Extracting Scientists from Wikipedia

Gustaf Harari Ekenstierna
Lund University
Lund, Sweden
dat11gek@student.lu.se

Victor Shu-Ming Lam
Lund University
Lund, Sweden
dat11vla@student.lu.se

Abstract

The Internet is, among other things, a very large and continuously growing source of information and knowledge. This knowledge can be found in the form of text, images, databases, tables, etc. In this article, we describe a system that gathers information from Wikipedia articles and existing data from Wikidata, which is then combined and put in a searchable database. This system is dedicated to making the process of finding scientists both quicker and easier.

1 Introduction

The amount of users on the Internet is growing at a steady rate and so does the amount of information. Statistics from Wikipedia, which describes itself as "the Internet's largest and most popular general reference work", show that the amount of text has been growing at a linear rate since 2006 (Wikipedia, 2015). This naturally makes things harder to find, which forces us to come up with new ways of managing the data to make searching, sorting and presentation of it easier. Wikipedia uses a system called Wikidata as its backbone for this job. It is not complete however and is missing a substantial amount of information that is present in the text of the articles.

Take Veikko Antero Koskenniemi, a Finnish poet, as an example. In the excerpt below, from his Swedish Wikipedia article, he is described as a literary historian:

> Veikko Antero Koskenniemi, född den 8 juli 1885 i Uleåborg, död den 4 augusti 1962 i Åbo, var en finländsk, finskspråkig författare och litteraturhistoriker.

This fact is confirmed by the article in the English version:

> Veikko Antero Koskenniemi (8 July 1885 – 4 August 1962) was a Finnish poet born in Oulu. In 1921, he took the title of Professor of Literary History in University of Turku, Finland. In 1948, he became a member of the Finnish Academy. He died in Turku.

while his Wikidata profile in Fig. 1 shows a list of three occupations where "literary historian" is absent.

Figure 1: Excerpt from Wikidata entry for Veikko Antero Koskenniemi

This issue means that we miss data unless we search both sources. One way of improving the search results is to encourage users to manually register information in the relevant data structures when adding new content, but this is by no means a seamless nor practical solution when considering the large volumes of text. If we want to apply this search to complete digital libraries, the process of taking information from text and putting it into a structured and easily searchable source needs to be automatic.

In this paper, we describe a system, which performs a simple text analysis to extract targeted pieces of information from a given text source. We designed a test procedure to evaluate the contribution of text and early results of our work show that another 70% of information can be found in text in addition to the information contained in Wikidata.

2 The Goal

Extracting people from specific categories is a frequent task: For instance, people from Berlin or Poland, or politicians, etc. Encyclopedias are relevant sources for this task, but as people are often not exhaustively categorized in the text, one needs to read it to find their categories.

If such work could be carried out by programs, we would be able to collect a lot more useful data and statistics with less work. That is what we aim at achieving using a simple text analysis. Instead of reading the articles one by one, we pre-process all the relevant data to create a comprehensive database that can either confirm or deny a person's status as a scientist and point out the references.

For this specific project, we have chosen to extract scientists from Wikipedia as it is a substantial group of people that most of us find interesting. More importantly, Wikipedia has a lot of information provided by the Wikimedia Foundation that we can process. We have created a proof-of-concept by extracting information from text and adding it to an easily searchable database presented on our project website.

To achieve this goal, we had to split our idea into two parts. The first step was to gather all data we needed and to extract only the relevant parts. The second part of our project was to present and make use of our results. We created a web interface where a user can search and explore the gathered information. The website provides a comprehensive preview of our results. However, there are several other ways that our results can be put to use so the website is mostly to demonstrate a case where information extraction can be really useful.

3 Related Work

The idea of extracting information from text is not new. A lot of projects extract information for computational knowledge systems, virtual assistants etc. One of the more known projects is Pantheon 1.0 (Yu et al., 2016), a manually verified dataset of globally famous individuals. They used Freebase(Bollacker et al., 2007) which is a knowledge database very similar to Wikidata and 277 language editions of Wikipedia to produce the dataset. Their first step which was getting the raw data used a process similar to ours by first collecting people entities from Freebase and then mapping those to their Wikipedia articles.

