Självständigt arbete på grundnivå

Independent degree project - first cycle

Computer Engineering

Development of a semantic data collection tool
The Wikidata Project as a step towards the semantic web

Ifeanyichukwu Ubah
Development of a semantic data collection tool
Ifeanyi Ubah 2014-06-05

MID SWEDEN UNIVERSITY
The Department of Information Technology and Media

Examiner: Ulf Jennehag, ulf.jennehag@miun.se
Supervisor: Magnus Eriksson, Magnus.Eriksson@miun.se
Author: Ifeanyichukwu Ubah, ifub1100@student.miun.se
Degree programme: Computer Engineering, 180 credits
Main field of study: Bachelor Thesis
Semester, year: 06, 2014
Development of a semantic data collection tool
Ifeanyi Ubah 2014-06-05

Abstract
The World Wide Web contains a vast amount of information. This feature makes it a very useful part of our everyday activities but the information contained within is made up of an exponentially increasing repository of semantically unstructured data. The semantic web movement involves the evolution of the existing World Wide web in order to enable computers make meaning of and understand the data they process and consequently increase their processing capabilities. Over the past decade a number of new projects implementing the semantic web technology have been developed albeit still in their infancy. These projects are based on semantic data models and one such is the Wikidata project.

The Wikidata project is targeted at providing a more semantic platform for editing and sharing data throughout the Wikipedia and Wikimedia communities. This project studies how the Wikidata project facilitates such a semantic platform for the Wikimedia communities and includes the development of an application utilizing the semantic capabilities of Wikidata.

The objective of the project is to develop an application capable of retrieving and presenting statistical data and also be able to make missing or invalid data on Wikidata detectable. The result is an application currently aimed at researchers and students who require a convenient tool for statistical data collection and data mining projects.

Usability and performance tests of the application are also conducted with the results presented in the report.

Keywords: Semantic web, World Wide Web, Semantic data model, Wikidata, data mining.
Foreword

I would like to extend my thanks and appreciation to my supervisor Magnus Eriksson for providing me with this topic and constant guidance and supervision as well as providing necessary information regarding the project.
# Table of Contents

Abstract .................................................................................................................................................. iii  

Foreword ........................................................................................................................................... iv  

1 Introduction ................................................................................................................................... 1  
1.1 Background and problem motivation ....................................................................................... 2  
1.2 High-level problem statement ................................................................................................. 2  
1.3 Detailed problem statement ..................................................................................................... 2  
1.4 Outline ....................................................................................................................................... 3  

2 Theory .......................................................................................................................................... 4  
2.1 The Semantic Web ................................................................................................................... 4  
2.2 The RDF and Semantic Web ..................................................................................................... 4  
2.3 Semantic MediaWiki .................................................................................................................. 5  
2.4 Wikidata ...................................................................................................................................... 6  
2.4.1 Semantic Structure of Data ................................................................................................. 6  
2.4.2 Time-line Deployment and Usage on Wikipedia ................................................................. 8  

3 Methodology / Model ..................................................................................................................... 12  

4 Design / Implementation .............................................................................................................. 14  
4.1.1 Translating Items to Labels ................................................................................................. 15  
4.2 Previous Implementations ......................................................................................................... 17  

5 Analysis and Results ...................................................................................................................... 18  
5.1 Performance ............................................................................................................................. 18  
5.2 Usability ................................................................................................................................... 21  

6 Conclusions .................................................................................................................................. 22  
6.1 Future Research ....................................................................................................................... 22  
6.2 Ethical Aspects ......................................................................................................................... 23  

References ....................................................................................................................................... 24
1 Introduction

The world-wide-web has become an increasingly important part of our everyday activities. It consists of inter-linked HTML pages, XML and dynamic content, multimedia and web services which dictate our means of communicating with each other, getting information about the world around us and entertaining ourselves at our leisure time. But the content being hosted on the Web is made up of semantically unstructured data primarily published for humans. Semantically unstructured in the sense that the data lack the semantic structure needed to facilitate efficient processing of this data by machines.

While the applications and web pages being published on the web are readily usable and readable by human beings, machines are not able to understand these content since, unlike humans they are unable to establish how the data relate to each other and consequently the data lacks all meaning to them. For example while a human is able to establish based on information presented on several websites, the current weather, local sports score or even population figures, computers are not able to do the same on the web because these data are presented in static HTML or other formats which are both human readable and understandable but only readable and not understandable by computers. The ongoing solution to this has been to implement a more semantic web of interlinked contents based on human and machine understandable data by including semantic content into web pages.

