What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content

Sören Auer[1,2] and Jens Lehmann[1]

[1] Universität Leipzig, Department of Computer Science, Johannisgasse 26, D-04103 Leipzig, Germany
{auer,lehmann}@informatik.uni-leipzig.de
[2] University of Pennsylvania, Department of Computer and Information Science Philadelphia, PA 19104, USA
auer@seas.upenn.edu

Abstract. Wikis are established means for the collaborative authoring, versioning and publishing of textual articles. The Wikipedia project, for example, succeeded in creating the by far largest encyclopedia just on the basis of a wiki. Recently, several approaches have been proposed on how to extend wikis to allow the creation of structured and semantically enriched content. However, the means for creating semantically enriched structured content are already available and are, although unconsciously, even used by Wikipedia authors. In this article, we present a method for revealing this structured content by extracting information from template instances. We suggest ways to efficiently query the vast amount of extracted information (e.g. more than 8 million RDF statements for the English Wikipedia version alone), leading to astonishing query answering possibilities (such as for the title question). We analyze the quality of the extracted content, and propose strategies for quality improvements with just minor modifications of the wiki systems being currently used.

1 Introduction

Wikis are established means for the collaborative authoring, versioning and publishing of textual articles. Founded on Ward Cunninghams design principles[1], wikis dramatically simplify the process of creating and maintaining content by a community of readers and at the same time authors.

A large variety of wiki systems for all possible technical environments and application domains emerged, ranging from lightweight personal wikis focusing on personal information management to full-fledged enterprise wiki systems with integrated groupware functionality. Services based on wikis, such as the provision of wiki farms, support knowledge base wikis, or the maintenance of wikis as intranet sites are provided and employed by startup companies and established global players. Countless special interest wikis on the Web build enormous content collections from travel information for the global village (e.g. on Wikitravel) to local news and gossip on city wikis (such as stadtwiki.net).

[1] http://c2.com/cgi/wiki?WikiDesignPrinciples

E. Franconi, M. Kifer, and W. May (Eds.): ESWC 2007, LNCS 4519, pp. 503–517, 2007.

However, the most famous and successful wiki project probably is Wikipedia[2]. It succeeded in creating the by far largest encyclopedia authored by a globally distributed community just on the basis of a wiki. Wikipedia editions are available in over 100 languages with the English one accounting for more than 1.5 million articles.

To be able to 'tap' this knowledge by machines, recently, several approaches have been proposed on how to extend wiki systems to allow the creation of structured and semantically enhanced content. Modulo minor variations, all of them suggest to enrich the textual wiki content with semantically interpretable statements. The project Semantic Wikipedia [15,24] for example proposes to integrate typed links and page attributes into Wiki articles in a special syntax. It is a straightforward combination of existing wiki systems and the Semantic Web knowledge representation paradigms.

Unfortunately, this approach has several drawbacks: Wikipedia authors have to deal with another means of syntax within wiki texts (in addition to many existing ones). Adding more and more syntactic possibilities counteracts ease of use for editors, thus antagonizing the main advantage of wikis - their unbeatable simplicity. In addition to that, existing (possibly already structured) content in Wikipedia may have to be manually converted or even duplicated. Finally, the approach requires fairly deep changes and additions to the MediaWiki software with unknown effects on its scalability. Scalability is, due to persistently enormous growth in access rates, presently the most pressing technical issue Wikipedia has to face.

However, the means for creating semantically enhanced structured content are already available and used (although unconsciously) by Wikipedia authors. More precisely, MediaWiki, the wiki software behind Wikipedia, enables authors to represent structured information in an attribute-value notation, which is rendered inside a wiki page by means of an associated template. Many Wikipedia pages contain templates, often for layout purposes, but still approximately a quarter to one third of the Wikipedia pages already today contain valuable structured information for querying and machine interpretation.

To be able to query, recombine and reason about this structured information, we present in this paper methods (a) to separate valuable information from less important one, (b) to extract this information from templates in wiki texts and convert it into RDF under usage of unified data types, (c) to query and browse this information even though its schema is very large and partly rudimentary structured. Further, we analyze the quality of the extracted content, and propose strategies for quality improvements with just minor modifications of the wiki systems being currently used.

