Extraction of lethal events from Wikipedia and a semantic repository

Norrby, Magnus; Nugues, Pierre

Published in:
Proceedings of the workshop on Semantic resources and semantic annotation for Natural Language Processing and the Digital Humanities at NODALIDA 2015

2015

Extraction of Lethal Events from Wikipedia and a Semantic Repository

Magnus Norrby
Lund University
Department of Computer Science
221 00 Lund, Sweden
mange.norrby@gmail.com

Pierre Nugues
Lund University
Department of Computer Science
221 00 Lund, Sweden
Pierre.Nugues@cs.lth.se

Abstract

This paper describes the extraction of information on lethal events from the Swedish version of Wikipedia. The information searched includes the persons' cause of death, origin, and profession. We carried out the extraction using a processing pipeline of available tools for Swedish including a part-of-speech tagger, a dependency parser, and manually-written extraction rules. We also extracted structured semantic data from the Wikidata store that we combined with the information retrieved from Wikipedia. Eventually, we gathered a database of facts that covers both sources: Wikipedia and Wikidata.

1 Introduction

Wikipedia contains a large number of biographies in many languages, where key events in the life of a person are described in plain text. As a complement to the Wikipedia narrative, Wikidata is a data repository that assigns Wikipedia entities unique identifiers across languages: A sort of social security number valid across all the Wikipedia territories.

Wikidata was primarily designed to ease the interconnection between multilingual web pages, but it now links entities with a growing number of semantic properties, such as for a person, a birth name (P1477), sex or gender (P21), etc.; for an administrative territorial entity (country), a capital (P36), a currency (P38), an official language (P37), etc. For instance, the entity *Jimi Hendrix* has the unique identifier Q5928 with properties such his date of birth, *27 November 1942*, and of death (P570): *18 September 1970*. The relatively language-agnostic structure of Wikidata makes it a very convenient semantic resource that can potentially be applied with no adaptation to any language version of Wikipedia.

This article explores means to extract information about lethal events from texts including the date and cause of death of people, using a relatively simple text processing architecture consisting of a part-of-speech tagger and a dependency parser and this large, language-independent, graph-oriented semantic repository.

We gathered the texts from the Swedish Wikipedia and the extracted data was then compared with data found in Wikidata. We stored the resulting extracted information in an SQL database. The collected data could then easily be queried to answer questions about the population on Wikipedia. A question we had in mind when we evaluated our database was how to answer the puzzling and much-debated coincidence:

> Do all the legendary musicians die at 27?

2 Previous Work

The work we describe in this paper is an information extraction task using the Swedish version of Wikipedia, Wikipedia categories, and a rich semantically annotated data set: Wikidata.

Information extraction systems are now ubiquitous and there are countless references that describe them. The Message Understanding Conferences (MUC) standardized their evaluation and the pipeline architecture eloquently advocated by the FASTUS system (Appelt et al., 1993; Hobbs et al., 1997) is still followed by scores of information extraction systems.

Many information extraction systems use now

Magnus Norrby and Pierre Nugues 2015. Extraction of lethal events from Wikipedia and a semantic repository. *Proceedings of the workshop on Semantic resources and semantic annotation for Natural Language Processing and the Digital Humanities at NODALIDA 2015*. NEALT Proceedings Series 27 / Linköping Electronic Conference Proceedings 112: 28–35.

the Wikipedia collection of articles as a source of information as it is freely available and, although disputed, of relatively good quality. It also provides semi-structured data which can complement the text.

Wu and Weld (2007) is an example of it, where the authors designed a system to extract values of predefined properties from the article text and complement infoboxes or to improve document categorization.

Lange et al. (2010) parsed the Wikipedia running text to extract data and populate the article's infoboxes. Their strategy is more complex than the one described in this paper since they handled more than 3,000 Wikipedia different template structures. In our case, we always looked for the same type of data and could therefore in many cases find the needed information in the page categories, instead of the free text.

