Creating a Dead Poets Society: Extracting a Social Network of Historical Persons from the Web

Gijs Geleijnse Jan Korst

Philips Research
High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
{gijs.geleijnse,jan.korst}@philips.com

Abstract. We present a simple method to extract information from search engine snippets. Although the techniques presented are domain independent, this work focuses on extracting biographical information of historical persons from multiple unstructured sources on the Web. We first similarly find a list of persons and their periods of life by querying the periods and scanning the retrieved snippets for person names. Subsequently, we find biographical information for the persons extracted. In order to get insight in the mutual relations among the persons identified, we create a social network using co-occurrences on the Web. Although we use uncontrolled and unstructured Web sources, the information extracted is reliable. Moreover we show that Web Information Extraction can be used to create both informative and enjoyable applications.

1 Introduction

Information extraction (IE) is the task of identifying instances and relations between those instances in a text corpus. Thanks to popular search engines, the Web is a well-indexed tera-size corpus, where precise queries often lead to highly relevant text fragments. When we assume that all knowledge available on a domain can be found on the Web, this corpus can be used for *ontology-driven information extraction* [1], i.e. we can assume that all required information is available within the web. Due to the redundancy of information on the web, the question is rather how to identify *at least one* formulation of every fact of interest (determined by the ontology), than how to extract *every* relevant formulation within the corpus (i.e. *corpus-driven* information extraction).

Pattern-based approaches have shown to be effective for ontology-driven information extraction. Such approaches can not only be used to identify lexicographical relations (e.g. hyponyms, part-of) [2, 3] but also various other relations. Patterns expressing relations (e.g. *'is the president of'*, [2]) or combined instance-pattern terms (e.g. *is the president of the United States*, [4, 1] are used as queries to a search engine. Such queries give access to highly relevant text fragments and instances (e.g. *George W. Bush*) and the expressed relations can be extracted simultaneously.

In this paper we show how web information extraction can indeed be a valuable addition to the current collective knowledge on the web. We present a method to populate an ontology using information extraction from search engine query results. Using the redundancy of information on the web, we can apply simple methods to extract term and relations. We illustrate this approach by populating an ontology with historical persons, their main biographical data and their 'degree of fame'. We also create a social network among the persons extracted. By combining and structuring information from a large collection of web pages, questions like *Who are the most famous novelists from Ireland?*, *Which people were born in 1648?*, *Who are popular female composers?* and *Who are considered to be most related to Vincent van Gogh?* can now be easily answered. This work illustrates that using simple techniques we can create a reliable ontology that gives insight in the domain and can be a true addition to the information available.

2 Related Work

KnowItAll is a hybrid named-entity extraction system [2] that finds lists of instances (e.g. 'Busan', *'Karlsruhe'*)of a given class (e.g. *'City'*) from the web using a search engine. It combines hyponym patterns [5] and class-specific, learned patterns to identify and extract named-entities. Moreover, it uses adaptive wrapper algorithms [6] to extract information from HTML markup such as tables. Contrary to the method used in this paper, the instances found are not used to create new queries. In [7] the information extracted by KnowItAll is evaluated using a combinatorial model based on the redundancy of information on the web.

Recently, the KnowItAll project has addressed the identification of complex named entities (such as book titles), using a statistical *n-gram* approach [8]. In [9, 4] such complex entities are recognized using a set of simple rules. For example, a movie title is recognized when it is placed between quotation marks.

Cimiano and Staab [10] describe a method to use a search engine to verify a hypothesis relation. For example, the number of hits to *"rivers such as the Nile"* and *"cities such as the Nile"*) are compared to determine the class of *Nile*. Per instance, the number of queries is linear in the number of classes considered.

Where Cimiano and Staab assume the relation patterns to be given, in [11] a technique is introduced to find precise patterns using a training set of related instances. In [4] a method is presented to extract information from the web using effective patterns, as precision is not the only criterion for a pattern to provide useful results.

