Mining Wikipedia's Snippets Graph: First Step to Build A New Knowledge Base

Andias Wira-Alam and Brigitte Mathiak

GESIS - Leibniz-Institute for the Social Sciences
Unter Sachsenhausen 6-8, 50667 Köln, Germany
{andias.wira-alam,brigitte.mathiak}@gesis.org
http://www.gesis.org/

Abstract. In this paper, we discuss the aspects of mining links and text snippets from Wikipedia as a new knowledge base. Current knowledge base, e.g. DBPedia[1], covers mainly the structured part of Wikipedia, but not the content as a whole. Acting as a complement, we focus on extracting information from the text of the articles. We extract a database of the hyperlinks between Wikipedia articles and populate them with the textual context surrounding each hyperlink. This would be useful for network analysis, e.g. to measure the influence of one topic on another, or for question-answering directly (for stating the relationship between two entities). First, we describe the technical parts related to extracting the data from Wikipedia. Second, we specify how to represent the data extracted as an extended triple through a Web service. Finally, we discuss the usage possibilities upon our expectation and also the challenges.

Keywords: knowledge extraction, wikipedia, knowledge base

1 Motivation and Problem Descriptions

In the recent years, the development of the Semantic Web Technologies has been growing very fast. A lot of research efforts aimed to develop ontology-based reasoning frameworks as the foundation of the Semantic Web's infrastructure. Nevertheless, building and maintaining good ontologies are such intellectual efforts that they have to be done mostly by humans (known as gold standard).

One of the major breakthroughs in the Semantic Web is to extract facts based on the structured information provided using ontologies. For instance, a structured data set such as DBPedia can be queried using SPARQL endpoint to list all cities in Europe with more than one million inhabitants.

The answer of such query lies both in the completeness and the relevance of the information provided. However, there is no guarantee that the answer given is always accurate, since it just represents the current state of the entities. A typical property, namely a time factor, is generally considered in increasing the complexity of the problems, thereby it has been typically excluded. Whereas, we note that "omitting a time factor" can reduce the accuracy, especially in stating historical facts. As a common case, a name of a city might be changed, a capital

of a country might be moved from one to another, or a number of inhabitants might change over time.

Let us assume another query that tries to find "things in common" between two cities in Germany, namely Bonn and Berlin. Obviously, it is simple to extract information that both cities are located in Germany since both belong to the same class, e.g. by querying any ontology which contains this information. Nevertheless, since the time factor is not considered, one of the most striking facts is missing: Bonn was also capital of Germany which moved to Berlin subsequently.

DBPedia provides structured information extracted from Wikipedia, but since it does not consider all parts of the article, the information in the body of the article remains "hidden". However, the text of the article is missing or rather not being extracted. As an analogy, a simple query in DBPedia to return all places that have a connection to Barack Obama only returns four locations[1]: the United States, Northern Mariana Islands, Virgin Islands, and Puerto Rico, with the relationship types `leader` and `isLeaderOf`. Other important locations such as Washington, D.C. or Chicago are missing. In contrast, if we have a look at the article of Barack Obama in Wikipedia, there are many links to places, e.g. a place where he was born, where he had lived, studied, worked, etc.

1.1 Links and Snippets

As we mentioned above, our work complements the current knowledge base such as DBPedia. But however, we do not attack these two problems: adding time factor and vocabulary completeness of the current available ontologies. We rather focus on the providing textual information attached in a simple ontology. Extracting information from the text of articles also produces many potential benefits and can reveal many interesting facts. To encompass this, we need to mention our starting point for this work. Strictly speaking, each Wikipedia article has a unique title and contains terms that point out to other articles.

Naturally, since the terms depict the title of the referenced articles, which are explicitly hyperlinked, they drive the walk of the readers from one to the other articles. Therefore, we believe that most readers pay more attention to such an area around the hyperlinks of the articles rather than the whole parts. This particular area is usually an excerpt or paragraph in the article, which we call it as snippet, with hyperlinks in it. By reading the snippets and following the links, the readers expect to get useful, coherent information effectively.

2 Experiments and Data

Wikipedia provides static dump-files[2] regularly in their website `http://dumps.wikimedia.org/`. For our purpose, we import four dump-files, which are `p`ages-articles.xml.bz2, `p`agelinks.sql.gz, `r`edirect.sql.gz, and `c`ategory.sql.gz. The first

[1] retrieved in Jan 2011.

[2] we currently focus only on the English and German Wikipedia, dump used enwiki20110526 and dewiki20120225.

dump-file contains the page IDs, page contents (text of the articles), and other information regarding to the page. The second contains the linkage / linking between all pages. The last two are also needed since it contains mapping information of the redirected pages and category respectively.