Chernov et al. (2006) is another project that used the semantic information between the category links in an attempt to improve Wikipedia's search capabilities.

In this paper, we combined Wikipedia, the relatively new Wikidata semantic repository, and language-processing components for Swedish to extract information and store it in a fashion that is easier to search.

3 Dataset

We chose the Swedish Wikipedia as initial dataset that we downloaded from the Wikipedia dump pages. The dump consists of a compressed XML tree that we parsed with the Bliki XML Parser (Bliki, 2014) to produce HTML pages. We then used Sweble's Wikitext Parser (Dohrn and Riehle, 2013) to reduce the articles from HTML to raw text. From this corpus, we extracted a set of persons, their place of origin, their professions, dates of birth and death, place of death, and finally the cause of their death.

4 Wikipedia Categories

We first extracted a set of human entities from the whole collection. To carry this out, we used the Wikipedia category: *birth by year* "Födda" (`Category:Births_by_year`) that lists persons by their year of birth followed by the word *births*, i.e. "1967 births". A similar category lists persons by their year of death: *deaths by year* "Avlidna" (`Category:Deaths_by_year`), for instance "1994 deaths".

The latter category allowed us to narrow the search from all the persons to persons who passed away and thus where the pages possibly contained a lethal event. We applied regular expressions on the raw text to find these categories. We parsed all pages and we saved the pages where we found both categories. This way, we collected a dataset of 78,151 Swedish Wikipedia pages, all describing persons.

5 Extraction from Text

For each page, the Bliki XML Parser extracts both the page title, here a person's name, and the Wikipedia numeric page ID. If two persons have the same name, Wikipedia adds more information to make the title unique. For example, the Swedish Wikipedia lists nine different Magnus Nilsson. These pages are given titles like *Magnus Nilsson (kung)* "Magnus Nilsson (king)", *Magnus Nilsson (född 1983)* "Magnus Nilsson (born 1983)", etc. The added information is always contained within two parentheses and we extracted the name by splitting the string at the "(" character. To maintain a unique ID for each person when the names could be identical, we used the numeric page ID instead.

5.1 Parsing the Text

We built a processing pipeline consisting of a part-of-speech (POS) tagger and a dependency parser that we applied to the documents.

We first tagged all the documents with the Stagger POS tagger for Swedish (Östling, 2013). We then applied the MaltParser dependency parser (Nivre et al., 2006) on the tagged sentences.

We saved the results in a CoNLL-like format consisting of the following columns: token counter (ID), form, lemma, coarse POS tag, POS tag, grammatical features, head, and dependency relation. In addition, Stagger output named entity tags that supplement the CoNLL columns.

5.2 Date Extraction

We extracted the dates from their POS tags. A date like *11 January 1111* is tagged as RG NN RG, where RG represents a base number and NN, a noun. We searched this specific pattern in all the sentences and we checked that the noun belonged to one of the twelve months using string matching.

We cross-checked these dates with the categories from Sect. 4 corresponding to the year of birth and the year of death and we saved them when they agreed.

This method could certainly be improved as it does not take into account that these dates could describe other events. To reduce the risk of saving a wrong date, we compared them internally. If more than one date was found on the same birth year, we chose the earliest and we applied the reverse for death years. This assumes that all dates referred to dates while the person was alive which, of course, is a simplification.

5.3 Extraction from Categories

The categories of Wikipedia pages are simple phrases that state facts about the page such as the person's profession or cause of death. The key difference between the page categories and free text is that the categories have a specific pattern and connect other pages with the same categories. This means that we can add common categories that describe information we want to extract to our search pattern.

Since the page categories are created and added by users, the categorization sometimes contains mistakes, while some information is omitted. A page describing a guitarist might lack the category *Guitarists* but contain the tag *Left-handed guitarists*. This means that we cannot solely depend on the string matching, but also apply our tagger to the categories. Fortunately as noted by Nastase and Strube (2008), the analysis of these short phrases are much easier than for free text or even simple sentences.