The number of search engine *hits* for pairs of instances can be used to compute a semantic distance between the instances [12]. The nature of the relation is not identified, but the technique can for example be used to cluster related instances. In [13] a similar method is used to cluster artists using search engine counts. However, the total number of hits provided by the search engine is an estimate and not always reliable [14]. In [15] an approach is presented where one instance is queried and the resulting texts are mined for occurrences of other instances. Such an approach is not only more efficient in the number of queries, but also gives better results.

The extraction of social networks using web data is a frequently addressed topic. For example, Mori et al. [16] use *tf·idf* to identify relations between politicians and locations and [17] use inner-sentence co-occurrences of company names to identify a network of related companies.

3 Problem Description

Suppose that we are given an ontology O with n classes $c_1, ..., c_n$ and a set $\mathcal{R}$ of relation classes $R_{i,j}$ on classes c_i and c_j. For example, if *'Person'* and *'City'* are classes, then *'is born in(Person, City)'* is a relation class on these classes. The tuple *'(Napoleon Bonaparte, Ajaccio)'* may be an instance of *'is born in(Person, City)'* if *'Napoleon Bonaparte'* is an instance of *'Person'* and *'Ajaccio'* is an instance of *'City'*. We call a class *complete* if all instances are assumed to be given beforehand. For *empty* classes no instances are given.

Problem. Given the ontology O,
(1.) populate the incomplete classes with instances extracted from the web, and
(2.) populate the relation classes in $\mathcal{R}$ by identifying formulations of all relations $R_{i,j}$ between the instances of c_i and c_j. □

In this work we focus on the population of an ontology on historical persons. In the given ontology (cf. Figure 1) all classes but *Person* are complete, while *Person* is empty. The class *Period* contains all combinations of years that are likely to denote the periods of life of a historical person. For example *'1345 - 1397'* is an instance of *Period*. The class *Nationality* contains all names of countries in the world. We identify derivatives of country names as well and use them as synonyms (e.g. *American* for *United States*, *English* for *United Kingdom* and *Flemish* for *Belgium*). A hierarchy among the instances of *Nationality* is defined using the names of the continent, such that we can for example select a list of historical persons from Europe. Likewise, the instances of *Profession* reflect 88 professions. For the instances *male* and *female* we have added a list

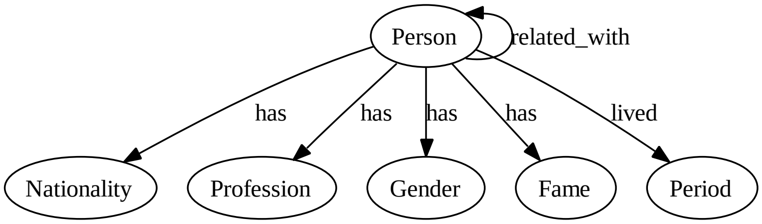

Fig. 1. The ontology on historical persons to be populated.

of derivatives to be used as synonyms, namely the terms *he*, *his*, *son of*, *brother of*, *father of*, *man* and *men* for *male* and the analogous words for *female*. We use the class *Fame* to rank the retrieved instances of *Person* according to their presence on the web. Hence the task is to identify a collection of biographies of historical persons and to identify a social network between the persons found. As persons may have more than one profession and can be related to multiple other people, we are interested in a ranked list of professions and related persons for each historical person found.

For efficiency reasons, we only extract information from the snippets returned by the search engine. Hence, we do not download full documents. As only a limited amount of automated queries per day are allowed[1], the approach should be efficient in the number of queries.

4 Finding Biographies

We use the given instances in the ontology O to populate the class *Person*, i.e. to find a list of names of historical persons. We use the instances available in the given ontology to formulate queries and hence create a corpus to extract information from.

Using a pattern-based approach, we combine natural language formulations of relations in O with known instances into queries. Such queries lead to highly relevant search results, as the snippets are expected to contain related instances. As search engines return only a limited amount of results and the same names may occur in multiple search results, we have to identify effective queries [4].

Suppose we use instances in the class *Profession* to extract the persons. When querying for the instance *'composer'*, it is likely that few well-known composers dominate the search results. As we are interested in a rich ontology of historical persons, this is thus a less-suited approach.

