SWSE: Answers Before Links!

Andreas Harth, Aidan Hogan, Renaud Delbru, Jürgen Umbrich, Sean O'Riain, and Stefan Decker

National University of Ireland, Galway
Digital Enterprise Research Institute
firstname.lastname@deri.org

Abstract. We present a system that improves on current document-centric Web search engine technology; adopting an entity-centric perspective, we are able to integrate data from both static and live sources into a coherent, interlinked information space. Users can then search and navigate the integrated information space through relationships, both existing and newly materialised, for improved knowledge discovery and understanding.

1 Introduction

Todays best in class search engines such as Google, Yahoo and MSN offer search over web documents. Paramount to their success is their data gathering prowess and their ability to allow consumers quickly and efficiently find documents matching a set of keywords. Results in such sites are simply links to documents; users are required to manually traverse these documents in order to achieve the answers to their information needs. Ranking tries to present relevant documents first, but little effort is made to enhance, extract or integrate data to provide precise answers, such as "who are the friends of Rudi Studer", "what organism pertains to the Protein SHNF1", or "where is SK Telekom located"?

In addition, most current search engines offer little support for disambiguation or refinement of results: adroit keyword query construction and reformulation is required to avoid trawling through a quagmire of irrelevant results. More recent developments in the search space have seen Vivisimio[1] offering search and clustering capabilities, where search is initially driven by keyword but also features filtering using generated clusters which provides the 'context' of the results. Quintura[2] also offers dynamic cluster creation based on keyword queries, with analysis of the contextual relationships between keywords. The result is a visual semantic keyword cloud that the user may call upon to further filter the result set. However, all of these tools are still based upon the traditional document-centric view of knowledge rather than real-world entities and their relationships. On the other hand, search engines operating on structured data sources, such as

[1] http://vivisimo.com/
[2] http://quintura.com/

A9's Web 2.0 offering[3], only merge information visually at the syntactic level. True integration at the data level is not attempted.

Broaching these key issues, SWSE (Semantic Web Search Engine) performs semantic integration of structured data: not only from the Web but also from monolithic data sources such as XML database dumps, large static datasets and even live sources; this is achieved using a hybrid data integration solution which amalgamates the data warehousing and on-demand approaches to integration. From this integration emerges a large graph of RDF entities with inter-relations and structured descriptions of entities: archipelagos of information coalesce to form a coherent knowledge base. This perspective of knowledge is a better reflection of its subject than the traditional document-centric philosophy. Entities are typed according to what they describe: people, locations, organisations, publications as well as documents; entities have specified relations to other entities: people know other people, people author documents, organisations are based in locations, and so on.

Since the entity-centric model closely reflects the real (and online) world, it becomes viable to develop a search and query engine which users will find intuitive to use. Users initially define a keyword search to hone in on relevant entities, results can then be refined according to type; users can then navigate to and from entities through known relations. Thus, rich descriptions of diverse entities are easily retrievable within the interface. Where the required data is not available as structured data, SWSE bridges the gap to traditional search by offering links to documents which are related to the entity.

As we will see, SWSE, by effecting such an entity-centric modus operandi, can truly offer Answers before Links.

2 Example Session

SWSE's user interaction model depends on an entity-centric view of the world: the cognitive model assumes entities with attributes, and relations between entities. The user interface primitives for operating in this space are: keyword matching in attributes, filtering results by entity type, and navigation of relations between entities.

In the following, we describe a use case that shows how SWSE outperforms current web search engines in term of information search potential. SWSE's entity-centric interface extends current search functionalities and assists the user's tasks along two dimensions:

- navigational - reaching a specific entity and exploring its surrounding entities;
- informational - acquiring an extensive representation of an entity.