2 Knowledge Extraction from Wikipedia Templates

We are not aware of any general approaches for extracting information from all templates in Wikipedia. We will first explain templates and then show how their

[2] http://www.wikipedia.org

inherent structure can be used to extract meaningful information. In contrast to the Semantic Wikipedia approach, this has the advantage that we can use already existing information, i.e. we do not need to modify the MediaWiki software (on which Wikipedia is based) to enrich it with support for expressing RDF statements. Hence, our approach can be of immediate use and overcomes the obstacles outlined in Section 1. Semantically enriching wiki templates has been discussed in [15, Section 3.1]. However, we will show that even with the template mechanisms currently available in many wikis, in particular Wikipedia, it is possible to accurately extract useful information.

2.1 MediaWiki Templates

MediaWiki supports a sophisticated template mechanism to include predefined content or display content in a determined way. A special type of templates are infoboxes, aiming at generating consistently-formatted boxes for certain content in articles describing instances of a specific type. An example of the application of an infobox template for the city Innsbruck and the generated HTML table representation on the resulting Wikipedia page is visualized in Figure 1. Such infobox templates are used on pages describing similar content. Other examples include:

- *Geographic entities:* countries, cities, rivers, mountains, ...
- *Education:* university, school, ...
- *Plants:* trees, flowers, ...
- *Organizations:* companies, sports teams, ...
- *People:* politicians, scientists, presidents, athletes ...

More information can be found in the Wikipedia infobox template article[3].

2.2 Extraction Algorithm

To reveal the semantics encoded in templates we developed an extraction algorithm operating in five stages:

*Select all Wikipedia pages containing templates.*Wikipedia pages are retrieved by an SQL query searching for occurrences of the template delimiter "`{{`" in the `text` table of the MediaWiki database layout. The SQL query can be adopted to select only pages for particular Wikipedia categories or containing specific templates to generate fragments of the Wikipedia content for a certain domain.

Extract and select significant templates. All templates on a Wikipedia page are extracted by means of a recursive regular expression. Since templates serve different needs, we extract those with a high probability of containing structured information on the basis of the following heuristic: templates with just one or two template attributes are ignored (since these are templates likely to function as shortcuts for predefined boilerplates), as well as templates whose usage count is below a certain threshold (which are likely to be erroneous). The extraction

[3] `http://en.wikipedia.org/wiki/Wikipedia:Infobox_templates`

```
1  {{Infobox Town AT |
2    name = Innsbruck |
3    image_coa =  InnsbruckWappen.png |
4    image_map = Karte-tirol-I.png |
5    state = [[Tyrol]] |
6    regbzk = [[Statutory city]] |
7    population = 117,342 |
8    population_as_of = 2006 |
9    pop_dens = 1,119 |
10   area = 104.91 |
11   elevation = 574 |
12   lat_deg = 47 |
13   lat_min = 16 |
14   lat_hem = N |
15   lon_deg = 11 |
16   lon_min = 23 |
17   lon_hem = E |
18   postal_code = 6010-6080 |
19   area_code = 0512 |
20   licence = I |
21   mayor = Hilde Zach |
22   website = [http://innsbruck.at] |
23 }}
```

Fig. 1. Example of a Wikipedia template and rendered MediaWiki output for Austrian towns applied for Innsbruck

algorithm can be further configured to ignore certain templates or groups of templates, based on specific patterns.

Parse each template and generate appropriate triples. A URL derived from the title of the Wikipedia page the template occurs in is used as subject for templates which occur at most once on a page. For templates occurring more than once on a page, we generate a new identifier being used as subject. Each template attribute corresponds to the predicate of a triple and the corresponding attribute value is converted into its object. MediaWiki templates can be nested, i.e. the attribute value within a template can again be a template. In such a case, we generate a blank node linking the attribute value with a newly generated instance for the nested template.

Post-process object values to generate suitable URI references or literal values. For MediaWiki links (e.g. "`[[Tyrol]]`" in line 4 of Figure 1) suitable URI references are generated referring to the linked Wikipedia article. (Currently, we ignore the special case that the link denoting brackets could be part of the template definition.) Typed literals are generated for strings and numeric values. Common units (such as m for meter, kg for kilogram, s for seconds) are detected and encoded as special datatypes (cf. Table 1). However, a conversion between different scales (e.g. between mm, cm, m, km) is not performed. Furthermore, comma separated lists of the form `[[Jürgen Prochnow]]`, `[[HerbertGrönemeyer]]`, `[[Martin Semmelrogge]]` are detected and, depending on configuration options, converted into RDF lists or individual statements.