Another large project in language processing with focus on Wikipedia is DBpedia (Lehmann et al., 2014), which aims to extract information from Wikipedia articles into structured data. The resulting data is in many ways similar to Wikidata. DBpedia is limited to Wikipedia data and will only change if the article also changes unlike Freebase which extracts data from multiple sources and combines it with user data to provide a structured data source.

4 The Scope

We are using Wikidata and Wikipedia as information sources. They make a great fit for our project because they are free, closely connected, and contain a lot of information in many different languages. For this paper, we have chosen to limit the scope to content that has a Swedish translation. This decision was not based on any technical limitation. In fact, the project website contains data extracted from both English and Swedish. The tools can easily be adapted to work in any language. But due to time and hardware constraints, we chose to narrow it down to only Swedish to be able to present accurate statistics.

Wikidata is a knowledge database run by the Wikimedia Foundation. It is easily searchable due to its clear structure and therefore makes a good starting point for our project. We used a JSON copy provided by the WikiParq project (Klang and Nugues, 2016). Each entity is represented by an object which holds key-value pairs. The key is called a property, e.g. "occupation" which is an area we are interested in, while the value usually refers to another object, e.g. "scientist". See Fig. 1 for an example.

Wikipedia is an encyclopedia containing about 3 million articles in Swedish and many more if including other languages (Meta, 2016). We used a local copy of the Swedish Wikipedia, also from the WikiParq project. Each article has its own JSON object containing the article's identifier, label, text etc. This makes the data ready to parse since we did not need to strip the text from HTML tags beforehand that otherwise exist in the articles.

Having explained the data sources, it is time to think about what makes a scientist? We decided to base our definition on what the Wikidata says. Each object has a property called "subclass of" which refers to a parent object. With that in mind, we started with the scientist object as the root and simply built a tree from it and all of its subclasses, more on this in Sect. 6.1.

One may object that accurately defining a scientist is not as simple. Our method simply builds on Wikidata and is not more arbitrary than the ontology it proposes.

5 Tools

We used open-source tools in this project. We will go through them below to explain what they are as well as what we use them for and how we use them.

Apache Spark is a great tool for managing very large amounts of data that can be distributed across several machines. We used the Spark SQL Java API to interact with the data from our programs.

To show our results we have deployed a standard Apache web server to provide a simple and accessible user interface, from which users can search the data from our project.

6 The Processing Pipeline

The entire process of extracting scientists is split into three steps, each one having its own program. All three are made by us in Java and are utilizing the Spark SQL Java API.

6.1 Scientific-Professions-Finder

The purpose of this program is to find all scientific professions that exist in the Wikidata. This part is a tree builder which finds all subclasses of a given object. This given object, the root of our tree, is "Q901", the ID of the scientist object. The following query is the first one to be executed after loading the Wikidata into a table.

```
SELECT id, labels FROM professions
LATERAL VIEW explode(property) prop AS p
WHERE p.key = 'P279'
AND p.value = 'Q901'
```

The result of this query contains the ID and a list holding the title in different languages of each profession that is a subclass to scientist. The process is then repeated for each profession it has found until no more subclasses exist. A slightly stylized snippet of the final output can be seen below. It is in English since many professions lack a Swedish title and would not make a good example.

```
,- Q901:scientist
|---, Q169470:physicist
|    |---, Q752129:astrophysicist
|    |   '--- Q2998308:cosmologist
|    '--- Q6804564:mechanician
|
```

6.1.1 Wikidata Searcher

The purpose of Wikidata Searcher is to find all people who have one or more scientific profession(s) registered as an occupation, i.e. to find all scientists. It searches through the Wikidata by passing queries to Spark and uses the list of scientific professions given by the previous program. That list may however hold duplicates since it contains a tree structure. Those duplicates need to be removed beforehand, along with the dashes used to indent the rows. This can be done with a simple shell script or similar.