While on first thought, the idea of having a human-understandable-only based web may seem inconsequential since machines are originally not made to understand or make meaning of the data they serve, it is in fact impossible to look past the benefits of having them make meaning of the data they process. Various projects are currently being developed so as to make the existing web more semantic and one of such projects which implement a semantic model is the Wikidata project developed by the Wikimedia Foundation to serve as a common data repository for other Wikimedia projects. The aim of the project is to study how the Wikidata project provides the semantic platform for the other Wikimedia projects and develop a semantic data collection tool which may also be used to query the availability or accuracy of the knowledge base being hosted by Wikidata.

1.1 Background and problem motivation

The richness in information of the data repository being hosted on the web has made it one of the primary source of knowledge in most countries. But due to the non semantic data-hosting structure of the web, it is very difficult to retrieve spe-
specific information from its growing repository. This is the reason why it sometimes takes us a minute or two to find that person or that song on Google. This flaw is even more evident in trying to collect large amounts of statistical information on the web.

The process of gathering data be it for scientific research or otherwise can be a tedious task. Current tools developed for data collection on the web do not fully automate the process and require manual labor and periodical intervention from the user. The tool developed in this project aims to make data collection on the web as automatic as possible by implementing an effective model while maintaining a friendly user interface and minimal or no intervention from the user during the data collecting process. The tool aims to make large amounts of statistical data available to any user at as few mouse clicks as possible.

Inaccurate data collection can have a severe negative impact on a study and as a consequence yield inaccurate results. Since the process of data collection using this tool is meant to be automatic, it is of high importance that the data source is as reliable as possible. This involves making sure that the data being hosted on the Wikidata repository is valid and available. The tool facilitates data checking on Wikidata by making invalid and missing information detectable.

1.2 High-level problem statement
The overall aim of this project is to look into the idea of the current research and development project that is the semantic web as well as the Wikidata project and develop a data collection tool that makes searching for data on the web easier. The tool will be an application capable of lifting the burden of tedious data collection on the web off data gatherers and making the process more automatic and efficient. The aim of the application is to automate the process of data collection on the web and also to verify the validity and accuracy of the data hosted by the Wikidata project.

1.3 Detailed problem statement
In this report, I will analyze the application developed alongside the project. The analysis of the application is going to be used to decide the following factors.

- Data collection on the web is possible with the tool.
- Missing and invalid information on Wikidata is detectable using this tool.
- The tool is usable i.e users are able to understand and use the tool for a purpose.
The purpose of this analysis is to have an idea of how useful and usable an application developed primarily for the semantic web can be.

1.4 Outline

Chapter 2 contains the theoretical and background aspects of the idea of the semantic web and the Wikidata project. Chapter 2.1 presents a high level explanation of the concept of semantics and how it leads to the semantic web while 2.3 starts with the introduction of the Wikidata project. Chapter 3 introduces the methodology used in this project and chapter 4 presents a lower level illustration of the application's architecture. Evaluation procedures are presented in chapter 5 along with their results with chapter 5.1 focusing on the performance evaluation and 5.2 on the usability evaluation. Chapter 6 discusses the future of the application and the semantic web in conclusion.
2 Theory

2.1 The Semantic Web

What is Semantics?

“Semantics is the process of communicating enough meaning to result in an action”[1]. As humans, semantic is inherent in our language. We don't have to make any extra effort in asking a person on the street where the nearest bus station is or describe what we had for lunch to our friends. The meaning of what we say is perceived as soon as we say it.

In order to understand semantics in our language, we look at two examples

1) Lisa Loves Homewor kes

2) Homeworks Frighten Bart

Both sentences take the form of a simple “subject-verb-object” grammatical structure.

From our perception of both sentences we can tell that the words “Bart” and “Lisa” refer to not necessarily people that we know but people nonetheless. The word “Homework” refers to a kind of doable task and the words “Loves” and “Frighten” establishes the relationship between the person and the task. Once this relationship has been established we are able to understand the sentences because the chances are we know what a homework is (we've all had to do one at some point) and once we've understood both sentences, we are now armed with some new knowledge of the world around us. We are now able to answer the question “who likes homeworks?”.