The class *Period* contains all combinations of years that are likely to denote the periods of life of a historical person. Hence, the number of instances known

[1] Yahoo! accepts 5,000 queries per day.

for the class *Period* is by far the largest for all complete classes in O. As it is unlikely that many important historical persons share both a date of birth and a date of death, the use of this class is best suited to obtain a long and diverse list of persons. The names of historical persons are often preceded in texts with a period in years (e.g. *'Vincent van Gogh (1853 - 1890)'*). As this period is likely to denote the period he or she lived in, we choose the pattern "(year of birth – year of death)" to collect snippets to identify the names of historical persons.

4.1 Identifying person names in snippets

Having obtained a collection of snippets, the next problem is to extract instances from the texts, in this case persons names. We choose to identify the names within the snippets using a rule-based approach. Although the design of rules is laborious, we do not opt for an approach based on machine learning (e.g. [18, 19, 8]) for the following reasons.

1. No representative training set is available. The corpus – the collection of snippets – contains broken sentences and improper formulations. Moreover, the corpus is multi-lingual.
2. The Named Entity Recognition (NER) task is simplified by the use of the patterns. Since we expect person names preceding the period queried, we know the placeholder of the named entity (i.e. preceding the queried expression). In a general NER task this information is not available.
3. Since the corpus consists of uncontrolled texts, there is need for post-processing. The snippets may contain typos, missing first or last names, and errors in dates and names.

We hence use a rule-based approach to identify person names in the snippets found with the periods.

First we extract all terms directly preceding the queried expressing that match a regular expression. That is, we extract terms of two or three capitalized words and compensate for initials, inversions (e.g. *'Bach, Johann Sebastian'*), middle names, Scottish names (e.g. *McCloud*) and the like.

Subsequently, we remove extracted terms that contain a word in a tabu list (e.g. *'Biography'*) and names that only occur once within the snippets. Having filtered out a set of potential names of persons, we use a string matching among the extracted names to remove typos and names extracted for the wrong period.

Using the 80,665 periods identified, we obtain a list of 28,862 terms to be added as instance to the class *Person*. Simultaneously, we extract the relations between the periods queried and the extracted instances.

Napoleon Bonaparte was the greatest military genius of the 19th century

Napoleon Bonaparte was born of lower noble status in Ajaccio, Corsica on August 15, 1769

Napoleon Bonaparte was effectively dictator of France beginning in 1799 and

Napoleon Bonaparte was the emperor of France in the early 1800s

Napoleon Bonaparte was a bully, rude and insulting

Napoleon Bonaparte was in Egypt and was not enjoying his tour

Napoleon Bonaparte was a great warrior and a notable conqueror

Napoleon Bonaparte was born on August 15, 1769 to Carlo and Letizia Bonaparte

Napoleon Bonaparte was defeated at Waterloo

Table 1. Example search results for the query 'Napoleon Bonaparte was'.

In the evaluation section we analyze the quality of the extracted instances and compare the rule-based approach with a state-of-the-art named entity recognizer based on machine learning.

4.2 Using Mined Names to find additional biographical information

Having found a list of instances of the class *Person*, we first determine a ranking of the instances extracted.

Finding a Rank. To present the extracted information in an entertaining manner, we determined the number of hits for each identified person. As names are not always unique descriptors, we queried for the combination of the last name and period (e.g. *'Rubens (1577 - 1640)'*). Although the number of hits returned a search engine is an estimate and irregularities may occur [14], we consider this simple and efficient technique to be well suited for this purpose.

Now we use the names of these instances in a similar fashion to acquire biographical information for the 10,000 best ranked persons. To limit the number of queries per instance, we select the pattern 'was' to reflect the relation between *Person* on the one hand and *Nationality*, *Gender* and *Profession* on the other hand. By querying phrases such as *'Napoleon Bonaparte was'* we thus expect to acquire sentences containing the biographical information. Table 1 contains examples of the sentences used to determine biographical information. We scan these sentences for occurrences of the instances (and their synonyms) of the related classes.

Relating persons to a gender. We simply counted instances and their synonyms within the snippets that refer to the gender *'male'* the opposite words that refer *'female'*. We simply related each instance of *Person* to the gender with the highest count.