The table of people it is searching in is limited to those who have Wikipedia articles in Swedish, as well as being born between two dates specified by the user. The reason for having these delimiters is to reduce RAM usage and execution time since we are using a single machine to run the programs on. This table will be referred to as the "limited table". Below is an example of a query which returns a list of all people in our limited table who have "scientist" as a registered occupation.

```
SELECT limitedTable.id
FROM limitedTable
LATERAL VIEW explode(property)
prop AS p WHERE p.key = 'P106'
AND p.value = 'Q901'
```

This query is repeated for each scientific profession. Wikidata searcher will create two files once it is done, the first one holds a list of scientists which contains their IDs and scientific professions. The second file holds a list of all the people that were in the limited table, scientists or not. This file will be used by the next and last program. Below is a snippet from the output file containing scientists.

```
urn:wikidata:Q169330
|-Q169470:fysiker
urn:wikidata:Q69571
|-Q593644:kemist
urn:wikidata:Q1702846
|-Q42973:arkitekt
|-Q1792450:konsthistoriker
```

6.1.2 Wikipedia Searcher

The third and final program tries to find scientists in Wikipedia articles by using simple regular expressions. The Wikipedia Searcher loads the file containing a list of people that came from the previous program. It then queries Spark, one time for each person to find his or her article. Each article is analyzed with regular expressions to determine if the person was or is a scientist. The regular expression search is performed once for each profession. An example of what the regular expressions can look like is seen below.

```
"(var|är)\\s.{0,40}kemist"
```

The program, using the regular expressions above, will look for the existence of a substring in each article that matches "was" or "is", followed by a space, up to forty characters of any type and ending with "chemist".

The Wikipedia Searcher will create two files once it has gone through the list of people. The first file contains what has been found and is using the same format as the Wikidata Searcher but with quotes included. Having quotes allows us to quickly judge the accuracy of the results. A short snippet is seen below.

```
urn:wikidata:Q733791
|-Q201788:historiker
"var en finlandssvensk konsthistoriker"
|-Q1792450:konsthistoriker
"var en finlandssvensk konsthistoriker"
urn:wikidata:Q937
|-Q169470:fysiker
```

```
"var en tysk-judisk teoretisk fysiker"
urn:wikidata:Q5726883
|-Q350979:zoolog
"var en finländsk zoolog"
```

The second file contains a list of IDs that belong to people who did not have an article associated with them. This file should be empty since each person must have an article to be included by the Wikidata Searcher. If it is not empty then that means there is a mismatch between the copy of the Wikidata and Wikipedia. This can be caused by a multitude of reasons.

6.2 Data Storage

All the information that the searcher programs have extracted is stored as categorized input/output files for reference. We have also inserted it into tables in a database to have a source optimized for presentation and statistical queries.

The database consists of the following tables:

Professions: A list of all scientific professions with their Wikidata ID, English and Swedish title.

Keywords: A list of all scientific professions with a Swedish title.

Scientists: A list of all the people that have at least one scientific profession, containing their Wikidata ID and Swedish name.

Occupations: A relational table that connects the scientists with their profession.

Quotes: A table with all the extracted quotes from Wikipedia and the Wikidata ID of the scientist described.

6.3 Interface

The resulting data from our work is presented as a search engine, seen in Fig. 2. Users can type the names they have in mind into the search box and the website will not only provide an instant answer but also show you which tags and quotes that can verify it. From there you can navigate to the Wikipedia and Wikidata pages containing those claims. The professions that the searcher programs are looking for are listed on the website as well.

Our website is available at this address `www.vetenskapsman.com` along with instructions on how to use it (in Swedish). “Vetenskapsman” is Swedish for “scientist”.

