Enabling a Knowledge Supply Chain: From Content Resources to Ontologies

Rodolfo Stecher[1], Claudia Niederée[1], Paolo Bouquet[2], Thierry Jacquin[3], Salah Aït-Mokhtar[3], Simonetta Montemagni[4], Roberto Brunelli[5], and George Demetriou[6]

[1] Fraunhofer IPSI
Integrated Publication and Information Systems Institute
Dolivostrasse 15, 64293 Darmstadt, Germany
{stecher, niederee}@ipsi.fhg.de

[2] University of Trento
Via Sommarive 10, 38100 Trento, Italy
bouquet@dit.unitn.it

[3] Xerox Research Centre Europe
6 Chemin de Maupertuis, F-38000 Meylan, France
{jacquin, Salah.Ait-Mokhtar}@xrce.xerox.com

[4] Istituto di Linguistica Computazionale
Via Moruzzi 56124, Pisa, Italy
simonetta.montemagni@ilc.cnr.it

[5] ITC-IRST
Via Sommarive 18, 38050 Povo, Italy
brunelli@itc.it

[6] University of Sheffield
211 Portobello St., Sheffield S1 4DP, UK
demetri@dcs.shef.ac.uk

Abstract. Semantic annotation of content is a crucial building block of making the Semantic Web fly. The (semi-)automatic support of the underlying semantic knowledge supply chain requires contributions from different research disciplines and well-defined pipelines, which step-by-step create such annotations from raw content objects. This paper presents an annotation pipeline that has been designed and implemented as part of the VIKEF project. A clear structuring of the pipeline, the selection of adequate representation formats for the intermediate results (products) as well as for configuration information have been identified as crucial ingredients for an annotation pipeline, that enables the application-specific customization of the pipeline components and the flexible integration of upcoming advanced methods like new extraction methods into the pipeline.

1 Introduction

Thanks to the considerable efforts spent by the members of the Semantic Web community in the Semantic Web Activity a first important step on the way to the Semantic Web has been completed. Central formats for capturing and describing semantic information (like RDF and OWL [1]) have been developed. They are agreed upon (de facto) standards and are also widely accepted and used within the community. The next big step is the ”operationalization” of the Semantic Web. It is generally understood that, at least initially, there will not be **the** Semantic Web as one big unit. Rather, communities and organizations will implement innovative applications based on Semantic Web technology. These islands might in the future be connected leading to a wider cross-community Semantic Web infrastructure.

For the implementation of such semantic-enabled applications the following challenges have to be met: a) a sufficient amount of content from the respective application domain has to be annotated with semantic information, and b) the different (application-specific) ontologies underlying this semantic information have to be developed, agreed upon, and kept up-to-date. Furthermore, useful semantic-enabled services have to be developed based on the semantic annotations and integrated into applications.

The challenge of ontology development is covered by current work on ontology engineering. Several different approaches focus on complemetary aspects and problems of the ontology engineering process (see e.g. [2] and [3] for an overview). This paper focusses on the challenge of (semi-)automatically annotating content objects of an application domain with semantic information. This task requires a multi-phased process, where linguistic entities discovered within a content object are coupled with domain knowledge represented by an ontology. For effective semantic annotation support, linguistic and knowledge representation aspects, approaches, and formats, have to be combined in a synergetic way. This paper describes a framework and a pipeline (together with the employed representation formats within the pipeline), which supports semantic annotation in a flexible and pragmatic way. The pipeline has been implemented as a prototype developed as part of the VIKEF project [7] and evaluated for content from the scientific domain. The pipeline process is supported by a set of components and tools of the framework so that a power user (user responsible of configuring a pipeline) can configure a new pipeline in a very flexible way. One mayor contribution of this work is the creation of a framework to allow easy customization of such a process.

The paper is structured as follows: Section 2 introduces the semantic annotation pipeline. This includes a description of the different pipeline steps as well as of the employed representation formats. Section 3 gives a short overview of the prototype implementing the annotation pipeline. Section 4 sketches a service that exploits the extracted semantic information to give the user a richer experience in working with content. Section 5 discusses related work.The paper concludes with a summary and a discussion of directions of future work in improving the annotation pipeline.

2 The Semantic Annotation Pipeline

The semantic annotation pipeline consists of a sequence of processing steps each of which produces an intermediate representation that is digested by the next processing step. After introducing the target semantic representation, this section describes the processing steps of the pipeline together with the respresentation formats.

[7] see http://www.vikef.net

2.1 The Target: Semantic Representation

The final target of the annotation pipeline is the explicit and elicited semantic representation of the knowledge implicitly conveyed in the content objects. This requires adequate underlying ontologies and a format for the representation of the semantic information on the instance level, in our case the *Semantic Resource Network* (SRN). A SRN is a specific set of triples representing instances of the used ontologies.

