Searching for Knowledge Instead of Web Sites

Fabian Suchanek und Gerhard Weikum
Max Planck Institute for Computer Science, Saarbrücken, Germany
Department for Databases and Information Systems
suchanek@mpii.mpg.de, weikum@mpii.mpg.de

January 18, 2008

Abstract

Today's Web search engines can find Web pages that contain certain keywords. Up to now, however, any advanced information demands that concern facts from multiple Web pages, let alone a logical connection between them, are inherently beyond the answering capabilities of search engines. This is why our approach is to collect information from Web sites and to organize it in a huge knowledge structure, an ontology. Using pattern extraction, structural analysis and statistical learning methods, we have developed tools that can automatically build and maintain such an ontology from the contents of the Wikipedia encyclopedia and other Internet sources. Our ontology, coined Yago, contains about one million entities and concepts, and knows more than six million facts about them. Yago also has a Web interface for answering knowledge queries online. Yago could be the starting point for a new generation of search engines, for searching the Web as well as digital libraries and e-science repositories.

This article describes the state of Yago in the year 2007. It has been translated from German by Krista Ames.

Searching on the Web

The internet has developed into a significant source of information over the last decade. Train schedules, news, academic articles, company data, and even entire encyclopedias are available online. The majority of these web pages is indexed by search engines. For example, Google enables us to search for keywords on billions of internet pages in a few split seconds. This technology is entirely sufficient for a lot of enquiries. We normally find what we are looking for after browsing through the search results for a short while. If, for example, you search for the physicist Max Planck, Google gives you a link to the Max Planck Society and several of the physicist's biographies. Even questions like

"When was Max Planck born?" get answers quick as a shot: "Max Planck - Date of Birth: 23 April 1858".

However, the internet user sometimes reaches the limits of what this technology can do. If you want to know, for example, which physicists were born in the same year as Max Planck, there is no suitable wording for a Google query. Any queries with "physicist, born, year, Max Planck" only get results about Max Planck. Thus, you are forced to google Max Planck's birth date first, and then ask about physicists also born in that year. If you would like more extensive information on the subject (which of these physicists was also politically active?), there is no way around reading through the respective internet pages.

The reason for this inconvenience is that Google cannot search for knowledge, but rather for web sites. Google can only satisfy the requests for information for which there is a ready-made answer on a web page. If the answer is scattered over several pages or is only obtainable through logical deduction, then Google is the wrong horse to bet on. The problem here, regarded abstractly, is that today's computers merely possess texts and not knowledge. This lack of general knowledge is also responsible for the often amusing machine translations, to name one example. If we were successful in taking all of the knowledge in the world and making it available to the computer as one giant composition of knowledge, it would be better capable of tackling these tasks.

Knowledge Representation by Ontologies

This kind of structured collection of knowledge is called an "ontology". In the simplest case, an ontology is a directed graph whose nodes are "entities" and whose edges are "relations". The entity "Max Planck" is linked to the entity "23 April 1858" by a "bornOn" relation, because Max Planck was born on 23 April 1858. Although this model is subject to restrictions, many facts can be represented in this way.

We put entities with several common features together into so-called classes. Max Planck and his fellow avid physicists, for example, all belong to the "physicists"' class. In this ontology, the "physicists" class is yet another entity to which all physicists are connected by an "isa" relation. Every physicist is a scientist, so that the "physicists" and "scientists" classes are both in the "subclass" relation. This results in a hierarchy of classes in which each of the more general (upper) classes includes the more specific (lower) classes.

For the next abstraction, we introduce a differentiation of words and their meanings. We distinguish between "Max Planck" (the word) and Max Planck (the physicist). This makes sense because different words can refer to the same individual (for example, "Dr. Planck or "M. Planck"). On the other hand, the same word can refer to different individuals (there are, for example, several people named "Planck").

Beyond that, this classification system abstracts away the choice of language. That means, put simply, that the words "Physiker", "physicist" and "physicien" can all refer to the "physicists" class. In our ontology, these words are nothing more than entities. This kind of ontology is often expanded to include axioms.