Table 1. Detection of literal datatypes (excerpt)

Attribute type	Example	Object data type	Object value
Integer	7,058	xsd:integer	7058
Decimals	13.3	xsd:decimal	13.3
Images	[[Image:Innsbruck.png|30px]]	Resource	c:Innsbruck.png
Links	[[Tyrol]]	Resource	w:Tyrol
Ranks	11th	u:rank	11
Dates	[[January 20]] [[2001]]	xsd:date	20010120
Money	$30,579	u:Dollar	30579
Large numbers	1.13 [[million]]	xsd:Integer	1130000
Big money	$1.13 [[million]]	u:Dollar	1130000
Percent values	1.8%	u:Percent	1.8
Units	73 g	u:Gramm	73

Determine class membership for the currently processed Wikipedia page. Wikipedia pages are organized in categories. In some cases, these can be interpreted as classes subsuming instances described by Wikipedia pages in the corresponding category. Furthermore, the name of the template can be an indicator for a certain class membership. The categories itself are Wikipedia pages and are often organized into super-categories. Unfortunately, here the sub-category super-category relationship often refers more to being "related-to" than constituting a subsumption relation. We are currently working on improving class membership detection.

2.3 Extraction Results

We tested the extraction algorithm with the English Wikipedia content (available from `http://dumps.wikimedia.org/enwiki`). The overall time needed to extract template instances and convert them to RDF for the approx. 1.5 Mio English Wikipedia articles (accounting for roughly 10GB raw data) was less than one hour on a computer with Xeon 2.80GHz CPU and 1GB of main memory. The raw extraction results as well as the source code of the extraction algorithm are available from `http://wikipedia.aksw.org/`.

Table 2 shows some extraction statistics. The first column contains general information about extracted quantities. Overall, more than 8 Mio triples were obtained from the English Wikipedia. Each triple belongs to one of about 750,000 templates, which can be grouped in approx. 5,500 template types. This means a template is used 137 times on average. In the extracted ontology, 650,000 individuals are connected by 8,000 properties and the class hierarchy consists of 111,500 classes (all numbers approximated).

The second column displays the most frequently used templates and how much instances of them exist in Wikipedia. The third column shows similar information for attributes. Table 3 exhibits the properties extracted from some frequently used templates.

Table 2. Extraction results: overall statistics, most used templates, and most used attributes

Statistics:	
Template types	5,499
Template instances	754,358
Templates per type	137.18
Attributes per instance	8.84
Categories	106,049
Classes	111,548
Instances	647,348
Properties	8,091
Triples	8,415,531

Templates:	
succession_box	72262
election_box	48206
infobox_album	35190
taxobox	29116
fs_player	25535
nat_fs_player	15312
imdb_title	15042
infobox_film	12733
imdb_name	12449
fs_squad2_player	10078
infobox_cvg	7930
infobox_single	7039
runway	6653

Attributes:	
name	301020
title	143887
image	110939
years	89387
before	79960
after	78806
genre	77987
type	74670
released	74465
votes	59659
reviews	58891
starring	57112
producer	53370

2.4 Obstacles

Since there are not many restrictions on the design of Wikipedia templates, there are a number of obstacles, which can lead to undesired extraction results in some cases.

First of all, templates are not yet used everywhere in Wikipedia, where they are appropriate. Sometimes tables or other means are used to display structured information.

For certain content (e.g. planets) infobox templates are not used to separate content from presentation, but for each content object a separate template containing attributes is created. Similarly, layout information is sometimes encoded directly in templates (e.g. color information) and templates for certain content are made up of from many small element boxes (e.g. chemical elements), even when this is not necessary.

Table 3. Extracted properties for specific templates

Template/ Class	**No. of Instances**	**Used properties**
Music album	35190	name, artist, cover, released, recorded, genre, length, label, producer, reviews, last album, next album
Species	29116	binomial, genus, genus_authority, classis, phylum, subfamilia, regnum, species, subdivision
Film	12733	starring, producer, writer, director, music, language, budget, released, distributor, image, runtime
Cities	4872	population_total, population_as_of, subdivision_type, area_total, timezone, utc_offset, population_density, leader_name, leader_title
Book	4576	author, genre, release_date, language, publisher, country, media_type, isbn, image, pages, image_caption

Attributes sometimes contain (from a knowledge representation viewpoint) redundant information, whose purpose is more intuitive visual representation as for example: `[[Innsbruck]], [[Austria]]`. In other cases, multiple values are encoded in one attribute instead of cleanly separating them in different attributes, e.g. `foundation = [[California]] ([[April 1]], [[1976]])`. Furthermore, duplicate information is sometimes present in attribute values, e.g. `height = 5'11" (180cm)`. The last example also shows that different units are used as attribute value, often depending on the locality of the intended audience of an article.

