

Semantic Enterprise Search (but no Web 2.0)

Ronald Winnemöller

Abstract—In this paper, we propose semantic enterprise search as promising technical methodology for improving on accessibility to institutional knowledge. We briefly discuss the nature of knowledge and ignorance in respect to web-based information retrieval before introducing our particular view on semantic search as tight fusion of search engine and semantic web technologies, based on semantic annotations and the concept of intra-institutionwise distributed extensibility while still maintaining free keyword search functionality. Consequently, our architecture implementation makes strong use of the Aperture and Lucene software frameworks but introduces the novel concept of "RDF documents". Because our prototype system is not yet complete, we are not able to provide performance statistics but instead we present a concise example scenario.

Index Terms—RDF documents, semantic web, ignorance.

I. INTRODUCTION

INTUITIVELY, it is the duty of Universities (and, to a certain degree, of technikons and other schools) to produce knowledge in research and teaching.

This, we might assume, is what they do very well.

Unfortunately, we may also find that keeping, consolidating and making accessible that knowledge, even when we restrict ourselves to electronically stored knowledge, is a field that is neglected in many cases – a fact that is acknowledged by several institutions as stated in the *Implementation of the Berlin Declaration on Open Access*, cf. [1] and the *Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities* itself, cf. [2].

We can, for example, identify the following “knowledge leaks” as common for many institutions:

- The conventional, if not required publishing process often means that a researcher sums up his knowledge in a journal article or conference paper – which eventually gets published by some commercial company. It does not necessarily mean that this publication is kept (electronically) in the realm of the home university of the mentioned researcher and be available to other members – students or fellow researchers – of that institution. This issue is also reflected on in [1].
- Many institutions run web-accessible publication databases, like eprint archives, citation indexes and such – sometimes even at the department or team level. While

these repositories usually provide for intra-repository search functionality, they cannot be searched using an institution-wise global methodology because of the well known “hidden web” problem (cf. e.g. [3], [4]).

- ELearning, content management and “Web 2.0” systems (such as departmental wikis, blogs, etc.) usually require authentication (and authorization) prior to accessing content. These knowledge sources cannot be accessed through global methods without special adaption to these requirements. Even then, automatic retrieval methods still face the above mentioned “hidden web” problem.
- Other types of knowledge material are transmitted electronically but are of transient nature, such as RSS-Feeds (on an organizational basis), mails, filestores, etc. We hesitate to call these types “documents” because of their temporary character - but nevertheless they might contain valuable knowledge nonetheless.
- Some publications are simply not available electronically, because they were published through a “pure-paper”¹ process.
- Certain documents may be accessible through the “visible” intraweb of an institution, but due to an ineffective implementation of the retrieval process they may still be inaccessible.

In this paper, we propose a *technical* methodology for improving on accessibility to institutional knowledge.

We will not, however, try to solve *social* institutional issues such as implementing open access publishing processes, etc.².

The remainder of this paper is structured as follows:

In the next section, we will briefly discuss the term “knowledge”. After this, we will put our work in an appropriate scientific context in section III, followed by a description of our own approach (section IV), including our proposed architecture and already implemented modules. Since we are still working on a concise evaluation procedure, we will instead preliminarily provide a realistic scenario in section IV-C as indication for the intended functionality. Subsequently, we conclude in section V

II. A BRIEF ELABORATION ON A SEARCH-RELATED NOTION OF “KNOWLEDGE”

In order to clarify what we are talking about, we need to discuss what we mean when using the term “knowledge”.

Manuscript received February 2, 2009. Manuscript accepted for publication March 17, 2009.

The author is with University of Hamburg, Germany; email: ronald.winnemoeller@uni-hamburg.de

¹As opposed to “paper-less”.

²We will rather leave this issue to the Web2.0 community ...

queries are subsequently built through means of a wizard, selecting attributes from the background ontologies (SUMO⁸ and others). The system itself is specialized in US-Patent discovery, which resembles - in our view - an enterprise search environment.

A more general approach was published by Duke et al. (cf. [15]) - their “Squirrel” system is also based on a specific ontology (in this case, the ontology allows to create user interest profiles which can be interpreted as pragmatic approach to ignorance modelling as described in section II) and integrates several semantic components.

Using quite a different methodology, Peter Mika presented “Microsearch” (cf. [5]), a Software developed at Yahoo! Research, which is able to enhance conventional search results by visualizing embedded metadata. The metadata Microsearch uses, stems from microformats, embedded in otherwise standard HTML pages. His approach explicitly assumes further advances (in terms of quantity of annotated web pages) in the Web 2.0 movement but he claims that even now Microsearch may motivate users to provide metadata for web pages thus bootstrapping its own data basis. Mika also proposes the positive effect of aggregating information from different result pages, e.g. by combining personal and geographic information.