Two types of ontologies are used for representing the output of the Semantic Annotation Pipeline. The first one is a domain ontology which covers the domain of the content sources contained in the analyzed collection and the second one, the Annotation ontology, is an ontology for representing physical location and other information related to the analysed content object and the extracted information. Our current domain ontology is based on the OWL [1] AKT Portal and Support ontologies [4] with some extensions to tailor them to our specific domain characteristics, i.e. the domain of scientific computer science publications. The Annotation ontology represents the annotations with several properties describing them: a) the language, b) the location URL to display it in a browser, c) the size so that a decision can be made if it should be accessed or not, d) the value of the annotated entity, e) a timestamp of the creation of the annotation, f) the mime format of the resource that can be accessed by traversing the given URL, and g) relations to the instance which represents the resource where the annotation is contained (if there is one) as well as h) a reference to a class representing the usage rights of the resource for representing intellectual rights and related information. The definition these properties is based on the LOM [5] definition present in SCORM [6].

Semantic Resource Networks (SRNs) are (A-Box) representations of instances and their relationships in a domain, based on an underlying domain ontology and on the Annotation ontology, represented by RDF graphs [7]. The SRN also contains navigable links to the underlying annotated content to allow later access to the sources.

2.2 Pipeline Overview

We support two pipelines for the construction of SRNs: One based on metadata collections like DBLP for the scientific domain (Pipeline I) and another based on the extraction of semantic information from content objects (Pipeline II). In this paper we focus only on Pipeline II. Figure 1 gives an overview of the processing steps in Pipeline II, which are described in more detail in the sections below.

2.3 Content Harmonization

There are two harmonization levels. The entry-level harmonization is a generic and universal document indexing schema, according to which each XML node is assigned with a unique ID; this ID is preserved through the annotation and exploitation phase. The second-level harmonization implements additional conversion steps (needed for a specific annotation service). The second-level harmonization includes components for layout and logical analysis (e.g. header/footer recognition, reading order reconstruction, paragraph segmentation, image extraction) [8] [9] but can also target semantic annotation [10].

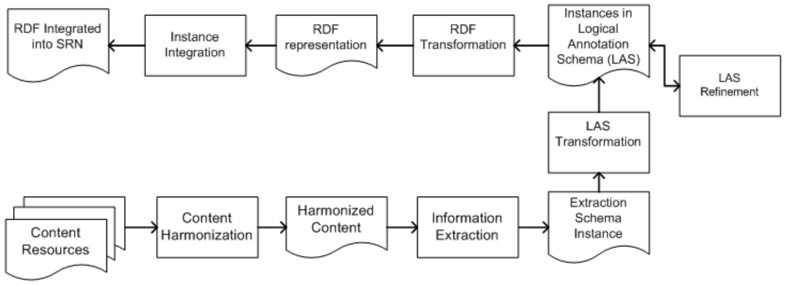


Fig. 1. Pipeline II Overview

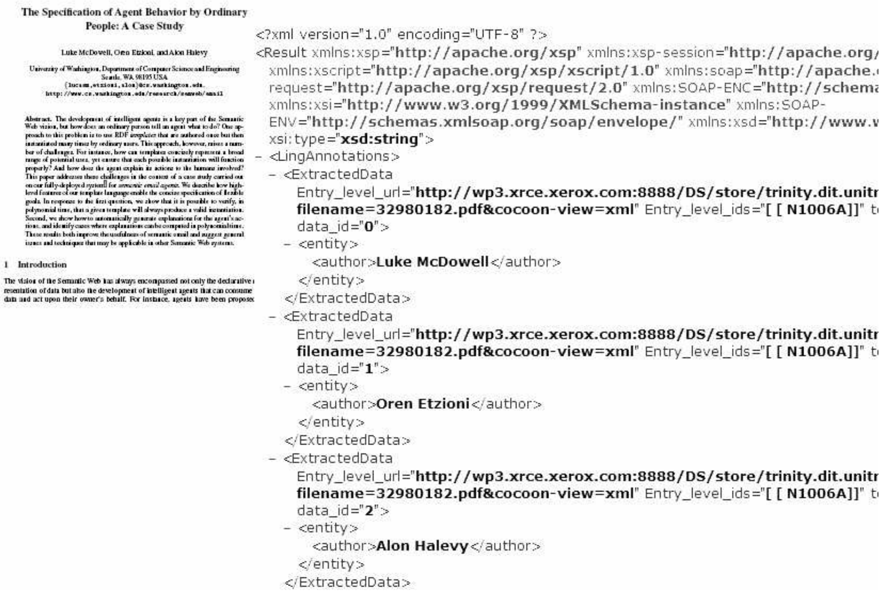
2.4 The Information Extraction Process

Semantic annotation refers to the categorization of document fragments according to predefined categories and to the attachment of semantic/linguistic tags to the classified fragments. The annotations are then used to associate different fragments among them and to discover semantic relationships between the fragments in the same or different resources. The semantic annotation of the content of a document relies on natural language or image extraction tools working on harmonized XML documents.