2.5 Guide for Designing Semantically Rich Templates

Despite all the obstacles described in the previous section, we were still surprised by the enormous amount of meaningful and machine interpretable information we were able to extract from Wikipedia templates. In order to improve extraction, we want to suggest some guidelines for defining templates in this section. We mention them here to outline how the extraction results could be improved further. Please note, that following these guidelines is not only good for semantic extraction, but usually also improves the corresponding template in general, i.e. it becomes more convenient to use by article authors.

- Do not define attributes in templates, which encode layout information. Rather, let the template handle the representation. (This corresponds to the well known principle of separating content and its presentation.) Along the same line, HTML markup should be used in attribute values only when necessary.
- Use only one template for a particular item of interest, instead of using one template for each attribute. Currently, the later version of templates is still present in many Wikipedia articles, although the former is emerging as the standard.
- Each attribute should have exactly one value within an article template. This value can be a list of values. However, one should not mix several statements (from an RDF point of view) within one attribute value.
- Currently, images in the English Wikipedia are retrieved from two places, depending on the definition of a template: Wikipedia Commons and the media library for the English Wikipedia . Thus, it is not possible to determine the location of an image without analyzing the definition of a template, which is an unnecessary complication. However, Wikipedia Commons is emerging as a standard, so the number of problematic cases is already decreasing.
- Do not use different templates for the same purpose, e.g. there are currently template infoboxes for "Infobox_Film", "Infobox Film" and "Infobox film". This problem is already tackled by the Wikipedia community[4].
- Do not use different attribute names for the same kind of content and do not use the same attribute name for different kinds of content. Support for this can be added to the wiki software (see below).
- Use standard representations for units, such that they can be detected by the extraction algorithm.

[4] `http://en.wikipedia.org/wiki/Wikipedia:Infobox_templates`

Furthermore, the following improvements could be made to the MediaWiki software, which is used for Wikipedia, and other software to make the design of clean templates easier:

- Offer the possibility to fix the data type of an attribute value, e.g. by attribute templates as mentioned above. For instance, the template designer could specify in the template definition that the attribute value of the attribute `budget` has to be a number. Although this greatly improves the extraction process, we are aware that sometimes verbal descriptions are necessary to explain attribute values, e.g. a value for budget could be "estimated to be between 20 and 30 million dollars".
- Offer the possibility of language and unit tags if one attribute value can be given in different languages or units, i.e. the budget can be specified in Euro and Dollar. A name of a French city can be given in English and French.
- If an attribute is defined in a template, the MediaWiki software could list templates, where this attribute already exists and show other characteristics of the attribute, to give the template designer the possibility to check whether these attributes have the same intended meaning.

One of the aims of these proposals is to improve extraction without putting the burden on the user (in this case the article author). In many cases, following the guidelines makes it clearer for the article writer how to use templates and many of these guidelines are common sense. They enable a clean extraction of information without the need to recreate the content of Wikipedia or dramatically change the way templates are currently defined.

3 Browsing and Querying Extracted Knowledge

Compared to most of the other Semantic Web knowledge bases currently available, for the RDF extracted from Wikipedia we have to deal with a different type of knowledge structure – we have a very large information schema and a considerable amount of data adhering to this schema. Existing tools unfortunately mostly focus on either one of both parts of a knowledge base being large, schema *or* data.

If we have a large data set and large data schema, elaborated RDF stores with integrated query engines alone are not very helpful. Due to the large data schema, users can hardly know which properties and identifiers are used in the knowledge base and hence can be used for querying. Consequently, users have to be guided when building queries and reasonable alternatives should be suggested. In this section, we present with the facet-based browsing in OntoWiki and our graph pattern builder two approaches to simplify navigation and querying of knowledge bases possessing a large information schema.

3.1 OntoWiki

To allow browsing of the extracted information in an intuitive manner, we adopted our tool for social semantic collaboration – OntoWiki [2]. It facilitates

the visual presentation of a knowledge base as an information map, with different views on the instance data. It enables intuitive authoring of semantic content and fosters social collaboration aspects (however these features are not of primary interest for browsing the extracted Wikipedia content). Furthermore, OntoWiki enhances the browsing and retrieval by offering semantically enhanced search strategies. In particular the facet-based browsing implemented in OntoWiki allows to intuitively explore the extracted Wikipedia content.