5.3.1 Causes of Death

We extracted the causes of death from the page categories. We collected manually these categories through the examination of Wikipedia pages. We then used string matching to extract the causes of death of all the persons we had in our collection.

Although relatively consistent, same kinds of events can be assigned different categories. Assassinations and murders commonly use the category *Personer som blivit mördade* in Swedish, literally *Persons that have been murdered*, that corresponds to the English category *Murder victims*. However, Martin Luther King is assigned another category instead: *Mördade amerikanska politiker* equivalent to *Assassinated American politicians* in English. The string patterns used to extract the causes of death are shown below while Table 1 shows their equivalent categories in the English Wikipedia.

- *Personer som blivit mördade*, English Wikipedia: *Murder victims* "Persons that have been murdered"
- *Mördade* "Murdered"
- *Personer som begått självmord* "Suicides"
- *Personer som drunknat* "Persons that have drowned"
- *Personer som blivit avrättade* "Persons that have been executed"
- *Personer som avrättades* "Persons that were executed"
- *Personer som stupat i strid* "Persons who died in battle"

Swedish category	Equivalent English category
Personer som blivit mördade	Murder victims
Personer som begått självmord	Suicides
Personer som drunknat	Deaths by drowning
Personer som blivit avrättade	Executed people
Personer som stupat i strid	Military personnel killed in action

Table 1: English equivalent to categories used to extract the causes of death.

Some categories were not as straightforward as with the phrase *Personer som dött av...* "Persons who died of..." that appeared in many different categories such as *Personer som dött av idrottsolyckor* "Persons who died of sport injuries". Since we knew that the categories only contained one sentence we could just strip the part *Personer som dött av* and we saved the remainder as the cause.

5.3.2 Places of Origin

We limited the places of origin to be either a country, a town, or another geographical region. A person was allowed to have multiple origins and

we did not make a difference between the different origin types. Because of the recurring syntax of the categories, we used simple patterns. The first approach was to find categories containing the string *från* "from". This method captured all the categories with the syntax *Person från ORIGIN* "Person from ORIGIN".

We used the strings *Musiker i* "Musicians in" as with *Musiker i Sverige under 1600-talet* "Musicians in Sweden during the 15th century". We chose these strings because they are relatively frequent in the Swedish Wikipedia. We used a similar approach with categories containing the string *Personer i* "Persons in" to match the pattern *Personer i ORIGIN TAIL*, where the tail could be anything.

This method found a lot of false positives as for instance *Personer i grekisk mytologi* "Persons in Greek mythology". Most of the false positives could be removed by checking if the word marked as origin began with a capital letter. However, this approach would not work well in English as can be seen in the example above.

5.4 Extractions using Dependency Parsing

In addition to the patterns applied to the categories, we analyzed the dependency parse trees from the text to extract further information.

5.4.1 Places of Origin

To complement the places of origin obtained from the categories, we searched for paths in the trees linking the name of the person to a word tagged as a place by Stagger. These paths had to go through the verb *föda* "bear", or its inflections. We added the relations we found to the database.

5.4.2 Details on the Death

To add details on the death, we searched the words *dödsorsak* "cause of death" or *avled* "passed away" and we examined their modifiers. Causes of death often involved the grammatical function tag PA corresponding to the complement of preposition. We used this function in combination with a noun (NN) to detect the cause. We applied additional rules to capture the specifics of the whole cause. If for instance, the word after the identified cause was *på* "on" that word and the word after was also added to the cause. This made sure that we handled causes like *ruptur på aortan* "Aortic aneurysm" correctly.

6 Extraction from a Semantic Repository

6.1 Wikidata

Wikidata is a database of Wikipedia entities. It started as a means to provide a better linking mechanism between the different language versions of the Wikipedia pages. Each entity was assigned a unique identifier in the form of a Q-number. For instance, the Swedish writer Astrid Lindgren has the number Q55767, Gustave Flaubert has the number Q43444, and Denmark has the number: Q35.