We use Debian/GNU Linux running on dual Intel®Core™2 CPUs with 2TB Harddisk and 3GB RAM. In order to import the XML dump into the database, pages-articles.xml.bz2, we use mwimport[3] to transform it into SQL format. Overall, the import process and indexing took place in several hours, but still in acceptable range. Table 1 shows the overview of the table records and the space usage after the dump-files had been imported.

Table name	*Number of records (EN/DE)*	*Size(EN/DE)*	*Summary*
`page`	11263184 / 2736906	1.9GB / 441.8MB	It contains all pages, which page articles have a namespace equals to 0.
`text`	11263184 / 2736906	35GB / 7.3GB	It contains the text of all pages.
`redirect`	5651143 / 980268	640MB / 30.3MB	It assigns all redirected pages.
`pagelinks`	565218236 / 82208776	44GB / 6.3GB	It contains all page-to-page links, which links among page articles have a namespace equals to 0.

Table 1. Number of records and space usage of the tables in the database.

2.1 Web Service / API

The Web Service provides a public API in order to get access to the data. The data will be provided in an N-Quads[5] format as $(subject, predicate, object, context)$. As an explanation, *subject* and *object* denote the titles of the articles, while *context* is the text snippet. Since we only consider *outlinks* of the articles, we only have one *predicate*, namely *has_link_to*. As an example, the following is an extended triple containing the relationship between *Bonn* and *Berlin*:

(<Bonn> <has_link_to> <Berlin> "Bonn is the 19th largest city in Germany... it was the capital of West Germany from 1949 to 1990 and... government institutions were moved from Bonn to Berlin...").

More importantly, we continually develop the web service not only to provide access to the data, but also to add with some features, e.g. measuring similarity scores between entities based on various algorithms. This might interest other researchers in this field.

[3] mwimport is a Perl script to import WikiMedia XML dumps, details see: `http://meta.wikimedia.org/wiki/Data_dumps/mwimport`

To extract the snippets, we use `mwlib`[4] to parse the Wikitext[5] and decompose it into text segments[6]. As we mentioned above, the snippet is simply expected as a paragraph. But however, as a trade-off, the snippets could be meaningless, e.g. if a snippet extracted from a link that is located in a table or item list. Eventually, the Web service processes a query posed by users, e.g. Bonn and Berlin with a maximum hyperlink distance[7], and gives the extended triples in N-Quads format as a result.

3 Discussions

Since the links graph is also accessible, it can be used to calculate the ranking of the articles by using PageRank or HITS algorithm, as part of network analysis reported by [7]. As we provide the data in such way that it is easy to be reused, it would be simple to compute it.

Recently, [2] developed such methods to compute the influence of a document to another. These methods are also supposed to reveal the *track* of the knowledge flows. However, the users have to read through all provided documents, instead of only reading a useful summary, which is not efficient. Analogous to entity linking task, e.g. [9, 10], we aim not only at the linking between entities but also how to describe their relationships. Moreover, in contrast to a question-answering system, e.g. IBM Watson[3] and YAGO2[11], which gives one specific answer to a complex question, our aim is to give a description about the relationships between two entities - in other words, to give a complex answer to a simple question.

By describing the relationships between two articles, we expect that the *influence* of an article to another article can be computed. As an illustration, Figure 1 shows how an article, Artificial Intelligence, might have an influence to another article, Semantic Web. In this earlier work[4], we evaluated the possibility of enriching the description of the relationships between two entities, which are Wikipedia articles, by leveraging the link snippets. However, it is an early phase of this work and a further approach to analyze the snippets must be further investigated. In our recent studies [8], we showed that describing a relationship between entities helps users to gain a better understanding.

4 Challenges

A big challenge is that we need to research on how to rank the snippets according to the relevancy and accuracy. For instance, in a query asking about the

[4] `mwlib` is a Python library for parsing MediaWiki articles, details see: `http://code.pediapress.com/wiki/wiki/mwlib`

[5] Wikitext is a markup language used to write pages in Wikipedia.

[6] to the date of the submission, we are still working on the Web service and we expect to finish it before the conference.

[7] according to our test, the maximum distance of 2 or 3 can be processed in a reasonable time.