Taxonomic structures give users only limited ways to access the information. Also, the development of appropriate taxonomic structures requires significant initial efforts. Only a very restricted taxonomic structure can be extracted from Wikipedia content by means of categories. As a pay-as-you-go strategy, facet-based browsing allows to reduce the efforts for a knowledge structuring, while still offering efficient means to retrieve information. To enable users to select objects according to certain facets, all property values (facets) of a set of selected instances are analyzed. If for a certain property the instances have only a limited set of values, those values are offered to restrict the instance selection further. Hence, this way of navigating through data will never lead to empty results. The analyzing of property values as well as the appropriate filtering and sorting of instances though can be very resource demanding. Since the respective optimizations in OntoWiki are not yet finished we deployed just an excerpt of the extraction for the film domain for demonstration at `http://wikipedia.aksw.org`. However, we aim to adopt and optimize OntoWiki further to function as easy to use browser for the complete semantic Wikipedia content.

3.2 Graph Pattern Builder

In addition to support browsing with OntoWiki we specifically developed a graph pattern builder for querying the extracted Wikipedia content. Users query the knowledge base by means of a graph pattern consisting of multiple triple patterns. For each triple pattern three form fields capture variables, identifiers or filters for subject, predicate and object of a triple. While users type identifier names into one of the form fields, a look-ahead search proposes suitable options. These are obtained not just by looking for matching identifiers but by executing the currently built query using a variable for the currently edited identifier and filtering the results returned for this variable for matches starting with the search string the user supplied. This method ensures, that the identifier proposed is really used in conjunction with the graph pattern under construction and that the query actually returns results. In addition, the identifier search results are ordered by usage number, showing commonly used identifiers first. All this is executed in the background, using the Web 2.0 AJAX technology and hence completely transparent for the user. Figure 2 shows a screenshot of the graph pattern builder.

3.3 Example Queries

In the previous sections, we have shown how to extract information from templates. We gave an impression about the sheer volume of knowledge we obtained

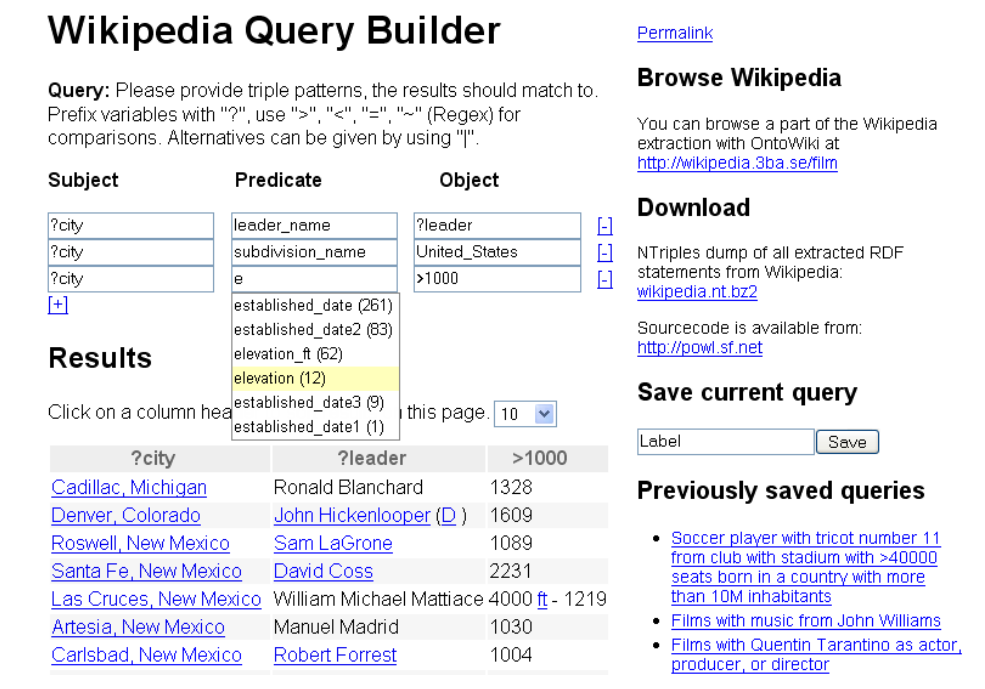

Fig. 2. Query interface of the Wikipedia Query Builder with AJAX based look-ahead identifier search

and suggested ways to improve templates to ease the conversion to RDF triples. The aim of this subsection is to present some example queries to justify our claim that we can obtain a huge amount of semantically rich information from Wikipedia – even in its current form.