Relating persons to a nationality. We assigned the nationality with the highest count.

Relating persons to professions. For each person, we assigned the profession p that most frequently occurred within the snippets retrieved. Moreover, as persons may have multiple professions, all other professions with a count at least half of the count of p were added.

Hence, using one query per instance of *Person*, we identify basic biographical information.

5 Identifying a Social Network

Having gathered a list of historical persons with biographical information, we are interested how the persons in the list are perceived to be related. Obviously such information can be extracted from the biographies, e.g. persons can be considered related when they share a profession, have the same nationality or lived in the same period.

However, we are interested in the way people nowadays relate the historical people extracted. For example, we are interested to identify the person who is considered to be most related to Winston Churchill. We therefore mine the Web for a social network of people extracted using the method in the previous section.

We assume that two persons are related when they are often mentioned in the same context. In early work, search engine queries were used that contained both the terms (e.g. [10, 12]). The number of Google hits to such queries were used as co-occurrence count co(u, v). However, work in [15] has shown that a pattern-based approach is not only more efficient in the number of queries, but also gives better results. Hence, we opt for an approach similar as described in the previous section, where we use patterns to identify relatedness between persons. We use the total numbers of co-occurrences co(p, q) for persons $p \neq q$ to compute the relatedness $T(p, q)$ of p to q. Using the hypothesis that enumerated items are often related, we use patterns expressing enumerations (Table 2) to obtain the snippets.

For each historical person p we could consider the person q with the highest co(p, q) to be the most related to p. However, we observe that, in that case, frequently occurring persons have a relatively large probability to be related to any person. This observation leads to a normalized approach.

$$T(a, b) = \frac{\text{co}(a, b)}{\sum_{c, c \neq b} \text{co}(c, b)} \qquad (1)$$

For each of the best 3,000 ranked persons, we computed a ranked list of most related persons in the large set of the 10,000 persons with biographies.

”like [person] and [person]”
”such as [person] and [person]”
”including [person] and [person]”
”for example [person] and [person]”
”namely [person] and [person]”
”[person] and [person]”
”[person] [person] and other”

Table 2. Patterns used to find co-occurrences within the search engine snippets.

6 Experimental Results

In this section we discuss the results of the ontology population method applied to the ontology on historical persons. We present examples of the extracted data to give the reader an impression of the results. Moreover, we show that structured data can be used to gain insights. Section 6.1 handles the extracted instances of *Person* and the identified biographical relations, while Section 6.2 handles the social network identified of the extracted persons. As no prior work is known in this domain, we cannot compare the results with others.

6.1 Evaluating the identified biographical information

The rank assigned to each of the persons in the list provides a mechanism to present the extracted data in an attractive manner. Table 3 gives the list of the 25 best ranked persons and the identified biographical information. Using the criterion defined in Section 4, Johann Sebastian Bach is thus the best known historical figure.

As the data is structured, we can also perform queries to select subsets of the full ranked list of persons. For example, we can create a list of best ranked artists (Table 4), or a ‘society’ of poets (Table 5). We note that Frédéric Chopin is often referred to as ’the poet of the piano’. Table 6 shows that Vincent van Gogh is the best ranked Dutch painter.

The reader can verify that the given list of extracted persons are highly accurate. However, lacking a benchmark set of *the* best known historical persons, we manually evaluated samples of the extracted ontology to estimate precision and recall.

Precision. To estimate the precision of the class *Person*, we selected three decennia, namely 1220-1229, 1550-1559 and 1880-1889, and analyzed for each the candidate persons that were found to be born in this decennium. For the first two decennia we analyzed the complete list, for decennium 1880-1889 we analyzed only the first 1000 as well as the last 1000 names. This resulted in a precision of 0.94, 0.95, and 0.98, respectively. As the decennium of 1880-1889 resulted in considerably more names, we take a weighted average of these results. This yields an estimated precision for the complete list of 0.98.