The use-case we introduce is focused on the task of learning more about a person: Rudi Studer. We examine how a user can procure, with minimal effort, an extended description of a person coming from a multitude of sources

[3] http://a9.com/

Fig. 1. Results list page after typing in keyword.

and describing not only his personal information (contacts, interests, work environment) but also his surrounding entities (people he knows, his projects, his organisation). Please note that the interface, although demonstrated on browsing a social network, is domain independent and can similarly be used to navigate information spaces in other domains as well.

In a common search engine, gathering information about an entity can be a cognitively intensive activity. For example, a user will enter the person's name as keyword key and will get a list of web pages as results. Then, the user must browse and review several web pages and manually filter the information, possibly reformulating the keyword search with extra terms in order to try to find new pages and eliminate irrelevant ones. To gather and assemble a coherent description of the person, current search engines can demand needless user energy and time.

With SWSE, a user will start, as with a common search engine, by entering the keyword query "rudi studer". The result of the search is shown in Figure 1, which is a list of all entities matching the keyword (area 1 in the Figure) accompanied with a small summary description. The entities often have various types, such as Person, Document, Professor (2 in the Figure). In order to refine the search, the user can click on "Person" to filter the results by entity type and get only a list of "Person" entities.

If users click on the first result (the Person Rudi Studer), they get presented (as captured in Figure 2) with a detailed information page about the person. The information about the entity can be aggregated from multiple sources and is presented in a homogeneous view (in this example nearly 200 sources contribute to the Rudi's representation). Here, users can find Rudi's homepage, telephone and fax number, people he knows, and so on (3, Figure 2). Users can then continue their exploration of the surrounding environment of Rudi just by following the semantic links. In addition to following outgoing links, users find an overview of

Fig. 2. Entity page containing information about Rudi aggregated from multiple sources.

incoming relations to the focus instance in a column on the left side of the pane (4). Users are shown that Rudi has authored 198 papers, that 48 people know him, that he is the maker of a file and editor of four things.

SWSE also assists users by linking various RDF entities with named entities found in full-text information (cf. Section 3.2) via additional navigational "see also" shortcuts. For example, when the entity "Rudi Studer" is identified in a document, SWSE is able to instantly provide its full description and to suggest related and relevant documents.

3 Architecture

The architecture is an adaptation of search engine and database/data warehousing architectures. Figure 3 illustrates the high-level architecture and the data flow within the system.

3.1 Semantic Search and Query Engine

The core of SWSE is the Semantic Search and Query Engine as depicted in Figure 3. YARS2 is a scalable distributed architecture for indexing and querying large RDF datasets and operates on a named graph data model, whereby

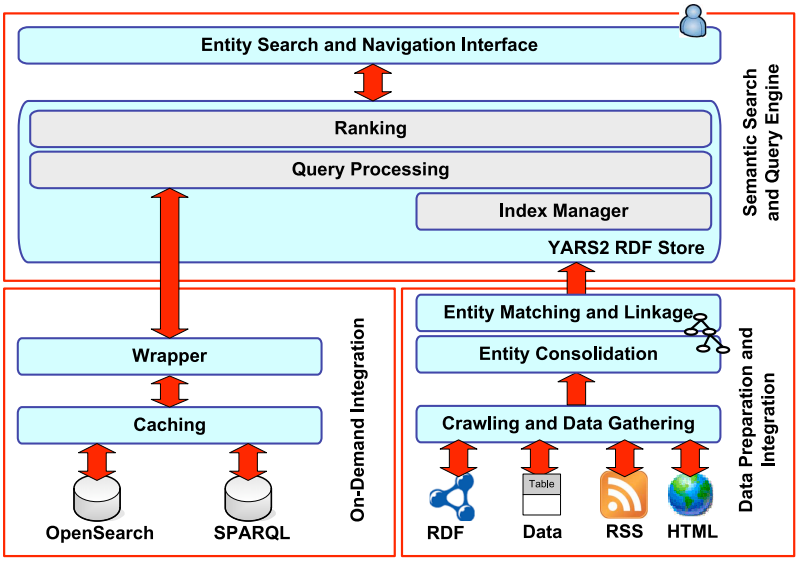

Fig. 3. Semantic Web Search Engine architecture consisting of data preparation and integration phase and Semantic Search and Query Engine.