Of course, reasonable queries can only involve a very small fraction of the information we obtained. We invite the reader to browse the obtained knowledge and pose example queries by visiting `http://wikipedia.aksw.org`.

The first query uses the film template. We ask for films starring an Oscar winner (as best actor) with a budget of more than 10 million US dollars[5].

```
1  SELECT ?film ?actor ?budget
2  WHERE {
3     ?film p:starring ?actor;
4           p:budget ?budget.
5     ?actor a c:Best_Actor_Academy_Award_winners.
6     FILTER regex(str(?budget),"1[0123456789]{6,}")
7  }
```

[5] The examples make use of the namespace prefixes `rdf` for RDF, `xsd` for XML-Schema data types, `p` for generated property identifiers, `w` for Wikipedia pages, `c` for Wikipedia categories and `u` for units.

The following table summarizes the result:

film	actor	budget
w:The_Da_Vinci_Code_(film)	w:Tom_Hanks	"125000000"^^u:Dollar
w:Ghost_Rider_(film)	w:Nicolas_Cage	"120000000"^^u:Dollar
w:Apocalypse_Now	w:Robert_Duvall	"31500000"^^u:Dollar
w:Jackie_Brown_(film)	w:Robert_De_Niro	"12000000"^^u:Dollar
w:Bobby_(2006_film)	w:Anthony_Hopkins	"10000000"^^u:Dollar
w:Confidence_(film)	w:Dustin_Hoffman	"15000000"^^u:Dollar
w:Apocalypse_Now	w:Marlon_Brando	"31500000"^^u:Dollar
w:The_Mission_(film)	w:Robert_De_Niro	"17218000"^^u:Dollar
w:The_Silence_of_the_Lambs	w:Anthony_Hopkins	"19000000"^^u:Dollar

Note the automatic detection of the budget unit (all US dollars in this case). The next query is more complex, involving different kinds of templates, namely soccer players, soccer clubs, and countries. We ask for soccer players with number 11 (on their jersey), who play in a club whose stadium has a capacity of more than 40000 people and were born in a country with more than 10 million inhabitants.

```
1  SELECT ?player ?club ?country ?capacity
2  WHERE {
3     ?player p:currentclub ?club .
4     ?player p:clubnumber 11 .
5     ?player p:countryofbirth ?country .
6     ?club p:capacity ?capacity .
7     ?country p:population_estimate ?population .
8     FILTER (?capacity > 40000) .
9     FILTER (?population > 10000000)
10 }
11 ORDER BY DESC(?capacity) LIMIT 1000
```

The following table shows the result of the query:

player	club	country	capacity
w:Mehrzad_Madanchi	w:Persepolis_FC	w:Iran	90000
w:Cicinho	w:Real_Madrid	w:Brazil	80354
w:Ram%C3%B3n_Morales	w:Chivas_de_Guadalajara	w:Mexico	72480
w:Lukas_Podolski	w:FC_Bayern_Munich	w:Poland	69901
w:Gonzalo_Fierro	w:Colo-Colo	w:Chile	62000
w:Robin_van_Persie	w:Arsenal_F.C.	w:Netherlands	60432
w:Michael_Thurk	w:Eintracht_Frankfurt	w:Germany	52000
w:Stein_Huysegems	w:Feyenoord_Rotterdam	w:Belgium	51177
w:Mark_Gonz%C3%A1lez	w:Liverpool_F.C.	w:South_Africa	45362

Both queries are interesting and realistic. In both cases the results are probably not complete, because templates are still not used everywhere where appropriate or are badly designed (see Section 2.5). You can find more queries and build your own ones at `http://wikipedia.aksw.org`.

4 Related Work

The free encyclopedia Wikipedia has been tremendously successful due to the ease of collaboration of its users over the Internet [16]. The Wikipedia wiki is the representative of a new way of publishing [4] and currently contains millions of articles.

It is a natural idea to exploit this source of knowledge. In the area of Machine Learning [17] the Information Retrieval community has applied question answering [14], clustering, categorization and structure mapping to Wikipedia content. An XML representation of (most of) the Wikipedia corpus has been extracted [6] to support these tasks. Within the Semantic Web community this corpus has been used to extract common sense knowledge from generic statements [22] by mapping them to RDF.