Entities with the same name as Casablanca receive different numbers: Q7903 for the Moroccan city and Q132689 for the American movie. Using the address: http://www.Wikidata.org/wiki/Q55767, it is possible to access all the versions of the Astrid Lindgren pages: 72 pages in total.

In addition to the unique number identification, Wikidata links the entity to properties, for instance a sex and gender, P21, a place of birth, P19, a country of citizenship, P27, a supercategory (instance of), P31, etc. For Astrid Lindgren, the corresponding key-value pairs are:

- P21 = female (Q6581072)
- P19 = Vimmerby (Q634231)
- P27 = Sweden (Q34)
- P31 = human (Q5)
- etc.

There are now hundreds of properties that are stored in the form of RDF triples and the list is growing.

As with Wikipedia, we parsed Wikidata with the Bliki XML parser and we stored the results in the JSON format.

6.2 Identifying Persons

We parsed all the Wikidata pages to identify persons using the JSON Simple parser (Fang, 2012). We extracted them by considering the property *instance of*, P31, and the value *human* (Q5).

We did not try to find new persons, but only to add information to persons already identified from Wikipedia. The next step was therefore to find the name of the person described and see if it matched any name we already had. This was done by looking at the JSON entity labels. If it

contained the entity "sv", which marked the existence of a Swedish page, we saved its value as a string. If that string matched any of our saved persons name from Wikipedia, we defined them as the same person and continued to parse the page.

6.3 Extraction

The extraction was fairly straightforward from of the Wikidata properties. We used:

- Place of birth (P19)
- Place of death (P20)
- Occupation (P106)
- Cause of death (P509)
- Date of birth (P569)
- Date of death (P570)
- Manner of death (P1196)

Wikidata makes a difference between *cause of death* and *manner of death*, which we did not do while searching Wikipedia. If we found both for a person, we used the manner of death.

We merged these properties with those we extracted from Wikipedia.

7 Combining Results

Table 2 shows the number of extractions from the Wikipedia text and from Wikidata, where the combined column is the combination of unique values. Table 3 shows the number of unique and shared values.

We can assume that we retrieved all the data available from Wikidata. This gives us an idea of the accuracy of the Wikipedia extractions. As the tables show, we found very few causes of death. The numbers might look pretty weak at a first glance but when they are compared to Wikidata, we see that the information was very rarely present.

The data set coming from Wikidata is smaller since we only accepted persons that we could link to those previously extracted from Wikipedia. This means that we cannot directly compare the absolute values from Wikidata and Wikipedia. Table 4 shows the extraction rates compared to the data set size.

If we use Wikidata as a baseline, we can see that our Wikipedia extractions performs well in both professions and origins. The high numbers in professions are due to the fact that many persons have more than one. As an example, Wikidata lists six professions for the artist Kurt Cobain.

	Wikipedia	Wikidata	Both
Persons	78,151	61,410	78,151
Origins	47,174	36,268	75,341
Professions	95,792	69,429	140,654
Place of death	34,909	35,166	52,545
Birth date	54,188	52,052	73,702
Death date	53,606	52,299	73,833
Cause of death	2,198	4,161	5,821

Table 2: Extraction numbers

	Wikipedia	Wikidata	Shared
Origins	39,073	28,167	8,101
Professions	71,225	44,862	24,567
Place of death	17,379	17,636	17,530
Birth date	21,650	19,514	32,538
Death date	21,534	20,227	32,072
Cause of death	1,660	3,623	538

Table 3: Unique and shared extractions

We also see that a perfect extractor would possibly find about 100% more causes of death on Wikipedia than we did in this paper. A large improvement could come from using more relations in dependency parsing and adding more search patterns to the extractor algorithm.