One of the most basic axioms says that an entity also belongs to all the classes above its own. So if it is known that Max Planck was a physicist, it follows that - because of the subclass relationship of physicist and scientist –Max Planck was also a scientist. If every physicist is a scientist and every scientist a person, then, every physicist is a person, too (transitivity of the subclass relationship).

An axiomatic system can also express that two relations are inverse to one another, that certain relations determine each other causally, or that time intervals include each other. In this way, a computer that knows Max Planck's place and date of birth and his day of death can come to the logical conclusion that Max Planck was a German scientist who survived both world wars.

Advanced knowledge representations use logical formulas to present facts like: Every person has two parents of two different sexes, a postdoctorate scientist has a Ph.D. advisor, a professor has to have published something, and so on. We might also represent speculative knowledge or leave room for different meanings of words, for which we can then provide probabilities. For example, with "Paris" we might mean the capital city of France or a character from Ilias, or we want to address the competing hypotheses for the cause of an illness, or record the measurement of uncertainty for Mars's time of circulation around the sun. We have to combine logical and probabilistic methods for knowledge representation in cases like these.

Ontologies play a central role in the vision of a "Semantic Web", which is seen by the WWW-inventor, Tim Berners-Lee, as the next generation of the current Web2.0-wave. It should then become possible to establish a direct relation between web sites and entities, along with the cognitive concepts behind them, and then draw intelligent conclusions by logic algorithms. You could then find the best clarinet teacher located less than half an hour away from your daughter's high school. For this to be possible, all web sites must be annotated explicitly with ontological concepts and represented in a logic formalism. Even today, such an undertaking requires an enormous amount of error-prone manual work for every web site, so that fundamental scalability problems are (at the moment) in the way of making this vision reality all too quickly. Our current research work on the subject of intelligent searches for knowledge has the same goal as Semantic-Web-Vision, but we use methods that come from readily available data sources out of which complex collections of knowledge are assembled.

Automatic Construction and Maintenance of Ontologies

The pivotal question is how to fill an ontology with knowledge. There are several approaches. One possibility is to insert all the entities and relations by hand. In fact, the most wide-spread ontologies today have been compiled manually bit for bit. WordNet is a English lexicon with 200,000 entries in an ontological structure. SUMO is an ontology with a hundred thousand entities, and the commercial ontology, Cyc, even contains two million facts and axioms. In spite of these great amounts of knowledge, an ontology assembled by hand will

simply not keep pace with current advancements. None of the above-mentioned ontologies knows, for example, the most recent Windows system or the names of players from the last soccer world cup.

That is why we, at the Max Planck Institute for Computer Science, are looking at different approaches for the construction and maintenance of ontologies. One approach uses the big online encyclopedia, Wikipedia. Wikipedia contains items on a zillion people, products, terms, and organizations. Each of these items has been put into certain categories. For example, the article about Max Planck can be found in the categories, "German", "physicist", and "born in 1858." We utilize this information to record the class and the date of birth of the "Max Planck" entity in the ontology. Although Wikipedia knows a great number of individuals, it doesn't offer a well-structured hierarchy of classes. The information that "physicists" are "scientists" and that "scientists" are "people" is difficult to find in Wikipedia. For this reason, we combine the data from Wikipedia with the data from the earlier mentioned WordNet-ontology through an automated process. In this way, we obtain a large knowledge structure in which all entities known to Wikipedia have their right place. We also utilize other structured knowledge sources (like the film data bank IMDB).

Unfortunately, not all knowledge is available in an already structured form. The most common internet site style is unstructured, natural-language text. Good examples are biographies, lexicon entries, or news texts. We use an approach called "Pattern Matching" to collect this information. If you want to add new birth dates to the ontology, you have to first look at existing web sites with birth dates to find out which pattern they are most often written in. A prevalent pattern for birth dates is, for example, "X was born on Y" ("Max Planck was born on April 23rd, 1858"). If you search the internet for further occurrences of this pattern, other pairs of people and birth dates are unearthed and can be added to the ontology. This approach is unsuccessful when small changes in sentence structure are made which can ruin the pattern. For example, the pattern "X was born on Y" doesn't fit the sentence "Max Planck, the great physicist, was born on 23 April 1858". This is why we have refined the pattern-matching-approach to consider the grammatical structure of the sentences. The pattern then only requires that the "X" remains the subject of the predicate "was born", which is in turn connected with the "Y" by the "on". Now this pattern also fits the sentence "Max Planck, the great physicist,...".