In the second step of the pipeline, data extracted from document content is represented as stand-alone objects, which conform to the VIKEF XML data Extraction Schema (see figure 2). This pipeline step is open to any image or text extraction service, provided it takes as input harmonized files and produces an output compliant with the data extraction schema.

A number of extraction services have been integrated into the pipeline, especially for processing documents in different languages or annotating the image content. Using the XIP parsing system [11], we have implemented a semantic annotator of English documents in the context of the scientific scenario[8]. The XIP parsing system is a modular, declarative and XML-empowered linguistic analyzer and annotator: it takes XML-based documents as input, linguistically analyzes their textual content (robust parsing) and produces the set of annotations in an XML format as output. Throughout the process, it keeps track of XML-encoded meta-data of the original document. XIP robust parsing provides mechanisms for identifying Named Entity (NE) expressions, and extracting relations between words or group of words, e.g. relations between NE expressions. The annotation prototype we have developed for the VIKEF scientific scenario annotates entities of type PERSON, LOCATION, ORGANISATION, TITLE, etc., and is available as a web service. It is currently being enriched to recognize basic relations between entities (AFFILIATION_OF, LOCATION_OF, etc.) and more advanced semantic annotations (co-reference, temporal relations, and concepts such as ”Novelty”, ”Contribution”, etc. in scientific articles).

[8] In the VIKEF project the support of community events by semantic-enabled services is considered for a scientific scenario (scientific congresses) and for a scenario with business content (trade fairs).



The Specification of Agent Behavior by Ordinary People: A Case Study

Luke McDowell, Oren Etzioni, and Alon Halevy

University of Washington, Department of Computer Science and Engineering
Seattle, WA 98195 USA
{lucasm,etzioni,alon}@cs.washington.edu,
http://www.cs.washington.edu/research/semweb/email

Abstract. The development of intelligent agents is a key part of the Semantic Web vision, but how does an ordinary person tell an agent what to do? One approach to this problem is to use RDF templates that are authored once but then instantiated many times by ordinary users. This approach, however, raises a number of challenges. For instance, how can templates concisely represent a broad range of potential uses, yet ensure that each possible instantiation will function properly? And how does the agent explain its actions to the humans involved? This paper addresses these challenges in the context of a case study carried out on our fully-deployed system for *semantic email agents*. We describe how high-level features of our template language enable the concise specification of flexible goals. In response to the first question, we show that it is possible to verify, in polynomial time, that a *given* template will always produce a valid instantiation. Second, we show how to automatically generate explanations for the agent's actions, and identify cases where explanations can be computed in polynomial time. These results both improve the usefulness of semantic email and *suggest* general issues and techniques that may be applicable in other Semantic Web systems.

1 Introduction

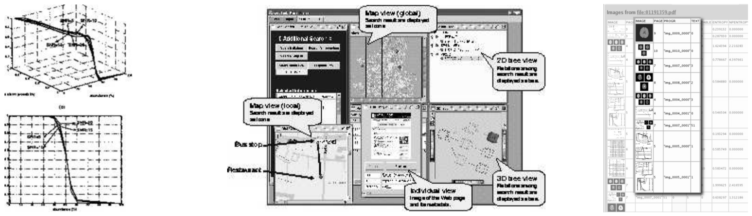
The vision of the Semantic Web has always encompassed not only the declarative r
resentation of data but also the development of intelligent agents that can consume
data and act upon their owner's behalf. For instance, agents have been proposec