While there is much more current related work worth to be discussed here, we decided to place the more pragmatic approaches next to ours in order to enable comparing sometimes subtle differences.

IV. A SEMANTIC APPROACH ON SEARCH

While there still exists a situation of hidden knowledge and knowledge leaks, and the open access movement still appears to be in it’s early stages, a conventional tool to look up institutional information is “enterprise search”, i.e. web search technology, imposed on an intranet infrastructure.

Unfortunately, current search engine technology is mostly based on (syntactic) open web⁹ search (cf. [13], [16], etc.), which in turn is based on common information retrieval techniques. These provide only basic tools, which are not very effective in a highly socialized and informationwise fine grained environment. Other tools, like link structure exploitation, also don’t work too well here (cf. [17]). To be more specific: while intranet *recall* seems an issue of providing a highly customized technical solution, the *precision* of search results can - by definition - only be raised by tuning the search engines relevancy¹⁰ algorithms.

Semantic Web methods on the other hand are well suited to shape indexed knowledge according to the real informational situation and needs of institution members. Providing semantically rich machine readable information about resources and the principle of distributed extensibility are key aspects of the Semantic Web theory. Yet, one major drawback is that they

still depend on a large amount of manual annotation work (sometimes it is simply assumed that the WWW will eventually contain appropriately annotated resources, cf. [18]). This indeed is one well-known problem of knowledge engineering, that annotating text basically is a huge amount of work with no apparent use to the annotator himself. Even in cases where people apparently want to annotate text (e.g. via the so-called “Web 2.0” technologies, i.e. folksonomies and such) they do it rather in a way that they gain reputation in their respective community but not in order to provide semantic annotations for automatic information retrieval (cf. [5]). Because of this issue, we think that annotation must come from automatic methods, if they are to be employed on large volumes of data.

Furthermore, we like the view of Chakrabarti that schema-free searches must be enabled, but schema knowledge should be honored by a query language (this enables freetext keyword searches but still rewards complex processing, cf. [19]). While following his advice forbids using strict schematic query wizards or formalized query languages (as proposed in [12] or [14]), it reveals the necessity for applying NLP methods (and enabling manual editing) in order to discover semantic relationships within the data.

These thoughts, in combination with those given in section II, lead us to an approach that combines enterprise search with semantic web technology.

While this in general is not really a novel idea, we *will* add some new aspects to it, expecting to overcome the difficulties we described beforehand.

A. The Hypothesis

As stated in section III on the preceding page, we aim for *Semantic Search* but not for *Semantic Web Search* for the following reasons: Search is often limited to searching literal text *or* URI nodes and is implemented as specific function within a RDF framework (cf. [20] or [21], [22]). We feel that is an unnecessary limitation because search functionality and RDF framework functionality should be tightly and efficiently integrated. It also should be possible to integrate schema information and use description logics and such when required.

We propose that fusing search engine and semantic web technology at the right level, i.e. enabling semantic annotations and intra-institutionwise distributed extensibility – while maintaining freetext search functionality – will create a certain amount of synergy which can raise the effectiveness of a semantic search approach in an institutional (enterprise) environment. From our preliminary evaluation of some query logs of our institution we found that queries are strongly biased towards personal information (~28% of all queries) and organizational or structural queries, related to the institution (~36%), such as querying for departments, scripts, elearning courses, etc.. This enterprise-search related aspect of course will have a great impact on the kind of semantics we need to employ — especially named entity processing should be treated with high priority.

⁸<http://www.ontologyportal.org/>, accessed 17.10.2008

⁹i.e. extranet/WWW

¹⁰See section II on page 1

be constructed to split documents at certain points and merge others for several reasons, leaving references to the original pages only as in-document “`rss:link`”-properties. This allows for creating “views” on the data where each view shows a different semantical focus, a different interpretation of the content.

The most important key heuristic is hidden in the post-processing step of our architecture – by querying the index we encounter three cases:

- 1) Using conventional keywords only: documents containing these keywords will be discovered and ranked according to the Lucene $tf \times idf$ scheme. Additionally, RDF URI nodes can be discovered, too – exploiting the fact that most RDF URIs contain semantically relevant, human readable parts. For example, a keyword search for “bob homepage” will also reward indexed items containing “`<foaf:homepage>`” – especially when in conjunction to the literal fragment “bob”. This can be quite useful, because many homepages in an institutional environment do not explicitly state that they are homepages!
- 2) Submitting a mixture of keywords and RDF URIs: queries like “`foaf:homepage bob`” will find “Bobs homepage” – but not “Jills homepage” with a reference to Bob! Because Lucene query analyzers eliminate non-alphanumeric characters, a domain-less URI is treated like a keyword; i.e. the query “`:homepage bob`” will not be restricted to the FOAF¹² domain but rather work like an ordinary keyword query in the above explained way. In the special case of web documents containing microformats such as RDFa¹³ these will be implicitly honoured the same way.
- 3) Submitting RDF URIs only will exhibit documents with certain semantic properties: the query “`foaf:homepage`” will return all indexed items that contain homepages in the sense of the homepage element of the FOAF schema, plus the FOAF schema itself (as it also contains the fragment “`foaf:homepage`”).