Johann Sebastian Bach (1685-1750)	Germany	composer,organist
Wolfgang Amadeus Mozart (1756-1791)	Austria	composer,musician
Ludwig van Beethoven (1770-1827)	Germany	composer
Albert Einstein (1879-1955)	Germany	scientist,physicist
Franz Schubert (1797-1828)	Austria	composer
Johannes Brahms (1833-1897)	Germany	composer
William Shakespeare (1564-1616)	United Kingdom	author,poet
Joseph Haydn (1732-1809)	Austria	composer
Johann Wolfgang Goethe (1749-1832)	Germany	philosopher,director,poet..
Charles Darwin (1809-1882)	United Kingdom	naturalist
Robert Schumann (1810-1856)	Germany	composer
Leonardo da Vinci (1452-1519)	Italy	artist,scientist,inventor
Giuseppe Verdi (1813-1901)	Italy	composer
Frederic Chopin (1810-1849)	Poland	composer,pianist,poet
Antonio Vivaldi (1678-1741)	Italy	composer
Richard Wagner (1813-1883)	Germany	composer
Ronald Reagan (1911-2004)	United States	president
Franz Liszt (1811-1886)	Hungary	pianist,composer
Claude Debussy (1862-1918)	France	composer
Henry Purcell (1659-1695)	United Kingdom	composer
Immanuel Kant (1724-1804)	Germany	philosopher
James Joyce (1882-1941)	Ireland	author
Friedrich Schiller (1759-1805)	Germany	poet,dramatist
Georg Philipp Telemann (1681-1767)	Germany	composer
Antonin Dvorak (1841-1904)	Czech Republic	composer

Table 3. The 25 historical persons with the highest rank.

We compare the precision of the rule-based approach with a state-of-the-art machine-learning-based algorithm, the Stanford Named Entity Recognizer (SNER [20]), trained on the CoNLL 2003 English training data. Focussing on persons born in the year 1882, using the rule-based approach we extracted 1,211 terms. SNER identified 24,652 unique terms as person names in the same snippets. When we apply the same post-processing on SNER extracted data (i.e. removing typos by string matching, single-word names and names extracted for different periods), 2,760 terms remain, of which 842 overlap with the terms extracted using the rule-based approach.

We manually inspected each of these 2,760 terms, resulting in a precision of only 62%. Around half of the correctly extracted names are not recognized by the rule-based approach, most of them due to the fact that these names did not directly preceded the queried period.

To estimate the precision of the extracted biographical relations, we inspected randomly selected sublists of the top 2500 persons. When we focus on the best scoring professions for the 2500 persons, we estimate the precision of this relation to be 96%. We did not encounter erroneously assigned genders, while we found 98% of the cases the right *Nationality*, if one is found.

Leonardo da Vinci (1452 - 1519)	Italy	artist, scientist,...
Pablo Picasso (1881 - 1973)	Spain	artist
Vincent van Gogh (1853 - 1890)	Netherlands	artist, painter
Claude Monet (1840 - 1926)	France	artist, painter,...
Pierre-Auguste Renoir (1841 - 1919)	France	painter
Paul Gauguin (1848 - 1903)	France	painter
Edgar Degas (1834 - 1917)	France	artist, painter,...
Paul Cezanne (1839 - 1906)	France	painter, artist
Salvador Dali (1904 - 1989)	Spain	artist
Henri Michaux (1899 - 1984)	Belgium	artist, poet
Gustav Klimt (1862 - 1918)	Austria	painter, artist
Peter Paul Rubens (1577 - 1640)	Belgium	artist, painter
Katsushika Hokusai (1760 - 1849)	Japan	painter
Amedeo Modigliani (1884 - 1920)	Italy	artist, painter
JMW Turner (1775 - 1851)	United Kingdom	artist, painter
James Mcneill Whistler (1834 - 1903)	United States	artist
Rene Magritte (1898 - 1967)	Belgium	artist, painter
Henri Matisse (1869 - 1954)	France	artist
Rembrandt van Rijn (1606 - 1669)	Netherlands	artist, painter
Edouard Manet (1832 - 1883)	France	artist, painter
Herm Albright (1876 - 1944)	-	artist, engraver,...
Marc Chagall (1887 - 1985)	Russia	painter, artist
Edvard Munch (1863 - 1944)	Norway	painter, artist
Wassily Kandinsky (1866 - 1944)	Russia	artist, painter
Francisco Goya (1746 - 1828)	Spain	artist, painter

Table 4. The 25 artists with the highest rank.