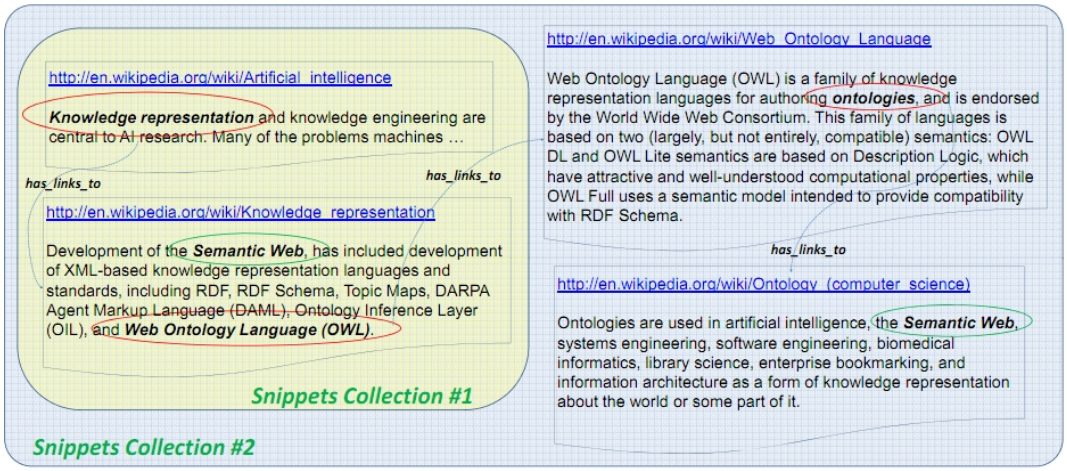

Fig. 1. The snippets collection as an expected result in answering a query on "how Artificial Intelligence influences Semantic Web". It could show the *transformation* between knowledge subjects.[4]

connection between Barack Obama and Kenya, one of the most desired results, intuitively, is that his father was originally from Kenya. Most search engines can provide good results explaining that relationship, but however the results are provided by relying on individual documents. If no single document on the Web covers both particular entities, the results are rather ropey. Unlike in most search engines, where the results provided are relied on individual documents, we believe that our approach will contribute to fill this gap.

Furthermore, we are still left with the problem on how to justify how good the snippets are. To best of our knowledge, this problem is novel and there is no general solution for this problem. We need to specify criteria or objectives in order to determine the quality of a snippet. The length of the snippet is an important objective; we consider a paragraph is an ideal length. Intuitively, each paragraph in an article represents a sub topic or idea, hence a paragraph could be meaningful to substantiate the relations. Using a simple technique such as Automated Content Extraction (ACE), we could extract basic relations between entities. Nevertheless, in order to extract richer relations, we need to find patterns that can recognize the relations between entities.

YAGO2 covers the anchor texts from the hyperlinks to add a textual dimension, however the snippets that more than just anchor texts are not further investigated. Most types of relation are extracted from the sentences by recognizing entities and their properties. Nevertheless, a type of relation such as `formerCapital`, as of the previous example about Bonn and Berlin, might not be extracted from the sentence, therefore the snippets are useful in this sense.

5 Acknowledgments.

We would to thank many colleagues at our Institute for the fruitful discussions as well as the reviewers for their useful inputs and advice on improving our work.

References

1. DBPedia Project. http://www.dbpedia.org/.
2. Shahaf, D., Guestrin, C.: Connecting the Dots Between News Articles. In Proceedings of the 16th Conference on Knowledge Discovery and Data Mining (KDD-2010), 2010.
3. Watson: IBM DeepQA Project. http://www-03.ibm.com/innovation/us/watson/.
4. Wira-Alam, A., Zapilko, B., Mayr, P.: An Experimental Approach for Collecting Snippets Describing the Relations between Wikipedia Articles. Web Science Conference (WebSci10). 2010.
5. N-Quads: Extending N-Triples with Context. http://sw.deri.org/2008/07/n-quads/
6. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In Proceedings of the 6th Symposium on Operating Systems Design and Implementation, 2004.
7. Bellomi, F., Bonato, R.: Network Analysis for Wikipedia. http://www.fran.it/articles/wikimania_bellomi_bonato.pdf
8. Mathiak, B., Martinez-Pena, V., Wira-Alam, A.: WHAT IS THE RELATIONSHIP ABOUT? Extracting Information about Relationships from Wikipedia. In Proceedings of the 8th International Conference on Web Information Systems and Technologies.
9. Lehmann, J., Monahan, S., Nezda, L., Jung, A., Shi, Y.: LCC Approaches to Knowledge Base Population at TAC 2010. In Proceedings of the Third Text Analysis Conference (TAC 2010).
10. Fernandez, N., Fisteus, J. A., Sanchez, L., Martin, E.: WebTLab: A cooccurrence-based approach to KBP 2010 Entity-Linking task. In Proceedings of the Third Text Analysis Conference (TAC 2010).
11. Hoffart, J., Suchanek, F., Berberich, K., Weikum, W.: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. Special Issue of the Artificial Intelligence Journal, 2012.