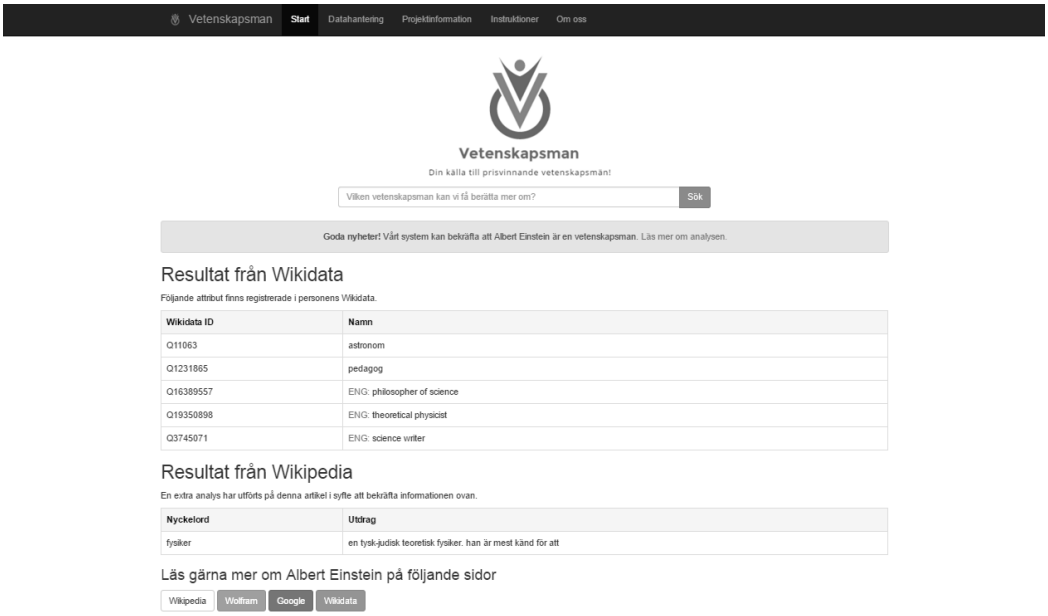

Figure 2: Web interface of vetenskapsman.com

7 Results

We evaluated the performance of our system using a small test sample of 90 people. They were selected by specifying our delimiters to be people born in March and April 1879 when running the Wikidata Searcher.

7.1 Evaluation

We computed the precision, recall, and F-score to evaluate the accuracy of our programs. They are calculated from knowing the number of true positives, true negatives, false positives, and false negatives. These numbers were acquired by manually checking each person in our test sample. Table 1 shows the results.

	Precision	Recall	F-Score
Wikidata Searcher	100	59.26	74.42
Wikipedia Searcher	95	70.37	80.85

Table 1: Performance figures

Results for the Wikidata Searcher are more dependent on the people maintaining the Wikidata than our program. The cause of this comes from how the program works. It merely fetches information, without any analysis of it, since the Wikidata is already sorted and has good structure. This means that the program will find what is there and will not find what is not there, for better or worse.

The Wikipedia Searcher which uses regular expressions to extract scientists from text had one false positive. The program saves a quote from each of its findings that can help us improve the analysis. The quote that was wrongfully interpreted is as follows:

> var son till ingenjör
> “was son to engineer”

which while conforming to the regular expression does not make the person a scientist.

7.2 Statistics

The combined knowledge from Wikidata and Wikipedia shows that 27 people were scientists and the remaining 63 were not. Neither program managed to find all 27 on its own. However, by adding the results together, we managed to achieve a total recall of 100%. In Fig. 3, we can see results from both programs compared against each other, with an overlap of 8 scientists.

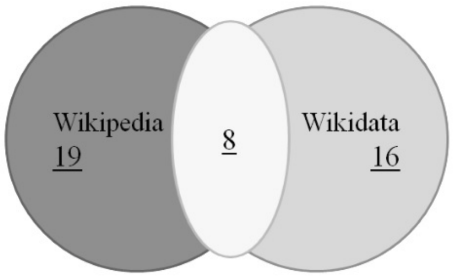

Figure 3: Results from Wikipedia Searcher compared to the Wikidata Searcher

7.3 Future Work

The results look very promising for the Wikipedia Searcher despite its simple regular expression analysis. They may however not be fully representative of the actual accuracy since they are based on a small test sample.