In a different ongoing project the link structure and basic metadata of Wikipedia articles are mapped to a constantly updated RDF dataset, which currently consists of approximately 47 million triples [19]. Amongst other uses, the link structure has been exploited to build semantic relationships between articles to analyze their connectivity [5], which can improve the search capabilities within Wikipedia.

In contrast to the information extraction approaches mentioned above, i.e. full text extraction and link structure extraction, we focused on the extraction of Wikipedia templates. The corresponding algorithm, presented in Section 2.2, uses standard pattern matching techniques [1] to achieve this. We argued that the resulting RDF statements are a rich source of information contributing to existing results of knowledge extraction from Wikipedia. The goal of a related, but much more focused project is to extract personal data from the "Personendaten" template in the german Wikipedia[6].

Our research also fits within the broader scope of integrating Semantic Web standards with different sources of information like LDAP servers [7] and relational databases [3]. The integration of different data sources is considered an important issue in Semantic Web research [8].

Additionally, there are strong links to knowledge extraction from table structures. A general overview of work on recognizing tables and drawing inferences from them can be found in [27]. [21] is an approach for automatic generation of F-Logic frames out of tables, which subsequently supports the automatic population of ontologies from table-like structures. Different approaches for interpreting tables [12,13,25], i.e. deriving knowledge from them, have been tested for plain text files, images and HTML tables. Naturally, an additional difficulty of these approaches compared to the template extraction we perform, is to properly recognize tables (see e.g. [9,11,18,26] for table recognition techniques) and the relationships between entries in these tables using techniques like row labeling and cell classification [11,20]. Amongst other target formats for extraction, there has also been work on ontology extraction from tables [23], in particular for HTML tables [10]. Since templates in Wikipedia have a predefined structure,

[6] `http://de.wikipedia.org/wiki/Hilfe:Personendaten/Datenextraktion`

our results are most likely more accurate than those one would obtain using more general table extraction approaches. (We are working on measuring the quality of the information we have extracted.) For this reason, we did not consider to apply general-purpose extraction of HTML tables in Wikipedia, but focused on the already structured templates.

5 Conclusions

As we outlined in the introduction, we created an approach for extracting information from Wikipedia and similar wiki systems, which is usable right now without further modifications on the MediaWiki software or expensive updates of Wikipedia content. We showed that we obtained a vast amount of useful machine processable information. Obstacles of our approach were discussed and suggestions for solving them have been given. They mostly involve common sense rules for template authors and reasonably small modifications of existing wiki software.

Further, we discussed the problem of querying and browsing the extracted knowledge. A simple and easy-to-use query engine for our extraction was developed and some example queries were presented to give the reader an intuition of the extracted knowledge. As with textual Wikipedia content the templates might contain incomplete or even wrong information. This cannot be detected by the extraction algorithm and prospective users should be aware of it. However, by providing a platform exhibiting the structured and interlinked content in Wikipedia, we were able to create one of the largest existing ontologies and hopefully a useful contribution to the Semantic Web in general. The presented approach was implemented for MediaWiki and evaluated with Wikipedia content. However, it can be similarly applied to different Wiki systems supporting templates and other means for handling frequently occurring structured content.

Finally, we want to solve the question from the title of this article for a commonality between Innsbruck and Leipzig: A fairly simple query on our extraction results reveals that both share Kraków as a twin town. However, this information can not be found on either of the Wikipedia pages and we are not aware of knowledge bases able to answer similar unspecific and domain-crossing queries.

Acknowledgments

This research was supported in part by the following grants: BMBF (SE2006 #01ISF02B), NSF (SEIII #IIS-0513778). We are grateful to Jörg Schüppel, who participated in the implementation of the extraction algorithm, and the anonymous reviewers for their suggestions.