RDF triples are extended with context, which encodes the source of data, forming a quadruple (subject, predicate, object, context). Within YARS2, there are packages for providing local index creation and management, distributed query processing, and runtime ranking of results. The Index Manager creates and services lookups on local keyword and quad (named graph) indices. The Query Processor co-ordinates with several Index Managers over the network and offers a SPARQL end-point. ReConRank[2] is used to rank entities in the result-set providing metrics for the importance of particular entities and also the trustworthiness of data sources; these metrics are used for ordering the presentation of results in the UI.

3.2 Data Preparation and Integration

In the following, we briefly describe the process of collecting and integrating data from a plethora of sources in a multitude of formats.

In order to acquire our raw dataset, we employed MultiCrawler[1]. To be able to demonstrate the use of real-world heterogeneous information sources with diverse ownership, we crawled native RDF sources, RSS streams, and HTML, MS Office, PS and PDF documents with MultiCrawler extracting metadata and converting to RDF where necessary. Table 1 shows some statistics about the currently used data set.

We enrich and interlink the base dataset using entity consolidation (a.k.a. object consolidation)[3]. As seen, SWSE integrates data from multiple sources

Description	Value
Number of statements	250,298,954
Data size uncompressed	48 GB
Text index size	7.1 GB
Quad index size	23.4 GB

Table 1. Data size and index statistics.

and often, within RDF, different sources may describe the same entities providing complementary data on a particular entity. When a common URI is used to identify the entity, data integration is automatic under the identifier; when URIs are not provided or do not match, we use entity consolidation to identify matches through analysis of values of inverse functional properties. The goal here is to avoid having the knowledge contribution of entities split over numerous instances: i.e., to have a 1:1 ratio of entities to results in the UI.

In addition, we perform entity linking achieving the "see also" links: we use our set of entities from structured data sources as the crystallisation points around which metadata from poorly structured data sources (mostly HTML documents) are arranged. Web documents (HTML pages, RSS feeds) are mainly unstructured but are widespread in the Web and remain a useful source of information that should be leveraged. In order to enable users to locate relevant information from the abundance trapped in poorly structured data, we link web documents with the existing RDF entities. We first create an inverted index over the text of such documents. Then, for each RDF entity, we query the inverted index by using one specific property; e.g., we use foaf:name for matching foaf:Person entities. Finally, for each query hit, we create an association between the web document and the RDF entity.

3.3 On-Demand Integration

Finally, in the architecture, we provide for runtime querying of external live datasources. Wrappers can be plugged into the architecture for querying external sources: the wrappers provide the same interface as an Index Manager and so can be handled by the Query Processor. Each wrapper handles a particular format of external source (e.g., OpenSearch, SPARQL) and handles multi-threaded access to multiple of such sources; the wrappers incorporate Squid caching. Please note, we have currently not enabled any wrappers for the demo.

4 Lessons Learnt

As one would expect, whilst designing, developing and implementing this architecture and its components, there were many lessons learnt and numerous conclusions arrived at.

Our first observation was regarding the importance of extending the RDF model with context. Under the RDF model, whereby community driven knowl-

edge bases are encouraged, anyone can say anything about any resource anywhere. Thus, tracking the source of data is vital to maintaining the integrity of the information provided to users. In fact, we created ReConRank on the premise that sources should also be ranked for a particular result set, offering metrics on trustworthiness and thus taking contextual information into account in the ranking procedure.

Secondly, we learnt the importance of extending our knowledge base with data from non-RDF sources. By only indexing RDF data, the sphere of knowledge indexed was quite limited; thus we created MultiCrawler to crawl and transform other data sources such as HTML and RSS and starting using entity-linking to create the see also links.