2.2 The RDF and Semantic Web

The above example gives us the idea of how we as humans get to understand what we say to each other and also one way by which understanding could be implemented in the way computers communicate with each other. We use symbols to not only communicate data but also the meaning of the data and consequently affect the behavior of the computers.

Just as we infer meaning from the two sentences based on their grammatical structure, an abstract model can be used to infer semantic structure into data when communicating between computers.
(RDF) model which was introduced in 1999 was implemented for this cause on the web. It was developed to facilitate the interoperability between computers when communicating semantic information on the web. The model of the RDF allows for the description of virtually every content and their relations existing in the real world[2].

Using the RDF model, it is possible to interpret our earlier sentences as RDF statements consisting of a subject-predicate-object structure (known as a triple) in order for machines to understand and take action based on the meaning of the sentences.

While we humans are able to order a pizza on-line or check for the lowest train ticket price, computers are not able to complete such tasks without our intervention at one point or another since the web pages hosting these services are designed to be read and understood by humans and not computers. Having machines “understand” the data they process enables us utilize them even more effectively than ever before. We can have them work in more flexible and efficient ways by having them perform for example tedious works involving data gathering and finding and also enable them act upon the data they process throughout the web. This process of creating this semantically structured web involves upgrading our existing web into a Semantic Web.

The idea of the semantic web movement was proposed by Tim Berners-Lee who is also known as the inventor of the World Wide Web and it is currently being led by the World Wide Web Consortium (W3C). It aims at upgrading the existing web which contains a increasing repository of unstructured data into a web of structured interlinked data that can be processed and understood by machines alike. The goal is to in the long term establish a system whereby computers are able to understand and carry out complex human requests based on their meaning.

2.3 Semantic MediaWiki

Different applications and projects are being developed throughout the web to facilitate the establishment of the semantic web. One such project developed by the Wikimedia Foundation is the Semantic MediaWiki (SMW). The SMW is developed as an extension to the MediaWiki (which is the software powering the Wikipedia and other WikiMedia Foundation projects) to add a semantic platform to the knowledge base being hosted by wikis on the web. It is based on the RDF Semantic model [2]. The SMW allows the MediaWiki software to insert semantic data within wiki pages consequently making the MediaWiki semantic.

For example, in a semantic wiki, the page of an orange may contain standard text information about the fruit that is orange which is human readable along with some machine readable data. Since it is a “semantic” wiki, we want the machine to “understand” that this item is about “a kind of fruit” so we want to insert
Development of a semantic data collection tool
Ifeanyi Ubah
2014-06-05

a semantic annotation within the web page using an SMW property to connect the page to some other piece of data (the data being Fruit in the case of Orange).

The orange is the [[Instance of :: Fruit ]] of the citrus species Citrus x sinensis …

and is semantically the equivalent to “ orange” “is a” “fruit”. This data encoded within the page creates a relationship for this item orange and can be used by a machine to for example generate a list of fruits simply by listing all items that are listed as instances of fruit.

The idea of extending the semantic platform to Wikipedia was proposed in 2010 since the vast amount of knowledge hosted on the Wikipedia base was dominated by unstructured data and published in plain text that was only available for processing by humans. The problem was that the data, especially the statistical ones were going to waste as they couldn't be processed with meaning since they lacked the necessary semantic structure to dictate how they were related to each other. The solution of implementing the Semantic MediaWiki proved not to be a viable solution for performance reasons and in April 2012 the Wikimedia Foundation started to develop the Wikidata project[3].

2.4 Wikidata

The Wikidata project is a relatively new project created by the Wikimedia Foundation. It is intended to be a “free, collaborative, multilingual, secondary database, collecting structured data to provide support for Wikipedia, the other Wikimedia projects, and well beyond that”[4]. Support well beyond that meaning that it is also intended for use by various other different services and applications instead of having its services limited to just the Wikimedia projects. The data hosted on Wikidata is maintained based on the semantic model of the project and is readily available in other languages. This enables services and applications making use of the project possess semantic capabilities.

2.4.1 Semantic Structure of Data

In Wikidata, virtually every real-life topic is represented and corresponding to all topics which Wikipedia articles are about regardless of the language edition. The semantic capabilities are added by creating relationships between the data representing these topics. To create relationships, specific identifiers are used.