8 Analysis with a Database

We stored the resulting data in a SQL database with three tables: persons, origins, and profes-

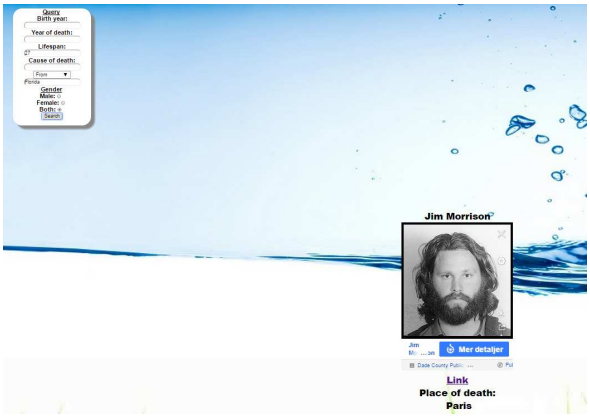

Figure 1: Screenshot of the interface

	Wikipedia	Wikidata	Both
Persons	–	79%	–
Origins	60%	59%	96%
Professions	123%	113%	180%
Place of death	45%	57%	67%
Birth date	69%	85%	94%
Death date	69%	85%	94%
Cause of death	3%	6.5%	7.5%

Table 4: Extraction ratios

sions. This separation was done since a person could have multiple values for both origin and profession.

We built a web application that we linked to the database and that enables a distant interaction with the system. A user can then query the system using a form and the results are presented as basic information about persons who matched the criteria with possible images and links to their Wikipedia page. Figure 1 shows an example of it.

The database can easily be queried to answer questions about our data. Table 5 shows, for instance, the most common causes of death.

Rank	Cause of death	Number of cases
1	Heart attack	885
2	Suicide	572
3	Execution	571
4	Stroke	333
5	Tuberculosis	296

Table 5: Top five causes of death

As discussed previously, the scores for professions and origins were high since persons could have multiple entries. With SQL, we could quickly check how many persons that we could connect to a profession and an origin. The result was 69,077 persons with one or more professions and 55,922 persons with one or more origins.

We also counted the number of unique property names and Table 6 shows the result.

Property	Number of unique cases
Cause of death	166
Profession	1,337
Origin	14,000
Place of death	10,106

Table 6: Unique properties

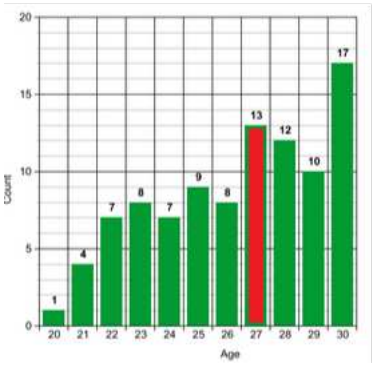

Figure 2: The death rate for musicians in the age range 20-30.

8.1 Evaluation of the Results on *Club 27*

Finally, to answer the initial question on the existence of a *Club 27*, the extractor produced a total number of 2,110 singers and musicians including many of the famous Club 27 members as Jimi Hendrix and Jim Morrison. Figure 2 shows the death count by age and this data could not support the existence of the Club 27 and, even if there is a spike at 27, more musicians died at 30...

The list of extracted musicians and singers in Table 7 can be compared with that in a page dedicated to Club 27 on the Swedish Wikipedia (Wikipedia, 2015). The latter contains 33 names, 34 with Jim Morrison, who was left out from the list, but mentioned in the text. Out of these 34 names, 19 did not have a Wikipedia page and were therefore ignored by our extractor. Out of the remaining 15 members, 11 were present in our list in Table 7. The four people our system did not find were:

- Alexandre Levy, pianist and composer
- Gary Thain, bassist
- Chris Bell, no dates of birth and death
- Richey James Edwards

Alexandre Levy existed in our database and was labeled as a pianist and composer. Gary Thain also existed with the profession of bassist. Both could have been included if we had broadened the definition of a musician to include the corresponding categories.

Chris Bell had neither birth or death date in his Swedish Wikipedia page and was thereby not included in our database.