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result xmlns:xsp="http://apache.org/xsp" xmlns:xsp-session="http://apache.org/
  xmlns:xscript="http://apache.org/xsp/xscript/1.0" xmlns:soap="http://apache.
  request="http://apache.org/xsp/request/2.0" xmlns:SOAP-ENC="http://schema
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w
  xsi:type="xsd:string">
- <LingAnnotations>
  - <ExtractedData
      Entry_level_url="http://wp3.xrce.xerox.com:8888/DS/store/trinity.dit.unitr
      filename=32980182.pdf&cocoon-view=xml" Entry_level_ids="[ [ N1006A]]" t
      data_id="0">
    - <entity>
        <author>Luke McDowell</author>
      </entity>
    </ExtractedData>
  - <ExtractedData
      Entry_level_url="http://wp3.xrce.xerox.com:8888/DS/store/trinity.dit.unitr
      filename=32980182.pdf&cocoon-view=xml" Entry_level_ids="[ [ N1006A]]" t
      data_id="1">
    - <entity>
        <author>Oren Etzioni</author>
      </entity>
    </ExtractedData>
  - <ExtractedData
      Entry_level_url="http://wp3.xrce.xerox.com:8888/DS/store/trinity.dit.unitr
      filename=32980182.pdf&cocoon-view=xml" Entry_level_ids="[ [ N1006A]]" t
      data_id="2">
    - <entity>
        <author>Alon Halevy</author>
      </entity>
    </ExtractedData>
```

Fig. 2. XML annotations derived from a remote PDF file through harmonisation and semantic annotation

Languages other than English are also dealt with in the pipeline. In particular, a semantic annotation component for Italian texts (VISTA, ”VIKEF Italian SemanTic Annotator”) has been developed, which takes harmonized documents as input and produces an XML output compliant to the Extraction Schema. Semantic annotation is performed by the AnIta system [12] [13], a robust parsing architecture for the analysis of Italian texts which was augmented with functionalities of named entity recognition and categorization to cope with the specific requirements of semantic annotation in the scientific scenario. In addition, an existing ontology learning tool (T2K, which stands for ”Text-to-Knowledge”), combining linguistic and statistical techniques, is being customised to extend and tune pre-existing ontologies for the VIKEF trade fair scenario. T2K [14] performs the extraction of domain terminology from texts (including both single and complex terms), structures the set of acquired terms into taxonomical chains (reconstructed from their internal linguistic structure) and into sets of semantically related terms (i.e. potential synonyms) on the basis of distributionally-based similarity measures [15].

The iconic part of the documents is managed by a specialized extraction service. The focus of image analysis has been directed to the broad categorization of figures within scientific papers into groups that are not specific to any scientific domain but are rather of *horizontal* nature. Non textual material is automatically extracted and described by low-level descriptors that can be effectively used by a classifier system [16] [17][18]. The resulting description is provided in XML format for easy distribution and reuse. The same information is exploited by a rule based system that classifies images into plots, tables, charts, annotated images, framed text and the like (figure 3). This two step approach to image classification increases system efficiency as the time consuming step of low level image annotation may be performed once while refined classification can be obtained by deploying a new classifier stage. The text overlapping pictures is considered as image annotation as well as the corresponding captions (if any). This textual information can be exploited to improve search and browse support in the scientific domain in several ways. Tables, charts, plots, and plots annotated with figures provide increasingly richer information. Responding to a typical search such as *"Give me the papers where reaction speed is accurately reported"* would benefit from sorting relevant paper according to the way they present the data requested, starting from the most informative ones. Image comparison techniques can also be used to spot use (or abuse) of specific images - *Give me all the paper where image segmentation has been tested on this specific example* - or to restrict browsing to papers whose image have specific visual qualities - *I would like oncology reports where image are presented in false colors for increased discriminability*.


Map view (global)
Search results are displayed as icons
2D tree view
Map view (local)
Search results are displayed as icons
Bus stop
Restaurant
Individual view
Image of the Web page and its metadata.
3D tree view

```
<image>
 <class>plot</class>
 <annotation>na</annotation>
</image>
```

(A)