Wikipedia is one of the biggest sources of information and being able to gather data with a real purpose and achieve something useful has been important to us since the start of our project. One of the most interesting and alternative use cases is to use the extracted information to contribute to the Wikidata. The determining factor is that we successfully found many scientists who had their scientific professions missing in the Wikidata. Having built a database of scientists and being able to extract information also helps us build statistics such as how many scientists there are and how that has changed over time. The methods used could also be integrated into the Wikipedia servers to actively create Wikidata by interpreting written articles as well as evaluating the reliability of facts by counting available sources.

8 Conclusions

While the website may be fun to browse and the statistics can be interesting, the user base for such a specific creation is limited but it helped us prove the potential of our idea. With a relatively simple text analysis we managed to find a lot more people who according to our definition are scientists, than what would have been possible by just querying the Wikidata.

If we recall the statistics from our comparison between Wikipedia and Wikidata we can see that out of 27 scientists only 16 were tagged accordingly in the Wikidata. Furthermore, we also know that the Wikipedia Searcher has a precision rate of 95% and the combined precision of both programs is thus 96.43%. What the results told us is that it is possible to increase the number of correctly identified scientists in the Wikidata by almost 70% with a miss rate of about 30% from the Wikipedia Searcher and this is if we exclude the fact that a more advanced analysis could have possibly caught the false negatives that we had with it.

Looking back at the entire project it would be fair to say that the results are satisfying and also better than we could have hoped for considering the relatively simple method that was used and limited time frame. We do see potential for even better results if we were to process more data using a larger collection of keywords and more complex regular expressions and filters. On the other hand, we should have expected good results given the facts that most scientists are clearly described as scientists in their Wikipedia articles and the Wikidata is still new with a lot of work yet to be done. It will be interesting to see how much the Wikidata has improved in a few years from now.

Acknowledgments

We want to thank our instructor Pierre Nugues for his guidance during the project. We also want to thank Marcus Klang for help and tips regarding technical issues and for his work on the WikiParq project. Without the freely licensed data provided by the Wikimedia Foundation and the hard work put into the WikiParq project to format it, our project would not have been possible, we are truly grateful for this. Lastly we want to mention our appreciation to a few developers for various open source frameworks like jQuery, bootstrap and typeahead that proved useful in the development of the website, these include John Resig, Mark Otto, Jacob Thornton, Tim Trueman, Veljko Skarich and Jake Harding.

References

Kurt Bollacker, Patrick Tufts, Tomi Pierce, and Robert Cook. 2007. A platform for scalable, collaborative, structured information integration. In *Intl. Workshop on Information Integration on the Web (IIWeb'07)*.

Marcus Klang and Pierre Nugues. 2016. Wikiparq: A tabulated wikipedia resource using the parquet format. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference (LREC 2016)*, Portorož.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. 2014. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Meta. 2016. *List of Wikipedias — Meta, discussion about Wikimedia projects*. Meta, discussion about Wikimedia projects., https://meta.wikimedia.org/w/index.php?title=List_of_Wikipedias&oldid=15207561. [Online; accessed 12-January-2016].

Wikipedia. 2015. *Wikipedia:Modelling Wikipedia's growth — Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia., https://en.wikipedia.org/w/index.php?title=Wikipedia:Modelling_Wikipedia's_growth&oldid=685257398. [Online; accessed 12-January-2016].

Amy Zhao Yu, Shahar Ronen, Kevin Hu, Tiffany Lu, and César A. Hidalgo. 2016. Pantheon 1.0, a manually verified dataset of globally famous biographies. *Scientific Data*, 3, 1.