Title

Every element representing a topic on Wikipedia is given a title on its corresponding Wikidata element. The title, also known as the Entity ID of an element is given to every item and property hosted on the Wikidata database and is used to uniquely identify it.

---

1 http://en.wikipedia.org/wiki/Orange_(fruit)
Text from wikipedia “The orange (specifically, the sweet orange) is the fruit of the citrus species Citrus × sinensis”
Item

An item is used to represent a real-life topic on Wikidata. This may range from an object to an emotion. All items are identified by their unique language-independent Entity ID beginning with the letter Q followed by a number. Sweden as an example is assigned the ID Q34 while its Prime Minister Fredrik Reinfeldt is identifiable by Q52920. Each item contains a label and description through which it may be recognized by humans. The description is a short text explaining what the item is about. The label is usually the name by which the item is known in the real-life scenario while description is used to clarify the item and differentiate it from any other items with the same label. The label-description combination is used to minimize the ambiguity of labels for different items representing different real life topics but may be known by the same name and consequently have the same labels. For example the items Q3196 and Q2121525 both have the same label “Fire” but based on their description a user is able to differentiate the former as the “rapid oxidation of a material” from the latter (a song by U2).

Property

A property is used to describe a quality of an item in Wikidata. It is the base upon which relationships between items are formed. They are used to assign value to the quality of an item. For a simple item “car” may need to be described by its color, hence the property “color” will be used to assign the value “red” to the item. This property “color” should not be mistaken with the item also named “color” which is a noun (defined as the visual perception of light wavelengths) and has its own unique identifier. As a result properties are assigned their unique Entity IDs starting with P along with a number. So “color” the property will be identifiable by P462 and the item “color” by Q1075.

Claims and Statements

Claims in Wikidata are used to show a relationship between two items in a human readable way. With claims it is possible to show that one developer of Facebook (Q355) is Mark Zuckerberg (Q36215). They are also used to assign values to properties for a certain item. Property-value relationships are used to establish a claim for a certain item. Example of a property value relationship within a claim is that the population (P1082) of England is 53,012,456 and its country calling code (P474) is +44. Another way through which meaning is given to the items are qualifiers. Qualifiers are used within claims to give more clarity to a particular claim. For example, with qualifiers, we can clarify the England population figures by using the point in time property (P585) with the value 2011 and the determination method (P459) to be census (Q39825). References are also optionally assigned to claims to clarify the source of the claim. A claim (consisting of a property-value pair) along with optional qualifiers or references make up a statement as shown in the Figure 1 below.
2.4.2 Time-line Deployment and Usage on Wikipedia

Deployment of the services and support offered by Wikidata to other projects was split into 3 phases [5].

The first phase is to provide a centralized interwiki system for interwiki links as an alternative to the former system.

The second phase involves collection of infobox-related data with a goal of providing a centralized repository for infoboxes on Wikipedia.

The third phase involves implementation of a system to allow for automatically updating and translation of list articles.

---

The first phase of the deployment process was to replace the “then” current system of interwiki language links with a centralized system. This would mean that all Wikimedia projects will offer their multilingual support based on the centralized support offered by Wikidata. The deployment on Wikipedia was completed in March 2013. On Wikipedia, the inter-language link system is used to link articles of the same topic in different language editions of Wikipedia (represented by the red dots in Figure 2). Instead of having to explicitly list every interwiki link in every of the article and consequently creating a mesh of duplicating interwiki links between the various editions all linking each other, the Wikidata knowledge base contains an item representing the same topic as well as a list of the article links in other Wikipedia language editions about that topic. This way, it serves as a centralized repository of interwiki links for the various Wikipedia editions. When the article for that topic is requested regardless of the language version, the software running on Wikipedia queries the Wikidata knowledge base for the list specific to the item such as in Figure 3 and presents it along with the article.

Figure 2: Wikidata Phase 1

![Image](http://www.wikidata.org/wiki/Wikidata:Introduction)
The second phase of the deployment of Wikidata on Wikipedia involves the population of the info-boxes on Wikipedia pages with data from the Wikidata repository. Info-boxes in Wikipedia are used to present a summary of information about the subject of the article being viewed by the user. The info-box for the Abisko National Park in Figure 4 will tell us in a glance, where it is located, when it was established, the governing body among other information. The Wikidata database serves as a centralized repository as in Figure 5 for data regarding the specific article which can be used to populate the info-box for that article. Having the Wikipedia application extract these information from the database upon the article's request by a user ensures that regardless of the language version, all Wikipedia editions have access to the same consistent data.