```
<image>
 <class>annotatedImageo</class>
 <annotation>full</annotation>
</image>
```

(B)

(C)

Fig. 3. Examples of images that are automatically categorized (A and B) followed by an example of image filtering (C) based on classification results. The lighter (background) image in C presents document images in appearance order while the highlighted one filters images removing tables/equation (automatically detected) and sorts remaining images according to their visual *richness*.

The variety of the information extraction approaches described above already shows the usefulness of the definition of an annotation pipeline and an agreement of well-defined intermediate formats like e.g. the Extraction Schema. Such unifying schemata have proven a good medium for combining different extraction processes. The extracted data, obtained in this step, is then transformed into a logical annotation schema (LAS) which will be explained in the next section. By providing a framework in which several approaches can be combined, the power user (the user in charge of setting up an extraction pipeline) has user friendly tools to configure and ensemble a new pipeline out of pre-configured (lower level) processes.

2.5 From Linguistic Entities to Logical Entities

The Logical Annotation Schema (LAS) is used to represent the information produced by the Information Extraction Process (see Section 2.4) in a way that it can be refined, adding new information (e.g. linking entity and relation types to ontology concepts and properties) and aggregating existing data. This representation is a step further in enabling the process of adding semantics to the extracted data. The linkage to the ontology can be performed in different ways, one is to specify the known correspondences between the extracted types and the corresponding ontological element when the extraction process is configured. In these cases this information is just added to the LAS and passed to the next steps. Other possiblities are the use of existing semantic elicitation techniques (like e.g. [19]) for finding the correspondences between the extracted types and the ontological elements.

The initial transformation of the Extraction results to LAS contains the extraction results.

The LAS includes five major types of elements: a) the elements representing information about the analyzed collection, b) the elements representing information about every content resource inside the collection, c) the elements representing annotation information (e.g. information about each extracted entity and relation), d) the elements representing logical entities, i.e. entities that were detected to represent the same instance, and e) the elements representing logical relations, i.e. representation of relations between logical entities recognized by analyzing the underlying extracted entities.

Since the schema is quite large for being comprehensive, just the most important parts are described in this paper:

Annotation Information represent specific occurrences of an extracted entity and contains elements representing information about the type and the value of the entity (e.g. the string in a textual content resource). It contains also an ID, so that it can be referenced, and a pointer to the content resource element where it was identified, the URL of the entity (so that it can be visualized and highlighted in a browser), and a reference to the logical entity where it is aggregated among others.

Logical Entities pull together extracted entities that represent the same object. Annotation Information representing the same ”real world” object are grouped together, so that all of them are represented by a single Logical Entity (e.g the same person being cited in several places inside one paper). Each Logical Entity contains the type of the entity, a value as a representative of the aggregated Annotation Information, an identifier for referencing each Logical Entity, and optional representations of the corresponding ontological concept, the URI of an existing instance in the *SRN* that stands for the instance represented by the Logical Entity (e.g. the extraction process detects the name of a person that is already represented in the *SRN* and the URI that represents this instance is added to the LAS file).

Relation Information represent a specific occurrence of a relation between two Annotations. Analogous to the Annotation Information, it contains elements for representing the identifier of the corresponding Logical Relation, the reference to the content resource where it is contained, a URL for accessing the occurrence of the relation in a browser, the type of linguistic relation that was detected, and references to the source and target Annotation(s). At the moment, only binary relations are considered. An extension to n-ary relations is planned.

Logical Relations represent relations between Logical Entities. Analogously to the Logical Entities an identifier, a relation type obtained from the linguistic analysis and an optional ontology property are represented. Additionally, it contains a reference to the origin (subject) Logical Entity and to the target (object) Logical Entity, in order to depict the two Logical Entities that are involved in the relation.

The LAS is iteratively refined and new information is added in each iteration. Using this approach we try to use already known information in order to narrow the gap between the extracted information and its semantic representation. After the refinement phase of the LAS, all information about the ontologies and the instances that could be detected are contained in the resulting LAS compliant XML document. This refined LAS is the input for the RDF transformation process that will be depicted in the next section.

2.6 Producing the semantic layer

The LAS, as it was described in the previous section, provides the starting point of a process that produces a collection of RDF statements about the content extracted from a given collection of document.

In the approach presented in this paper, part of the problems of producing a semantic representation from the outcome of the information extraction problems have been addressed in the phase of creation and refinement of the LAS. Indeed, the LAS can be viewed as the result of two different processes:

- on the one hand, the LAS is obtained by mapping annotations which identify Named Entities into a suitable logical entity of LAS (and, analogously, for relations);
- on the other hand, the LAS is enriched with pointers to ontological knowledge. For example, as we said in Section 2.5, a linguistic entity type like PERSON can be linked to the URI of the corresponding concept in an ontology, and any entity can be linked to the URI of an ontology's instance.

The outcome of these two processes is a filled instance of a LAS which is already quite rich from a semantic point of view, but is expressed in a format (a plain XML file) which does not make its semantic content explicit. Therefore, the next step in the Pipeline we are describing is a conversion of the LAS into a collection of RDF statements which make explicit the statements which are implicitly made in the LAS.

