermark	Styria
Niederösterreich	Lower Austria
Oberösterreich	Upper Austria
Tirol	Tyrol
Vorarlberg	Vorarlberg
Klagenfurt	Klagenfurt
Graz	Graz
Linz	Linz
Innsbruck	Innsbruck

A.2 List of Keywords

Table 4: List of Keywords (themes.csv)

Keyword	**Theme**
Haushalt	finance
Budget	finance
Rechnungsabschluss	finance
Geld	finance
Bank	finance
Ausgaben	finance
rechnung	finance
Gesellschaft	people
Einwohner	people
Demographie	people
Geschlecht	people
Alter	people

Mitglieder	people
Personal	people
Arzt	people
Bürgermeister	people
Arzt	people
Spartenmitglieder	people
Umwelt	environment
Wasser	environment
Naturschutz	environment
Geodaten	environment
Wetter	environment
schutzgebiete	environment
klima	environment
bodennutzung	environment
WU	education
lecture	education
books	education
Bildung	education
courses	education
schüler	education
schule	education
Pflege	health
Krankenanstalten	health
Arzt	health
Medikamente	health
Apotheke	health
krankenhaus	health
Industrie	economy
Arbeitsmark	economy
tourismus	economy
haushalt	economy
bodennutzung	economy
bank	economy
versicherung	economy
verkehr	economy
Museen	art
Kunst	art
events	art

tourismus	art
kirche	art
Wahlen	politics
Recht	politics
Verordnung	politics
Gemeinde	politics
Grenzen	politics
Bürgermeister	politics
Bundesland	politics
rechtsform	politics
Standorte	geography
POIs	geography
Grundwasser	geography
Geodaten	geography
Grenzen	geography

A.3 List of Publishers

ra ... Regional authorities
si ... State institutions
pi ... Private institutions

Table 5: List of Publishers (publishers.csv)

Publisher	**Group**
Stadt Wien	ra
Stadt Linz	ra
Land Oberösterreich	ra
Land Salzburg	ra
Stadt Graz	ra
Stadt Klagenfurt	ra
Land Kärnten	ra
Land Tirol	ra
Land Niederösterreich	ra
Land Vorarlberg	ra
Land Steiermark	ra
Gemeinde Engerwitzdorf	ra
Stadt Innsbruck	ra

Stadt Salzburg	ra
Cooperation OGD Österreich	ra
Gemeinde Kremsmünster	ra
Stadt Wolfsberg	ra
Stadt Vöcklabruck	ra
regionalentwicklung.at	ra
Statistik Austria	si
Parlament	si
Sozialministerium	si
Umweltbundesamt GmbH	si
ZAMG	si
Wirtschaftskammer Österreich	si
BKA	si
BEV	si
Institute for Information Business at Vienna University of Economics and Business	si
Universitätsbibliothek	si
Österreichische Lotterien Ges.m.b.H.	si
KDZ - Zentrum für Verwaltungsforschung	si
mumok - museum moderner kunst stiftung ludwig wien	si
Geoland.at	si
BMVIT	si
Geologische Bundesanstalt	si
Wien-Ticket.AT	si
BMEIA	si
BMI	si
Rechnungshof	si
Österreichische Akademie der Wissenschaften (ÖAW)	si
Österreichische Post Aktiengesellschaft	si
BMF	si
BMWFW	si

Österreichische Mediathek - Technisches Museum Wien	si
Wirtschaftsförderungsinstitut der Wirtschaftskammer Österreich	si
AGES	si
HexaPlant	pi
Computerwelt, CW Fachverlag GmbH	pi
Greenpeace in Zentral- und Osteuropa	pi
BRAU UNION ÖSTERREICH AG	pi
HP Enterprise Business Hewlett Packard	pi
Semantic Web Company (SWC)	pi
IBM Österreich	pi
NEOS - Das Neue Österreich und Liberales Forum	pi
Klub der Köche Kärnten	pi
RTR GmbH	pi
Purkersdorfer Jagdklub	pi
open3.at	pi
Arbeitsgemeinschaft für Datenverarbeitung (ADV)	pi
UbiGo KG	pi
A-Trust	pi
zoomsquare GmbH	pi
HPC Dual	pi
Das Lastenrad	pi
School of Data Austria	pi
Die Grünen	pi

A.4 List of Formats

Table 6: List of Formats(formats.csv)

Format	**Group**
shp	geo

kml	geo
kmz	geo
gml	geo
wfs	geo
wms	geo
shape	geo
geotiff	geo
shx	geo
wmts	geo
geojson	geo
georss	geo
prj	geo
gpx	geo
sbn	geo
sbx	geo
ovl	geo
sld	geo
png	pic
jpeg	pic
gif	pic
svg	pic
jpg	pic
tif	pic
dxf	pic
glg	pic
jp2	pic
scv	pic
json	structured
html	structured
xml	structured
www	structured
rss	structured
xsd	structured
rdf	structured
wsdl	structured
trig	structured
sparql	structured
csv	text

pdf	text
txt	text
xlsx	text
tfw	text
rtf	text
ods	text
odt	text
asc	text
ascii	text
doc	text
ger	text
zip	other
dbf	other
glr	other
ordner	other
file	other
api	other
E-mail	other
xcl	other
asmx	other
link	other
otf	other
sup	other
mif	other
ress	other
aspx	other
sps	other
dat	other
mid	other
mdb	other
binary	other
ttf	other