Figure 3: Layout of a Wikidata page showing the links to articles of the same topic in various language editions of Wikipedia

http://upload.wikimedia.org/wikipedia/commons/1/13/Wikidata_layout_Phase_I.png
Also, updating these values are made much easier and efficient since the update only has to be done in one place and the change takes place automatically in all Wikipedia editions of the article. Should the geographical coordinates of the national park change, the update only needs to be made in the Wikidata entry for the statement corresponding to the property-value relationship for coordinate location (P625) for item Q953092.

3 Methodology / Model

In order to enable users make advanced data query on the web, an application will be developed in this project. The application will be a data collection tool which may be used by students and researchers or hobbyists who require time consuming statistical data gathering to simplify the tedious process of having to search for the data on the web. The application should make the data collection process as easy as possible for the user by offering an interactive and simple user interface. The results of collected data will be presented in a convenient format that not only allows for further automated processing but also allows the user to detect if there are any inconsistencies within the result. For example, the researcher or student should be able to generate a list of population figures for all cities in Japan with minimal or no intervention during the process and detect if the population figure for any of the cities is missing or incorrect. This feature allows the application to be able to detect incorrect and missing data on the Wikibase repository which is the database of the Wikidata project.

One purpose of the application developed in this project is to tell how well a computer can make meaning of the semantically structured data being hosted on the web today with the Wikidata project as a use case. The long term goal of a semantic web involves machines being able to carry out tedious and complex human requests with minimal or no human intervention so the application is going to serve as a token of how far this goal is from realization or how close it is to fruition. The application developed in this project takes advantage of the semantic capabilities of Wikidata.

The application functions by making queries to Wikidata using the Wikidata API [6] and generating a presentable list to the user based on the inputs and responses from Wikidata. For each list generated, the user may request additional information for the items in the list by specifying their properties to the app. For example, in a list containing specific cities, the user may want to get each city's data for population, date of foundation, head of government, official website etc as they are existent in the Wikibase repository. The Wikidata property ID is used to specify these properties by the user.

To answer the question of how well machines are able to utilize the semantic model being implemented on the web today evaluation tests are carried out on the application in the form of usability and performance tests. The performance tests will give an idea of how efficiently machines can figure out how semantically structured data are related to each other and perform flexible data processing with them while the usability tests includes a survey with an objective to
find out among other things, how much human intervention the application needs to handle complex requests as well as how easy or difficult the process of generating data with the application is. The application will be a web based application in order to make it more accessible to users and will not require any installations.
4 Design / Implementation

The application is developed on the Java JSP Web framework. It gathers data based on user inputs, the user interface of the application shown in the Figure 6 allows the user to construct a query through a form. This query will be used by the application to generate a live list which will be downloaded in CSV format.

![User Interface of the application](http://ace-coda-577.appspot.com/)

**Figure 6: User Interface of the application**

The first input field is used by the app to generate a preliminary list of items. If the user wants to collect a list of specific countries, the EntityID of country Q6256 is inserted into this field.

The preliminary list generated by the root item uses the MediaWiki special “what links here” facility to generate a list of items. Using the Wikidata API [6] this feature is called backlinks and making a backlink query to Wikidata returns a list of “backlinks” that is a list of items linked to the root item. To get the preliminary list of items linking to the item Q6256, The app makes the Wikidata query over HTTP

```
http://www.wikidata.org/w/api.php?
action=query&list=backlinks&format=json&bllimit=max
```

All responses to queries made to Wikidata are in JSON format[7].

A query is then made to Wikidata to retrieve all statements for each item. The response consists of a JSON Object.

---

The second input field is used to filter the backlink list based on statements inserted by the user. The statement format in this field consists of only a property-value pair and multiple statements are added using the && symbol while values are combined using the || symbol. For example to filter the preliminary list of countries and return only countries from Africa and Europe, the following statement “P30=15||46” is inserted to represent “continent=Africa||Europe”.