Hence, we conclude that the ontology populated using the rule-based approach is precise.

Recall. We estimate the recall of the instances found for *Person* by choosing a diverse set of six books containing short biographies of historical persons. Of the 1049 persons named in the books, 1033 were present in our list, which gives a recall of 0.98. For further details on the chosen books we refer to [21].

From Wikipedia, we extracted a list of important 1882-born people[2]. The list contains 44 persons. Of these 44 persons, 34 are indeed mentioned in the Google snippets found with the queried patterns. Using the rule-based approach, we identified 24 of these persons within the snippets. The other ones were only mentioned once (and hence not recognized) or found in different places in the snippets, i.e. not directly preceding the queried period. Using SNER, we identified 27 persons from the Wikipedia list.

For the recall of the identified biographical relations, we observe that for the 10,000 persons that we considered all were given a gender, 77% were given a nationality, and 95% were given one or more professions.

[2] http://en.wikipedia.org/wiki/1882, visited January 2007

William Shakespeare (1564-1616)	United Kingdom	author,poet
Johann Wolfgang Goethe (1749-1832)	Germany	poet, psychologist, philosopher..
Frederic Chopin (1810-1849)	Poland	composer,pianist,poet
Friedrich Schiller (1759-1805)	Germany	poet,dramatist
Oscar Wilde (1854-1900)	Ireland	author,poet
Jorge Luis Borges (1899-1986)	Argentina	author,poet
Victor Hugo (1802-1885)	France	author,poet,novelist
Ralph Waldo Emerson (1803-1882)	United States	poet,philosopher,author
William Blake (1757-1827)	United Kingdom	poet
Dante Alighieri (1265-1321)	Italy	poet
Robert Frost (1874-1963)	United States	poet
Heinrich Heine (1797-1856)	Germany	poet
Robert Louis Stevenson (1850-1894)	Samoa	engineer,author,poet
Alexander Pope (1688-1744)	United Kingdom	poet
Hildegard von Bingen (1098-1179)	Germany	composer,scientist,poet
Lord Byron (1788-1824)	Greece	poet
John Donne (1572-1631)	United Kingdom	poet,author
Henri Michaux (1899-1984)	Belgium	poet
Walt Whitman (1819-1892)	United States	poet
Robert Burns (1759-1796)	United Kingdom	poet

Table 5. The 20 best ranked poets.

Vincent van Gogh (1853-1890)	Kees van Dongen (1877-1968)
Rembrandt van Rijn (1606-1669)	Willem de Kooning (1904-1997)
Johannes Vermeer (1632-1675)	Pieter de Hooch (1629-1684)
Piet Mondrian (1872-1944)	Jan Steen (1626-1679)
Carel Fabritius (1622-1654)	Adriaen van Ostade (1610-1685)

Table 6. The 10 best ranked painters from the Netherlands.

6.2 Evaluating the social network

Aiming for a reflection of the collective knowledge of web contributors on historical figures, the extracted social network of historical persons is not a verifiable collection of facts. We illustrate the social network extracted by two examples. Figure 3 depicts the relatedness among the best ranked persons. An arrow from person p to q is drawn if q is among the 20 nearest neighbors of p. Using the same criterion, Figure 4 depicts the relatedness among the best ranked authors.

We are able to verify the precision of the relatedness between historical persons if we make the following assumptions. We consider two persons to be related if either (1.) they lived in the same period, i.e. there is an overlap in the periods the two lived, (2.) they shared a profession, (3.) they shared a nationality, or (4.) they are both female.