The RDF statements we can produce from a LAS can be divided into two main classes:

- the first class contains statements which refer to the content of the resources. Examples are: the logical entity with URI U_1 is the author of the paper with URI U_2, the logical entity with URI U_1 is affiliated to a logical entity with URI U_3, U_3 is an organization, the title of the paper with URI U_2 is 'ABC', and so on;

- the second class contains statements which connect some content (e.g. a Named Entity, and indirectly a logical entity) to the location of a ContentResource where that entity was detected (i.e. the position of the document where the Named Entity was found).

The two classes of statements serve two different, but equally important, purposes. Indeed, the first class is the virtual layer in which knowledge extracted from some content resource is represented; such a layer can be used to implement a large variety of services, including semantic-based search, reasoning, integration from different sources, and so on. The second class of RDF statements anchor such a virtual layer to the sources themselves; this information is essential when we want to implement services which need to go back directly to the sources, e.g. semantic-enabled browsing of document collections, retrieval of pictures, and so on.

Since all the relevant information for producing the two types of RDF statements is already present in the LAS, the creation of the RDF collection does not present significant conceptual or technical difficulties. Indeed, in the current version of the Pipeline, it has been implemented as a standard XSL Transformation, which takes in input the LAS itself and produces what we call a temporary Semantic Resource Network (SRN) (see below for an explanation of why we say “temporary”).

To have an idea of the amount of information we can currently extract from a collection of documents, consider the following figures from two preliminary runs of the system:

- starting from a single document, we were able to produce a LAS with 64 logical entities, 180 entity annotations, and 1078 RDF triples;
- starting from other 5 documents, we were able to produce a LAS with 456 logical entities, 6600 entity annotations, and 34.307 RDF triples.

The main effort in the design of the transformation stylesheet is to identify types of the statements which we want to produce with the transformation, and to write the corresponding transformation rules. However, in the future, it may be that we produce RDF statements which cannot be obtained by simple transformations from the LAS.

2.7 Semantic Integration

We use the information extraction process to extend an *SRN* that has already been created by Pipeline I and/or the application of Pipeline II for other content collections. The temporary SRN (tempSRN) obtained by the previous step of the pipeline process can contain RDF triples that refer to already existing instances in the *SRN*. The Semantic Integration is the integration of statements obtained from newly extracted information into the existing set of statements of an *SRN*.

A successful integration requires to detect that two statements refer to the same instance even if different URIs are used to represent it. One approach for detecting this is to analyze the properties attached to instances of the same concept (in the SRNs to be integrated) and to search for overlaps of properties that are known to hold unique values (at least in the considered domain). It can then be assumed that both URIs refer to the same instance if the values of the compared properties give a match. This approach of identifying overlapping instances in a post-processing step is complemented by an approach to employ information of existing instances in the earlier phases of the pipeline. For this purpose, we included the possibility to specify the instance URI already at extraction time or when refining the LAS (see previous subsections). This can be exploited in the LAS to RDF transformation step to already use the correct instance URIs to create the statements.

In the tempSRN, the URIs of instances that are detected to be already present in the *SRN*, are replaced with the respective URIs used in the *SRN*. After this processing step, duplicate triples are removed and the remaining statements are appended to the *SRN* enhacing the existing *SRN* with the newly obtained semantic information.

3 Prototype Overview

This section presents an overall view of our implemented prototype, explaining the most important steps in the process.

The VIKEF prototype provides options for executing the different components in the pipeline process:

1. The content Harmonization component for transforming the content sources in a representation independent format.
2. The Information Extraction component responsible for recognizing and extracting entities and relations from the content sources.
3. The mapping components for defining and executing mappings from the Information Extraction output format to the LAS format.
4. The LAS refinement component, for enriching the LAS with ontological information and for aggregating different references to one entity/relation into one logical entity/relation.
5. The RDF transformation components for the generation of RDF statements based on the contents of the LAS compliant data.
6. The Information Integration component that receives as input the newly generated RDF statements and integrates them in the (possibly) already existing *SRN*.

The remaining of the paper describes some services that make use of the available semantic information obtained in this pipeline and some future directions we are interested in exploring.

4 Semantic Content Navigation

The Semantic representation obtained as a final outcome of the Pipeline can be used to implement a large number of community services. Some of these services may rely exclusively on the virtual layer of information built from one or more collection of resources; an example may be a query engine which allow users to ask queries on the content of documents which require some reasoning on the RDF triples stored in a *SRN*. However, here we'd like to briefly discuss another kind of services, which exploit the mixture of abstract and physical information stored in a *SRN*, and that we call semantic content navigation.

Semantic content navigation is a possible realization of the idea of a Semantic Web browser. Indeed, in this scenario the content of documents can be automatically extracted, represented in a virtual layer and reasoned about, or portions of text in a document can be highlighted to signal that it has been recognized as a relevant entity or property. In addition, the combination of logical and physical information which is produced as an output of the Pipeline would allow users to navigate from document to document (not only HTML documents, but also documents in formats like PDF) by following the logical links which are associated to physical portions of the documents themselves.