The query results (possibly filtered by a predefined document relevance threshold or by a first- N -documents-only heuristic) are merged into a single resulting RDF model that can be searched by means of templates, implemented structured RDF querying languages, e.g. SPARQL (cf. [29]), in order to provide end-user application functionality¹⁴. Most prominently, this will be the list of relevant links to web pages (fetched by applying a SPARQL search for “`rss:link`”-nodes), but a wealth of other applications is possible as well (for other examples, cf. e.g. [5]).

In this way, a query-centric RDF model is constructed dynamically on each search occasion that reflects the “ignorance-artifact” created by the user. Because schemata are discovered

as well (and can be further tracked by using the PREFIX RDF document fields), we are not restricted to structured RDF queries only but can also apply description logics in order to further examine query results. For example, we can deduce subclassing etc. On the other hand, when it’s just a portion of the textual content, it is being searched for, we can simply output the value of the “`nie:content`” predicate triple. In this way, we are able to defer complex processing until it is really needed.

C. An Example Scenario

Let us suppose, for instance, an information pool of public staff profiles should be created that provides research domain information. The system should answer questions that are semantically similar to “I am new to this university. which professor can i ask about topic X ?”¹⁵.

Our proposed system would employ a web crawler to harvest our university web documents. The resulting RDF documents are then run through a POS-tagger, a text boundary detector and a person entity tagger within the UIMA framework (and possibly more components). Subsequently, a set of person information extraction components (e.g. phone number extractors, etc.) is applied on the resulting data, thus creating new RDF documents that resemble person profiles¹⁶ of the gathered data and links to the original web pages. Further components might collect bibliographic references associated with these people and search library databases for associated keywords. The keywords found are then consolidated into a short list and added to the respective persons profile. After indexing the profile RDF document, a user can search for these keywords in order to find people as well.

For example, when crawling the university website, we see that a person named Wolfgang M. was co-author of a paper titled “Hybrid parsing: using probabilistic models as predictors for a symbolic parser” – the university library database associates the following fictitious keywords to this article: *linguistics parsing stochastic grammar*, and other publications of this person may include the keywords: *analysis aspects reasoning artificial intelligence*. From his web page, we discover that Mr. M. is currently professor at the informatics department at the University of Hamburg. Creating a list of keywords from these publications and associating the most common ones with the author results in a set of author-related keywords. These can be added to the profile RDF document which in turn is associated with Mr. M.’s homepage. Other information (such as his profession) is added in the same way. A hypothetical user querying for “professor artificial intelligence linguistics” will then discover the (high ranked) homepage of Wolfgang M. even though it factually only contains the term “artificial”.

¹⁵we don’t aim for a natural language interface - we are just describing the question informally. a real search engine will require a more formal specification.

¹⁶Please note that an appropriate level of privacy and security must be maintained but discussing this is beyond the scope of this paper.

¹²cf. <http://xmlns.com/foaf/spec/>, accessed 29.10.2008

¹³cf. <http://esw.w3.org/topic/RDFa>, accessed 29.10.2008

¹⁴These templates are supposed to be pre-built by staff members

- [24] D. Ferrucci and A. Lally, "Uima: an architectural approach to unstructured information processing in the corporate research environment," *Nat. Lang. Eng.*, vol. 10, no. 3-4, pp. 327-348, 2004.
- [25] R. Winnemöller, "Using meaning aspects for word sense disambiguation," in *9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, Haifa, Israel, February 2008.
- [26] S. Raghavan, R. Krishnamurthy, E. Kandoga, Y. Mass, N. Har'El, M. Bluger, H. Zhu, G. Ramakrishnan, K. Sah, and S. Vaithyanathan, "Ibm omnifind personal e-mail search," <http://www.alphaworks.ibm.com/tech/emailsearch>, accessed October 2008.
- [27] Y. Lei, V. S. Uren, and E. Motta, "Semsearch: A search engine for the semantic web," in *Knowledge Acquisition, Modeling and Management (EKAW)*, S. Staab and V. Svátek, Eds., Pödebrady, Czech Republic, October 2006, pp. 238-245.
- [28] J. Umbrich and S. Blohm, "Exploring the knowledge in semi structured data sets with rich queries," in *Proceedings of the Workshop on Semantic Search (SemSearch 2008)*, Tenerife, Spain, June 2008, pp. 89-101.
- [29] S. Schenk, "A sparql semantics based on datalog," in *KI 2007: Advances in Artificial Intelligence*. Springer, 2007, pp. 160-174. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74565-5_14