Based on the specific statement inserted by the user, the application filters the list leaving a list of requested items. Accuracy of the generated list is based on the ability of the user to make efficient claims, hence basic knowledge of Wikidata terminology and claims-statement structure as explained in chapter 2.3 is required to use this application.

The third input field in the list is used to collect characteristic data about items on the list. If the user requires information like the country’s official language, population or currency, the Entity ID for these properties are inserted in this field combined using the && symbol (P37&&P1082&&P38). This field is optional.

For each entity that fulfills the criteria set by the user (Entities containing the property-value pairs inserted into the form) An Entity Object is created with its ID and its JSON object (jobject) as constructor parameters.

```java
if(checkClaims(claims)){
    Entity e = new Entity(q,jobject);
    EntityList.put(q,e);
}
```

### 4.1.1 Translating Items to Labels

The JSON object (jobject) for an entity contains all its statements with each statement represented by a JSON Array and consisting of a property-value pair among other things. For properties which have an item as a value, the Entity ID is used to represent that item. The JSON Array for the continent (P30) statement for the entity Sweden (Q34) is used as an example.

```
"P30": [
    {
        "id": "q34$A34546B8-A940-4ED1-8077-C13C6E55CC58",
        "mainsnak": {
            "snaktype": "value",
            "property": "P30",
            "datatype": "wikibase-item",
            "datavalue": {
                "value": {
                    "entity-type": "item",
                    "numeric-id": 46
                },
                "type": "wikibase-entityid"
            }
        },
        "type": "statement",
        "rank": "normal"
    }
]```
To collect the value of this property the “numeric-id” object is used to represent the item Europe (Q46) with the value 46. Since this data is represented by its ID only, it is not human readable to populate a list with IDs like Q46 instead of simply “Europe” in the column for continent.

Logic within the application recognizes this and translates the ID value to its human recognizable name. This is done within a method that makes a query to Wikidata to retrieve the label for an item based on its ID.

http://www.wikidata.org/w/api.php?
action=wbgetentities&format=json&ids=Q46&props=labels&languages=en

And the response from Wikidata will contain the label of the item "Q46":