Name	Gender	Born	Deceased	Death place
Jim Morrison	Male	1943-12-08	1971-07-03	Paris
Jimi Hendrix	Male	1942-11-27	1970-09-18	London
Kurt Cobain	Male	1967-02-20	1994-04-05	Seattle
Brian Jones	Male	1942-02-28	1969-07-03	Hartfield
Janis Joplin	Female	1943-01-19	1970-10-04	Los Angeles
Kristen Pfaff	Female	1967-05-26	1994-06-16	Seattle
Alan Wilson	Male	1943-07-04	1970-09-03	Topanga
Amy Winehouse	Female	1983-09-14	2011-07-23	London
Soledad Miranda	Female	1943-07-09	1970-08-18	Lisbon
Jeremy Michael Ward	Male	1976-05-05	2003-05-25	Los Angeles
Oskar Hoddø	Male	1916-01-01	1943-11-17	
Mia Zapata	Female	1965-08-25	1993-07-07	Seattle
Ron McKernan	Male	1945-09-08	1973-03-08	

Table 7: The members of Club 27 according to our extractor

Richey James Edwards on the other hand was marked as a musician, but was assigned a wrong year of death by the extractor. When analyzing his Wikipedia page, we could find the cause of this incorrect date: He was reported missing on February 1st, 1995 at 27 years old but his body was never found. He pronounced dead, much later, on November 23rd, 2008, which was the date the extractor collected and stored in the database.

We also found the two musicians, Soledad Miranda and Oskar Hoddø, that died at 27, but were not part of Wikipedia's list of Club 27.

9 Conclusion

In this paper, we described a system to combine data from a language-dependent information extraction system and a large semantic repository: Wikidata. Although the scope of the evaluation was limited, we showed that neither the extraction component, nor the repository could identify the complete set of facts we had in mind and that the interplay of both components significantly improved the final results.

We found that Wikidata was easy to integrate with a language-dependent pipeline and we believe that it has the potential to serve as a core semantic resource in many other information extraction applications.

Acknowledgments

This research was supported by Vetenskapsrådet through the *Det digitaliserade samhället* program.

References

Douglas Appelt, Jerry Hobbs, John Bear, David Israel, Megumi Kameyama, and Mabry Tyson. 1993. SRI: Description of the JV-FASTUS system used for MUC-5. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland*, pages 221–235, San Francisco, August. Morgan Kaufmann.

Bliki. 2014. Bliki engine. bitbucket.org/axelclk/info.bliki.wiki/wiki/Home.

Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, and Xuan Zhou. 2006. Extracting semantic relationships between wikipedia categories. 1st International Workshop: SemWiki2006 – From Wiki to Semantics.

Hannes Dohrn and Dirk Riehle. 2013. Design and implementation of wiki content transformations and refactorings. In *Proceedings of the 9th International Symposium on Open Collaboration*, WikiSym '13, pages 2:1–2:10.

Yidong Fang. 2012. Json.simple. code.google.com/p/json-simple/.

Jerry R. Hobbs, Douglas E. Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1997. FASTUS: a cascaded finite-state transducer for extracting information from natural-language text. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, chapter 13, pages 383–406. MIT Press, Cambridge, Massachusetts.

Dustin Lange, Christoph Böhm, and Felix Naumann. 2010. Extracting structured information from wikipedia articles to populate infoboxes. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1661–1664.

Vivi Nastase and Michael Strube. 2008. Decoding wikipedia categories for knowledge acquisition. In *AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence*, volume 2, pages 1219–1224.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*.

Robert Östling. 2013. Stagger: an open-source part of speech tagger for Swedish. *Northern European Journal of Language Technology*, 3.

Wikipedia. 2015. Club 27. `http://sv.wikipedia.org/wiki/27_Club#Musiker_som_avlidit_vid_27_.C3.A5rs_.C3.A5lder`. Accessed March 11, 2015.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 41–50.