The pattern extractions behind this learning process have been implemented in the software tool "Leila" developed at the institute. To keep Leila from being fooled by the variety and haziness of natural language and generating false hypotheses for patterns too quickly, sample candidates are being checked for robustness in a statistical learning test. Leila extracts predominantly correct facts. For example, it can learn with a great deal of confidence from the collection of all Wikipedia articles that world weariness is a feeling, that Calcutta is located on the Ganges River Delta and Paris on the Seine River - namely from sentences like "Calcutta is on the delta of the Ganges River" and "Paris has a lot of museums along the banks of the Seine River", and even that Saarlander is an ethnic group but a hamburger (the sandwich) is not.

Yago

By combining these techniques, we have been successful in building a large ontology: Yago (Yet Another Great Ontology). Yago has almost one million entities and knows around 6 million facts about these entities at the moment. The core of Yago contains - as an experiential evaluation shows - almost exclusively correct facts, which we have extracted and organized with our most robust methods from Wikipedia articles and their combination with WordNet. We can add more knowledge automatically with the analysis of web sites and databases using tools like Leila. If, in the process, statistical learning processes and heuristics come into play, you could expect the correctness rate to decrease. However, if you already have a first-class ontology at your disposal, as in our case, you can validate new hypotheses by checking their consistency with the ontology. You simply add those new facts that do not conflict with the ones already there. Each new and valuable fact not only makes the ontology grow, but is then also available and useful in judging further hypotheses. In a way, this learning process is self-regulating; the more knowledge Yago acquires, the easier and more robust the acquisition of new knowledge becomes.

The Search for Knowledge

Our collection of knowledge, Yago, is available online[1] and can respond to queries by means of a special query language. Our original question, "Which physicists were born the same year as Max Planck?" can be formulated for Yago as follows:

```
"Max Planck" bornInYear $year
```
(The variable $year will contain Max Planck's date of birth.)
```
$otherPhysicist bornInYear $year
```
(We are asking about another person also born in $year...)
```
$otherPhysicist isa physicist
```
(...on the condition that this person is also a physicist.)

Yago answers promptly with several dozen other physicists. Should you want to know which of them was also politically active, you add the condition "$other-Physicist isa politician". Yago answers that Thomas King, from New Zealand, served in the parliament in addition to being an astronomer. (Remark: In the meantime, Yago has been extended and improved, so that the above query would have to be formulated in a different way. The general principle, however, remains the same.)

[1] `http://www.mpi-inf.mpg.de/yago`

These methods for knowledge searches with ontologies can also be integrated into future search engines, leading to a more powerful form of knowledge search and networking on the largest corpora of our planet. At the Max Planck Institute for Computer Science, we are working on methods for a more intelligent search engine, which represents explicit knowledge from the contents of all web sites, digital libraries and e-science databases - with concepts (e.g. enzymes, quasars, poets, etc.) and entities (e.g. Steapsin, 3C 273, Bertolt Brecht, etc.) and their relations - and makes them findable with a high degree of accuracy. Such a Search engine would be a breakthrough for the step forward from the advanced information society to a modern knowledge society, where all the world's knowledge is not only on the internet, but can also be used effectively.

References

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum:
"Yago - A Core of Semantic Knowledge".
16th international World Wide Web conference (WWW 2007).

Fabian M. Suchanek, Georgiana Ifrim and Gerhard Weikum
"Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents"
Knowledge Discovery and Data Mining (KDD), Philadelphia, USA, 2006

Peter Baumgartner and Fabian M. Suchanek:
"Automated Reasoning Support for First Order Ontologies",
Fourth Workshop on Principles and Practice of Semantic Web Reasoning. Lec-

ture Notes in Computer Science, Springer-Verlag, 2006.

Steffen Staab and Rudi Studer:
"Handbook on Ontologies".
Springer-Verlag, 2004.

Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates:
"Unsupervised named-entity extraction from the Web: An experimental study".
Artificial Intelligence 165(1), pp. 91-134, 2005.