Yet another aspect we discovered: sometimes it is not feasible to crawl large database-backed sites with millions of exported files, since harvesting the entirety of such a site at a rate of one page every ten seconds would take in the order of months. We see two alternatives to remedy this problem: either the sites provide data dumps of their datasets for download, or provide a SPARQL interface which allows for on-demand integration of these sources. To be able to detect and utilise the data dumps automatically, we propose to use an extension of the sitemap protocol for semantic crawling[4]. In addition to pointing to data dumps, the sitemap extension allows to discover SPARQL endpoints on the Web.

Data quality is an issue, too. The native data we acquire is sparsely interlinked, since URI's often do not match up: sometimes agreement cannot be reached, othertimes a new URI is created in ignorance of pre-existing ones. Applying entity consolidation becomes a powerful tool in such a scenario that can dramatically improve the quality of the dataset. However, object consolidation can "incorrectly" consolidate instances referring to different entities. This can be attributed to two main factors. Firstly, dud values are often assigned to inverse functional properties such as, "N/A", "foo" or "ask" etc.; these properties match for instances referring to different entities and cause incorrect consolidation. Secondly, properties are sometimes used in a manner contrary to their formal definition as being inverse functional.

Perhaps the current dearth of URI agreement could be attributed to the absence of a site where data providers could view aggregated information and verify the integrity of their data files; we see SWSE as filling this niche and supporting the interlinking and reuse of identifiers for entities on the Web, i.e. SWSE in the short term can act as a reference tool for data providers. Thus, we hope that SWSE will help motivate the production of better quality data.

In general, dealing with web data is more difficult than data provided and managed by an enterprise. We have achieved higher data quality and thus improved browsing capabilities in vertical search settings. Given a confined domain, it is possible to arrive at datasets of better data inter-linkage and data quality using a few high-quality sources. In particular, entity consolidation performs exceptionally in cases where different data sources use different identifiers to denote entities (e.g. ticker symbol, cik, CUSIP in the area of securities). The

[4] http://sw.deri.org/2007/07/sitemapextension/

user interface presented here is domain independent and does only depend on a few selected RDF primitives (such as `rdf:type` and `rdfs:label`); however, in an enterprise setting, making the user interface domain-specific can facilitate much more powerful browsing and navigation functionality at the expense of generality.

5 Conclusion

We have described the application of semantic web technologies to the scenarios of entity-centric search and navigation and large scale web data source integration. The current system is available online[5], and we also provide an experimental SPARQL endpoint[6].

The SWSE architecture features components to crawl, transform, enhance, integrate, index and provide advanced querying and browsing of data from a plethora of sources and formats. SWSE reuses hundreds of thousands of RDF sources on the Web, and assumes a completely open world: new RDF data can easily be integrated, without any change in the architecture or user interface. In summary, we provide a complete end-to-end system for advanced web search.

Of interest to commercial parties is SWSEs capability to provide enhanced and accurately linked information spaces, covering both Web and Intranet document and data repositories, which can then be used for domain specific browsers in a manner that current search engines do not allow.

Particular attention is paid to scale in the architecture. We still have many data-sources to exploit and we foresee both the quality and quantity of RDF data and live data-sources increasing; we look forward to such developments and foresee our system as being able to scale accordingly. We are currently implementing additional OWL reasoning primitives that go beyond the ground equality reasoning required for entity consolidation. Thus, we hope to continue to improve and provide Answers before Links in a new generation of web-search.

References

1. A. Harth, J. Umbrich, and S. Decker. Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In *International Semantic Web Conference*, pages 258–271, 2006.
2. A. Hogan, A. Harth, and S. Decker. ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In *2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.
3. A. Hogan, A. Harth, and S. Decker. Performing object consolidation on the semantic web data graph. In *Proceedings of 1st I3: Identity, Identifiers, Identification Workshop*, 2007.

[5] http://swse.deri.org/
[6] http://swse.deri.org/yars2/