References

1. A. Apostolico and Z. Galil, editors. *Pattern Matching Algorithms*. OUP, 1997. SEP.
2. Sören Auer, Sebastian Dietzold, and Thomas Riechert. Ontowiki - A tool for social, semantic collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 736–749. Springer, 2006.
3. Christian Bizer. D2R MAP - A database to RDF mapping language. In *WWW (Posters)*, 2003.
4. Bryant, Susan L., Andrea Forte, and Amy Bruckman. Becoming wikipedian: transformation of participation in a collaborative online encyclopedia. In *GROUP'05: International Conference on Supporting Group Work*, Net communities, pages 1–10, 2005.
5. S. Chernov, T. Iofciu, W. Nejdl, and X. Zhuo. Extracting semantic relationships between wikipedia categories. In *1st International Workshop: "SemWiki2006 - From Wiki to Semantics" (SemWiki 2006), co-located with the ESWC2006 in Budva, Montenegro, June 12, 2006*, 2006.
6. L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.
7. S. Dietzold. Generating rdf models from ldap directories. In C. Bizer S. Auer and L. Miller, editors, *Proceedings of the SFSW 05 Workshop on Scripting for the Semantic Web, Hersonissos, Crete, Greece, May 30, 2005*. CEUR Workshop Proceedings Vol. 135, 2005.
8. Dimitre A. Dimitrov, Jeff Heflin, Abir Qasem, and Nanbor Wang. Information integration via an end-to-end distributed semantic web system. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 764–777. Springer, 2006.
9. Shona Douglas and Matthew Hurst. Layout and language: lists and tables in technical documents. In *Proceedings of ACL SIGPARSE Workshop on Punctuation in Computational Linguistics*, pages 19–24, jul 1996.
10. David W. Embley, Cui Tao, and Stephen W. Liddle. Automatically extracting ontologically specified data from HTML tables of unknown structure. In Stefano Spaccapietra, Salvatore T. March, and Yahiko Kambayashi, editors, *Conceptual Modeling - ER 2002, 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7-11, 2002, Proceedings*, volume 2503 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2002.
11. J. Hu, R. S. Kashi, D. P. Lopresti, and G. T. Wilfong. Evaluating the performance of table processing algorithms. *International Journal on Document Analysis and Recognition*, 4(3):140–153, 2002.
12. M. Hurst. Layout and language: Beyond simple text for information interaction – modelling the table. In *Proceedings of the 2nd International Conference on Multimodal Interfaces, Hong Kong*, 1999.
13. M. Hurst. *The Interpretation of Tables in Texts*. PhD thesis, University of Edinburgh, 2000.

14. B. Katz, G. Marton, G. Borchardt, A. Brownell, S. Felshin, D. Loreto, J. Louis-Rosenberg, B. Lu, F. Mora, S. Stiller, O. Uzuner, and A. Wilcox. External knowledge sources for question answering. In *Proceedings of the 14th Annual Text REtrieval Conference (TREC2005), November 2005, Gaithersburg, MD*, 2005.
15. Markus Krötzsch, Denny Vrandecic, and Max Völkel. Wikipedia and the Semantic Web - The Missing Links. In Jakob Voss and Andrew Lih, editors, *Proceedings of Wikimania 2005, Frankfurt, Germany*, 2005.
16. Bo Leuf and Ward Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison Wesley, Reading, Massachusetts, apr 2001.
17. Thomas Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.
18. Hwee Tou Ng, Chung Yong Lim, and Jessica Li Teng Koo. Learning to recognize tables in free text. In *ACL*, 1999.
19. System One. Wikipedia3. http://labs.systemone.at/wikipedia3, 2006.
20. David Pinto, Andrew McCallum, Xing Wei, Croft, and W. Bruce. Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, IR theory, pages 235–242, 2003.
21. Aleksander Pivk, Philipp Cimiano, and York Sure. From tables to frames. *Journal of Web Semantics*, 3(2-3):132–146, 2005.
22. S. Suh, H. Halpin, and E. Klein. Extracting common sense knowledge from wikipedia. In *Proceedings of the ISWC-06 Workshop on Web Content Mining with Human Language Technologies*, 2006.
23. Yuri A. Tijerino, David W. Embley, Deryle W. Lonsdale, and George Nagy. Ontology generation from tables. In *WISE*, pages 242–252. IEEE Computer Society, 2003.
24. Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *Proceedings of the 15th international conference on World Wide Web, WWW 2006*, pages 585–594. ACM, 2006.
25. Xinxin Wang. *Tabular abstraction, editing, and formatting*. PhD thesis, Waterloo, Ont., Canada :University of Waterloo, Computer Science Dept.,, 1996.
26. Yalin Wang, Ihsin T. Phillips, and Robert M. Haralick. Table structure understanding and its performance evaluation. *Pattern Recognition*, 37(7):1479–1497, 2004.
27. R. Zanibbi, D. Blostein, and J. R. Cordy. A survey of table recognition: Models, observations, transformations, and inferences. *International Journal on Document Analysis and Recognition*, 7(1):1–16, mar 2004.