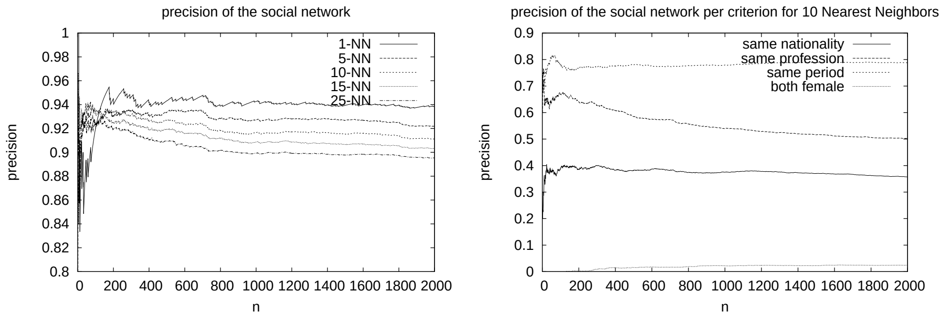

Fig. 2. (l.) Precision for the social network for the n best ranked persons and their k nearest neighbors. (r.) Precision for 10-NN per criterion.

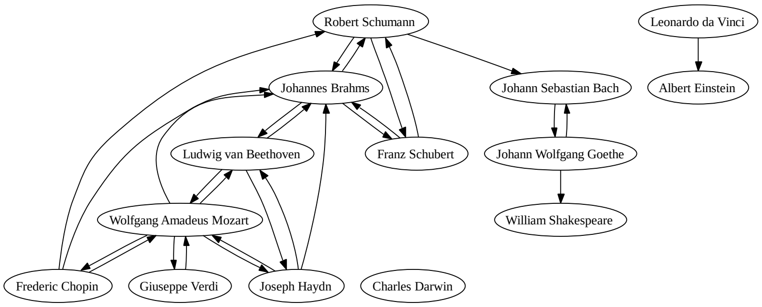

Fig. 3. The extracted social network for the 15 most best ranked persons.

Of course we cannot evaluate recall of the algorithm on these criteria, as for example not all persons sharing a nationality need to be consider to be related. We therefore the evaluate precision of the social network on these “minimal criteria”. For the 3,000 best ranked persons, we select the k most related persons. Per pair we evaluate whether either one of the four criteria is being met. This precision rate is presented in Figure 2 (l.). In comparison, the probability of any of the 3,000 persons to be related to a person in the large list of 10,000 is 45%. The precision rates for the social network per criterion can be found in Figure 2 (r.). The probabilities for two randomly selected persons to share a period, profession and nationality are 38%, 7.5% and 6.5% respectively. The chance that two historical persons are both female is only 0.5%. We hence conclude that these results give good confidence on the quality of extracted social network.

7 Conclusions

We illustrated a simple method to populate an ontology using a web search engine by finding biographical information on historical persons. Starting with

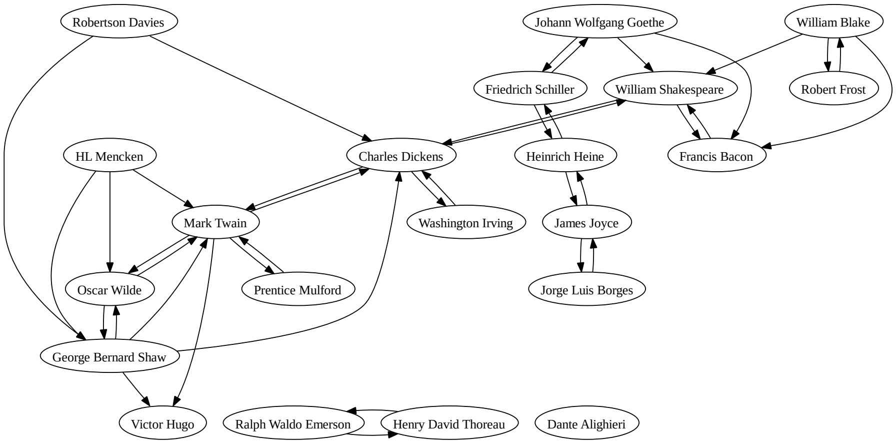

Fig. 4. The extracted social network for the best ranked authors.

the empty class *Person*, we found over 28 thousand historical persons with a high precision rate. The biographical information identified for the best ranked persons has shown to be of high quality as well. The same method is used to create a social network for the historical persons, with convincing results.

Hence, we show that simple Web Information Extraction techniques can be used to precisely populate ontologies. By combining and structuring information from the Web, we create a valuable surplus to the knowledge already available.