Imagine a more advanced service, for example, that the system has recognized that some string in a PDF document corresponds to the name of its author, and that such an author is an instance of a concept defined in one or more ontologies. Suppose then that some SRN stores statements about this entity, for example that he or she works for a given University, or is co-author of another paper with another researcher. This information (which is derived from the virtual space of information) can be combined with information about the physical location (e.g. positions in documents) from which is was extracted to implement a new type of navigation, where traversing a link would be a mix of using knowledge about an entity and at the same time being referred to a precise point of the document from which this information was extracted. We call such a service the Semantic Infusion service and it will be described in a next paper due to space constraints.

Of course, the type of navigation which is allowed depends on the domain of application. In a scenario where we deal with scientific papers, it can be used to browse through document following the history of a new idea, or the discussion about some research issue, or to trace the contributions of a given author. Another scenario which we are investigating has to do with the organization of trade fairs. Here the same approach might be used for example to allow the visitors to browse through a large number of catalogs in search of similar items, or different items produced by the same maker, or tracing to trace the history of a product through different fairs in a series.

5 Related Work

The Semantic Resource Networks that we construct as part of developing semantic-enabled services in VIKEF are knowledge networks on top of the underlying content collections. They are thus related to the construction of Topic Maps [20, 21], that also act as knowledge networks (or maps) for the description and navigation of content collections. However, since VIKEF is part of the Semantic Web activity and since the SRN are targeted towards interpretation by software (as well as by humans), we decided to use the Resource Description Framework (RDF, [7]) and not the Topic Map standard for the representation of the SRN.

Over the years a number of useful tools have been developed to help with the manual or semi-automatic markup of Web documents including SHOE [22], Annotea [23] and CREAM [24]. Closer to the annotation approach undertaken in VIKEF are information extraction systems from the area of Natural Language Processing (NLP) such as the General Architecture for Text Engineering (GATE) [25] and the Unstructured Information Management Architecture (UIMA) [26]. GATE builds complex processing pipelines from modularised language resources (e.g. documents, corpora, lexicons etc.) and software components (e.g. tokenisers, lemmatizers, parsers etc.). The processing resources use a central database in order to modify existing annotations in the database or generate new annotations that comply with the TIPSTER annotation model [27]. Support for ontology-based annotation is provided by an ontology gazetteer that links the classes of a specified ontology to the annotations created by a lexicon lookup component. UIMA's architecture consists of *analysis engines* that act as larger blocks of annotation modules and other analysis engines. The sharing and processing of annotations amongst annotators is facilitated by object-based containers that manage typed objects with properties and values. To analyse an entire collection of documents, UIMA uses *collection readers* that iterate through the document collection in order to initialize the annotation containers for further analysis and *consumer modules* that process the annotations in order to perform tasks such as, for example, populating a relational database or indexing the text collection. VIKEF differs from these systems in (i) the conceptual distinction between annotations for linguistic entities and those for logical entities, (ii) the explicit representation of relation annotations, and (iii) the use of relation annotations to produce RDF statements that implement a virtual layer of services for search, reasoning and information integration.

Tools that use the semantic annotations for browsing and navigation include the Magpie/ASPL [28] semantic browser and Flink system [29] for social networks. Magpie is a tool that aids users in learning tasks such as surveying or interpreting scientific texts using a domain ontology to dynamically annotate pages and highlight phrases associated with ontology classes. Specific services are available for each class and these range from services that provide explanatory material (e.g. "Explain concept A") to services that provide relational information about an instance (e.g. "Shares institution with", "People active in"). Additional services link to external sources such as CiteSeer or ACM to provide extra information that cannot be extracted directly from the text (e.g. "Find Co-citing community", "Find in ACM library").

The Friend of a Friend (FOAF) [9] project is an interesting example of how information describing personal identity, work and affiliations can be aggregated and interlinked over time. This idea is adopted by the Flink system which employs semantic analysis of personal Web pages, e-mails and publication archives to generate "who-is-who" profiles of researchers in the Semantic Web community. Navigation from a personal profile is performed by hyperlinking the names of co-authors, e-mail recipients or other affiliates. The system identifies links between researchers which are visualized in graphical form to provide a fish-eye view of the network. The research interests in the profiles can also be used to generate ontologies of Semantic Web topics.

Of course the work in the VIKEF project is related to research in the different research areas that were identified in the introduction including research in content harmonization, extraction of semantic information from content objects, etc. However,

[9] http://www.foaf-project.org

these topics are not in the focus of this paper and a discussion of all the related work goes beyond the scope of this paper.

6 Conclusions and Future Work

In this paper we presented a general approach for enabling a complete knowledge supply chain from content sources (documents, multimedia repositories, etc.) to ontologies, and to support the runtime access from the semantic layer back to the content sources. This process, as we said, raises several difficult issues, including the automated connection from linguistically extracted entities, entity types, relations to ontological objects (instances, classes, properties), the detection of duplications (statements that already exist in the semantic layer) and the merging of the newly extracted information with the already existing statements, taking into account the restrictions stated in the ontology and the truth of the information in different points in time (truth value depending on context).