```json
"id": "Q46",
"type": "item",
"labels": {
  "en": {
    "language": "en",
    "value": "Europe"
  }
}
```

Translation from item IDs to labels makes sure that the generated list as shown in figure 7 is human understandable. Instead of having a list populated with Wikidata Entity IDs, the user can instantly understand the contents of the list and also point out irregularities that may exist within the list. If the currency of a particular country in a list as shown in figure turns out to be incorrect as it is stored in the Wikidata repository, the user can detect this almost immediately.

![Figure 7: Generated list screenshot](image_url)
Values represented by the string “N/A” in the generated list are missing in Wikidata. The last column in the Figure 7 shows the population data for countries in Africa and Europe as stored in Wikidata and it is evident from the list that the population values for most countries as missing in the Wikidata repository.

4.2 Previous Implementations

A similar tool used to generate list of items based on a Wikidata Query is the Autolist tool [8]. This application makes use of a Wikidata dump as its source and generates lists by running queries on items existing in the local dump source. The Autolist does not make use of the “what links here” facility to generate a preliminary list especially since it makes use of a local Wikidata dump. Instead it runs the query directly from the user input on its local database. The user interface of this tool in Figure 8, although somewhat more complicated compared to that developed in this project, is also based on a form however, they are based upon different input formats. For example properties are combined using commas for the Autolist while they are combined using the and operator (&&) on the developed app. The Autolist also requires the user to make a direct query using its form which adds to its complicated interface and increased learning curve but this feature allows it to make complex nested queries on the Wikidata database resulting into more specific lists generated. The project app requires only the root item and required statements while the query is generated automatically at the back end. Hence the user does not have to know how to make a query using the Wikidata API in order to use the project app but it consequently does not offer support for complex and nested queries. Other feature are included in the Autolist like running queries on Wikimedia commons and Wikipedia for example as well as support for various language editions. This enables the list generated by the Autolist tool to include a link to the Wikipedia page for each item in the list.

![Figure 8: The Autolist Tool](http://tools.wmflabs.org/wikidata-todo/autolist.html)
Analysis and Results

5.1 Performance
The performance evaluation for the application focuses on speed tests in terms of how long it takes the user to generate a list. One aim of the test was to determine how to best affect the response time of the user. This test was initially based on two factors regarding the size of the list.

- How many items are on the list and
- How many properties the user requests for each item in the list.

In carrying out the tests, 4 queries (Q1 to Q4) were used to generate 4 different lists. The queries are all based on generating the same preliminary list so as to get as accurate a result as possible while the statements conditions for filtering the items on the list were different resulting in a different list for the four queries. The preliminary list had a total of 500 items and for each query the response time (time between the user clicks send and the download able file is ready) is measured for the two factors mentioned above. The measurements were carried out at three different times to account for more accuracy in the results and the average response time is calculated.

Scenario 1
- Preliminary list items -500
- No additional properties included in query.

<table>
<thead>
<tr>
<th>Query</th>
<th>Generated List Items</th>
<th>Avg Response Time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>2</td>
<td>12.34</td>
</tr>
<tr>
<td>Q2</td>
<td>15</td>
<td>15.55</td>
</tr>
<tr>
<td>Q3</td>
<td>45</td>
<td>16.72</td>
</tr>
<tr>
<td>Q4</td>
<td>150</td>
<td>18.13</td>
</tr>
</tbody>
</table>

Table 5.1: Scenario 1 results

Scenario 2
- Preliminary list items -500
- Additional properties added included with query.
Table 5.2: Scenario 2 results.

Based on the results for both scenarios as shown in Table 5.1 and Table 5.2, it is difficult to predict how much of an effect the size of the generated list has on the response time. This is mainly due to the programming architecture of the application. In order to eliminate unnecessary requests over HTTP to the Wikidata server and consequently waste time and processing power, the whole JSON file for all 500 items are downloaded upon initial query and the filtering option involves eliminating items from the list of 500 that do not contain the required statements based on the local file. The result of this is evident in the slight variation of less than 2 seconds between response times for the list generated by query Q3 and Q4 which is over 3 times its size.

In the case of scenario 1 the table shows that the number of generated list is directly proportional to the response time but this is not the case when the list is generated along with additional properties. Remember that the properties are identifiable by their Entity ID and for each JSON response received from Wikidata, items are represented by their Entity ID. The inclusion of properties requires that the application translates the data response by getting the label for each item identified in the list. This is done in batches over HTTP and duplicate requests are avoided. Hence if there are a lot of duplicate items that need to be translated, the request to Wikidata is not necessarily larger. For this reason the response times for generated lists is affected by the number of properties included but both factors are not directly proportional.

The two test scenarios were not able to provide a clear lever for directly affecting the response time of the application so a third factor was included.

The size of the preliminary list.

**Scenario 3**
- Preliminary list items 150
- Additional properties included with query
From the table 3 results for scenario 3, it is evident that the response time for the application is drastically reduced and that the size of the generated list also has reduced. This is due to the fact that the size of the preliminary list based on which to filter for results was changed from 500 to 150. Since there is a smaller list to choose from, the app has a smaller file to scan through hence the response time is directly affected. But the shorter response time obviously comes at a price which is that a smaller list of items is generated. For this reason, the user interface now contains an option to specify the size of the preliminary list along with the form inputs. This allows the user the choice to trade-off the size of the list for speed or vice versa. A user making the query Q1 for example will benefit from choosing to filter the preliminary list of 150 items in less than 4 seconds since the number of generated items is not affected. Or if the user making query Q4 needs just about 40 items in his list he can choose the same option in order to get his result 10 seconds earlier.