In future work, we plan to further address the automatic identification of characterizations for other items such as movies and musical artists. The identification of collective knowledge and opinions is perhaps more interesting than collecting plain facts, which often can be mined from semi-structured sources. By combining information extracted from multiple web pages, we plan to research methods to automatically tag items.

References

1. McDowell, L., Cafarella, M.J.: Ontology-driven information extraction with ontosyphon. In: Proceedings of the 5th International Semantic Web Conference (ISWC 2006). Volume 4273 of LNCS., Athens, GA, Springer (2006) 428 – 444
2. Etzioni, O., Cafarella, M.J., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence **165**(1) (2005) 91 – 134
3. van Hage, W.R., Kolb, H., Schreiber, G.: A method for learning part-whole relations. In: Proceedings of the 5th International Semantic Web Conference (ISWC 2006). Volume 4273 of LNCS., Athens, GA, Springer (2006) 723 – 736

4. Geleijnse, G., Korst, J.: Learning effective surface text patterns for information extraction. In: Proceedings of the EACL 2006 workshop on Adaptive Text Extraction and Mining (ATEM 2006), Trento, Italy (2006) 1 – 8
5. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th conference on Computational linguistics, Nantes, France (1992) 539 – 545
6. Crescenzi, V., Mecca, G.: Automatic information extraction from large websites. Journal of the ACM **51**(5) (2004) 731 – 779
7. Downey, D., Etzioni, O., Soderland, S.: A probabilistic model of redundancy in information extraction. In: 19th International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, UK (2005) 1034 – 1041
8. Downey, D., Broadhead, M., Etzioni, O.: Locating Complex Named Entities in Web Text. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India (2007)
9. Sumida, A., Torisawa, K., Shinzato, K.: Concept-instance relation extraction from simple noun sequences using a full-text search engine. In: Proceedings of the ISWC 2006 workshop on Web Content Mining with Human Language Technologies (WebConMine), Athens, GA (2006)
10. Cimiano, P., Staab, S.: Learning by Googling. SIGKDD Explorations Newsletter **6**(2) (2004) 24 – 33
11. Ravichandran, D., Hovy, E.: Learning surface text patterns for a question answering system. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, PA (2002) 41 – 47
12. Cilibrasi, R., Vitanyi, P.: Automatic meaning discovery using Google. http://www.cwi.nl/˜paulv/papers/amdug.pdf (2004)
13. Zadel, M., Fujinaga, I.: Web services for music information retrieval. In: Proceedings of 5th International Conference on Music Information Retrieval (ISMIR'04), Barcelona, Spain (2004)
14. Véronis, J.: Weblog (2006) http://aixtal.blogspot.com.
15. Geleijnse, G., Korst, J., de Boer, V.: Instance classification using co-occurrences on the web. In: Proceedings of the ISWC 2006 workshop on Web Content Mining with Human Language Technologies (WebConMine), Athens, GA (2006) http://orestes.ii.uam.es/workshop/3.pdf.
16. Mori, J., Tsujishita, T., Matsuo, Y., Ishizuka, M.: Extracting relations in social networks from the web using similarity between collective contexts. In: Proceedings of the 5th International Semantic Web Conference (ISWC 2006). Volume 4273 of LNCS., Athens, GA, Springer (2006) 487 – 500
17. Jin, Y., Matsuo, Y., Ishizuka, M.: Extracting a social network among entities by web mining. In: Proceedings of the ISWC 2006 workshop on Web Content Mining with Human Language Technologies (WebConMine), Athens, GA (2006)
18. Zhou, G., Su, J.: Named entity recognition using an hmm-based chunk tagger. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, PA (2002) 473 – 480
19. Brothwick, A.: A Maximum Entropy Approach to Named Entity Recognition. PhD thesis, New York University (1999)
20. Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), Ann Arbor ,MI (2005)
21. Korst, J., Geleijnse, G., de Jong, N., Verschoor, M.: Ontology-based extraction of information from the World Wide Web. In: Intelligent Algorithms in Ambient and Biomedical Computing. Philips Research Book Series. Springer (2006) 149 – 167