Our future aims at addressing these issues and proposing general purpose solutions. First, we will work on more advanced methods for the refinement of the LAS and the integration of information; we will consider using statistical and machine learning approaches. Second, we will improve the automatic recognition of attribute and relation relevance for duplicate detection, and develop an iterative duplicate detection approach of related instances. Another important research area has to do with the contextualization of RDF statements and repositories; the idea is that adding contextual information to collections of RDF statements may help in making decisions, for example solving potential conflicts (e.g. between two apparently contradictory statements, when they implicitly refer to different points in time), and in merging independent collections of statements (some of these ideas are preliminary discussed in [30]).

Finally, there are good reasons to believe that we will quickly find ourselves in a scenario where multiple (and typically independent) repositories of RDF statements will become available. One reason is the potentially huge number of statements that can be generated from a relatively small set of initial sources (see Section 2.6); this may require a physical partition of the repository. The second, and conceptually more relevant reason, is that these different repositories may not only be a partition of a logically single repository, but may be the outcome of independent processes, potentially highly heterogeneous from a semantic point of view (e.g. if each process adopts different background ontologies for giving semantics to RDF statements). This means that we will need to define methods not only for distributing queries across physically partitioned knowledge bases, but also for using existing mappings across ontologies to retrieve and integrate statements which are not already “aligned”. We plan to do this by exploiting the rich work on ontology matching and alignment which is under development in the Semantic Web community (see e.g. Deliverables of WP2.2 in the Knowledge Web network of excellence[10]).

The extraction methods are constantly being enhanced and different options tested to make the results more accurate.

[10] See `http://knowledgeweb.semanticweb.org/` for public documentation.

Acknowledgements The work described in this paper has been partly funded by the European Commission through grant to the project VIKEF under the number IST-507173. We would like to thank the other members of the VIKEF project team for the numerous discussions and contributions to the idea of the Core VIKEF Infrastructure and the VIKEF annotation pipeline presented in this paper.

References

1. World Wide Web Consortium: (Ontology Web Language) `http://www.w3.org/TR/owl-semantics/`.
2. Stecher, R., Niederée, C.: Ontology fitness - supporting ontology quality beyond logical consistency. In: Formal Ontologies Meet Industry Workshop (FOMI), June 9-10, Italy. (2005)
3. Xindong Wu, L.J., ed.: Ontological Engineering. Springer-Verlag London Limited (2004)
4. Advanced Knowledge Technologies: (AKT Reference Ontology) `http://www.aktors.org/publications/ontology`.
5. IEEE Learning Technology Standards Committee: (IEEE 1484.12.1-2002 - Learning Object Metadata) `http://ieeeltsc.org/wg12LOM`.
6. Advanced Distributed Learning: (Sharable Content Object Reference Model) `http://www.adlnet.org/scorm/index.cfm`.
7. Klyne, G., Carroll, J.J.: Resource Description Framework (RDF): Concepts and Abstract Syntax W3C Recommendation. (2004) `http://www.w3.org/TR/rdf-concepts/`.
8. Déjean, H., Meunier, J.L.: A system for converting pdf documents into structured xml format. In: Proceedings of the Workshop on Document Analysis Systems, Nelson, NZ (2006)
9. Meunier, J.L.: Optimized xy-cut for determining a page reading order. In: ICDAR05: Proceedings of the 8th International Conference on Document Analysis and Recognition, Seoul, Korea (2005)
10. Chidlovskii, B., Fuselier, J.: A probabilistic learning method for xml annotation of document. In: IJCAI05: Proceedings of the IJCAI, 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland (2005)
11. Aït-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond shallowness: incremental deep parsing. Natural Language Engineering **8** (2002) 121–144
12. Bartolini, R., Lenci, A., Montemagni, S., Pirrelli, V.: Grammar and lexicon in the robust parsing of italian: Towards a non-nave interplay. In: COLING 2002 Workshop on 'Grammar Engineering and Evaluation', Nankang, Taipei, Taiwan. (2002)
13. Bartolini, R., Lenci, A., Montemagni, S., Pirrelli, V.: Hybrid constraints for robust parsing: First experiments and evaluation. In: Fourth International Conference on Language Resources and Evaluation (LREC 2004), Portugal. (2004)
14. Bartolini, R., Giorgetti, D., Lenci, A., Montemagni, S., Pirrelli, V.: Automatic incremental term acquisition from domain corpora. In: 7th International conference on Terminology and Knowledge Engineering (TKE-2005), Denmark. (2005)
15. Allegrini, P., Montemagni, S., Pirrelli, V.: Learning word clusters from data types. In: COLING-2000, Saarbrücken, Germany. (2000)
16. Sauvola, J., Pietikäinen, M.: Page segmentation and classification using fast feature extraction and connectivity analysis. In: ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 2), Washington, DC, USA, IEEE Computer Society (1995) 1127
17. Cullen, J.F., Hull, J.J., Hart, P.E.: Document image database retrieval and browsing using texture analysis. In: ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition, Washington, DC, USA, IEEE Computer Society (1997) 718–721

18. Hu, J., Bagga, A.: Functionality-based web image categorization. In: WWW (Posters). (2003)
19. Bouquet, P., Serafini, L., Zanobini, S.: Peer-to-peer semantic coordination. Journal of Web Semantics **2** (2005)
20. Biezunski, M., Bryan, M., Newcomb, S.: Iso/iec fcd 13250:1999 - topic maps (1999)
21. Pepper, S.: Navigating haystacks and discovering needles - introducing the new topic map standard. Markup Languages: Theory and Practice **1** (1999) 41–68
22. Heflin, H., Hendler, J.: Searching the web with SHOE. In: Artificial Intelligence for Web Search, Papers from the AAAI Workshop, AAAI Press (2000) 35–40
23. Kahan, J., Koivunen, M., Prud'Hommeaux, E., Swick, R.: Annotea: An Open RDF Infrastructure for Shared Web Annotations. In: Proceedings of the 10th International World Wide Web Conference, ACM Press (2001) 623–632
24. Handschuh, S., Staab, S.: Authoring and annotation of web pages in cream. In: Proceedings of the 11th International World Wide Web Conference, ACM Press (2002) 462–473
25. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics. (2002)
26. UIMA: An open, industrial-strength platform for unstructured information analysis and search. (2004) `http://www.alphaworks.ibm.com/tech/uima`.
27. Grishman, R., and the TIPSTER Phase III Contractors: TIPSTER Text Architecture Design, Version 3.1. (1998)
28. Dzbor, M., Domingue, J.: Magpie: supporting browsing and navigation on the semantic web. In: Proceedings of the 9th international conference on Intelligent User Interfaces, ACM Press (2004) 191–197
29. Mika, P.: Flink: Semantic web technology for the extraction and analysis of social networks. Web Semantics **3** (2005)
30. Bouquet, P., Serafini, L., Stoermer, H.: Contextualizing rdf knowledge bases. Proceedigs of the second Italian workshop on Semantic Web: Applications and perspectives (SWAP-2005) (2005) `http://www.ceur.org`.