5.2 Usability

Usability testing of the application will also be carried out along with the project development process. The tests are designed to expose the extent to which the application can be understood, used and liked by a user. As a result, a questionnaire based usability test has been created based on the software usability measurement inventory (SUMI)[9]. The aim of the questionnaire is to measure the application's perceived quality of use from a developer and user perspective and questions are based to reflect several qualities of the app such as

- Efficiency- The degree to which the application responds to the user's request and interaction.
- Learn-ability- How easily the user can understand and learn to use the product.
- Helpfulness- The degree to which the application seems to be of domestic or professional assistance to the user.

The results of the questionnaire are presented in the Table 5.4 by categories based on the nature of the question asked in the survey. Reviewers were asked to grade on a scale of 1 to 5 with one being the most negative result and 5 being the most positive.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Efficiency</th>
<th>Presentation</th>
<th>Learn-ability</th>
<th>Stability</th>
<th>Helpfulness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median</td>
<td>3</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
</tr>
</tbody>
</table>

Table 5.4: Average Rating from Usability test

The aim of the usability test is to continually gather responses and feedback from a user and developer point of view and adapt the functionality of the application based on the results perceived from that testing community.
6 Conclusions

The main contribution of this project is the development of the client web application which can not only be used for data collection on the web but also detect irregularities within the data being hosted on Wikidata. The benefits of upgrading our existing web into a more semantic web are undeniable. It encourages a more scalable web. Instead of having an increasingly growing interlinked repositories of contents, a semantic web based upon the semantic homogeneity of its contents allows for a single repository serving as a large database from which we are offered services.

Based on the outcome of the project, it is possible to see what ways a more semantic web will be useful to us. The web application demonstrated with what ease computers can perform tedious time consuming data gathering in the place of a human and although we were able to find out that basic knowledge of semantics is required, as the semantic web movement and standardizations of its technologies gains more popularity with time, we will be able to see more user friendly semantic applications that offer better, more efficient and innovative services in terms of data and information sharing to users. Results of web search engines become less ambiguous and increasingly exact since knowledge representation systems are exported to the web.

Although technologies based on semantic web are still in their early stages, there seems to be a positive accord in terms of the future of the semantic web. However several hinders exist in the course of it’s implementation. There is no global format for publishing and processing semantics throughout the web although the standardizations put forth by the W3C body are gaining tractions. Also the massive size of the web means that the complete evolution of our web into a semantic one will take a little longer than we may want it to.

6.1 Future Research

The generation of the preliminary list is based on the “what links here” facility, this has proven to be a less efficient solution of querying Wikidata because the response generated is a list of all items that links to the root item. For example the backlink list of country Q6256 not only presents a list of countries but also every item linked to the country item and it is up to the application to sort throughout this list in order to make sure each item is a country and get rid of irrelevant results before filtering consequently wasting the processing time and capacity of the machine. A future implementation of this tool may involve a solution which uses a more efficient alternative to the backlinks facility and instead of returning a preliminary list of items that links to the root item, it return only a list containing instances of the root item. Such an implementation should only receive a list of country items when the value Q6256 is used as the root item.
Future development of the application will include features allowing the generated lists to include the links to Wikipedia articles for all items on the generated list as well as audio and image support for list items. This will involve integrating the application with the Wikipedia and Wikimedia Commons API. These features have not been implemented yet as a result of lack of time but it will be in the future.

6.2 Ethical Aspects

Although there is a strong desire for a World Wide Web that has a humanistic element in the sense that it can “know” or make meaning of the data it processes just as we humans do, there are also some ethical factors to consider. The first being in the case of the project application and other future applications that offer similar services. The idea is to make unlimited amount of information readily available to everyone with minimal effort. While this may seem to be a positive feat from almost every perspective, several steps may have to be taken into making sure there is a degree of privacy and data confidentiality especially in government security and health care sectors and that this information does not fall into wrong hands and end up being used for malicious purposes.

Another aspect to consider with semantic data collecting applications comes with the fact that the retrieval of knowledge using this tool is automatic. This feature comes with the risk that people will tend to rely solely on the tool as their source of knowledge and as a result, their ability to work creatively and curiously may deteriorate since they become fully accustomed to the automated process of generating data and do not have to make use of their logical and critical thinking ability when searching for data.
References

[1] Toby Segaran, Colin Evans, and Jamie Taylor
“Programming the Semantic Web”
Published by O'Reilly Media Inc 2009
ISBN: 978-0-596-15381-6
Chapter 1: Why Semantics? Page 3-18

Published 2011 Springer-Verlag Berlin Heidelberg
e-ISBN 978-3-642-15970-1
Chapter 2: The Building Block for the Semantic Web: RDF page 19-78

Retrieved 2014-05-24

Magazine
July/August 2013 pg 90-95

Retrieved 2014-05-12

http://www.wikidata.org/w/api.php
Retrieved 2014-05-10

[7] Introducing JSON
http://www.json.org/
Retrieved 2014-05-22

[8] Autolist Wikidata
http://tools.wmflabs.org/wikidata-todo/autolist.html
Retrieved 2014-05-21

[9] Drs. Erik P.W.M. van Veenendal CISA “Questionnaire Based Usability Testing”,
Published in Conference Proceedings European Software Quality week,
Brussels November 1998