Breaking the deadlock

Rinaldi, Fabio ; Kaljurand, K ; Dowdall, J ; Hess, M

Abstract: Many of the proposed approaches to the semantic web have a substantial drawback. They are all based on the idea that web pages (or more generally, resources), will contain semantic annotations that would allow remote agents to access them. However the problem of the creation of those annotations is seldom addressed. Manual creation of the annotations is not a feasible option, except in a few experimental cases. We propose an approach based on Language Processing techniques that addresses this issue, at least for textual resources (which still constitute the vast majority of the material available on the web). Documents are analized fully automatically and converted into a semantic annotation, which can then be stored together with the original documents. It is this annotation that constitutes the machine understandable resource that remote agents can query. A semi-automatic approach is also considered, in which the system suggests candidate annotations and the user simply has to approve or reject them. Advantages and drawbacks of both approaches are discussed.

DOI: https://doi.org/10.1007/b94348

Breaking the Deadlock

Fabio Rinaldi, Kaarel Kaljurand, James Dowdall, and Michael Hess

Institute of Computational Linguistics,
University of Zürich,
Winterthurerstrasse 190, CH-8057 Zürich,
Switzerland
{rinaldi}@cl.unizh.ch

Abstract. Many of the proposed approaches to the semantic web have a substantial drawback. They are all based on the idea that web pages (or more generally, resources), will contain semantic annotations that would allow remote agents to access them. However the problem of creating these annotations is seldom addressed. Manual creation of the annotations is not a feasible option, except in a few experimental cases.
We propose an approach based on Language Processing techniques that addresses this issue, at least for textual resources (which still constitute the vast majority of the material available on the web). Documents are analized fully automatically and converted into a semantic annotation, which can then be stored together with the original documents. It is this annotation that constitutes the machine understandable resource that remote agents can query. A semi-automatic approach is also considered, in which the system suggests candidate annotations and the user simply has to approve or reject them. Advantages and drawbacks of both approaches are discussed.

1 Introduction

The major purpose of activities in the Semantic Web area is to help users better locate, organize, and process content, irrespective of its physical location and of the way it is presented. Adding machine-understandable semantics to web resources will make them processable by software agents, and ultimately make them more useful to all of us.

There is a wealth of research efforts focusing on the foundations of the semantic web [8], and in particular on the problem of how to represent the semantic information carried by web resources (be they structured databases or unstructured natural language documents, or a combination of both). The XML-based Resources Description Framework [14] is the standardized Semantic Web language, however it is really meant for use by computers, not humans. The same applies to all the extensions that have been proposed, such as RDF Schema [2], which provides a basic type system for use in RDF models, or DAML+OIL [4], which provides a language with well-defined semantics for the specification of Ontologies.

However, there seems to be significantly less interest in the problem of how to help users in the transition from conventional web pages to richly annotated semantic web resources. The major barrier to a wider adoption of the Semantic Web proposals is a classic deadlock problem [11]. On the one hand, significant additional effort is required

to add semantic annotations to existing (or newly created) web resources, and people are not willing to pay this price until they can see a clear benefit for it. On the other hand, software agents that can reap the benefit of richer annotations will not be useful (and thus there will be fewer incentives to develop them) until a "critical mass" of semantically annotated web resources has been achieved.

Current efforts to tackle this problem seem to focus on the development of user-friendly editors for semantic annotations: details of XML/RDF should be hidden behind GUI authoring tools. Users do not need (and do not want) to get in contact with XML/RDF. However this approach defeats the purpose of the Semantic Web vision: to make the web more effective for users by making it machine-understandable. Instead it makes it less effective for users: by forcing them to add machine-level markup (albeit shielded by an effective GUI). Unless the users can see the real benefit, they will not be motivated to adopt such editors and be prepared to pay the price (in terms of additional effort that might be required).

The benefits of the semantic web should come for free to most of the users: semantic markup should be a by-product of normal computer use. There is a real need to lower the barrier of entry: the vast majority of the users cannot be expected to understand and use formal ontologies. In order to achieve interoperability between software agents, a lot of human understandability has been sacrificed: precise ontologies and formally defined semantics are foreign concepts to the average users.

As a very large proportion of existing web resources are represented by human-readable documentation, we believe that a possible way to break the deadlock mentioned above is to start using available information extraction tools to enrich the documents with automatically generated annotations. In this paper we propose an approach based on natural language processing (NLP) techniques, geared towards the creation of semantic annotations, starting from the available textual documents.

One of the motivations behind the semantic web movement was that computers are not powerful enough to process (and understand) natural language. Therefore machine understandable information should be added to web resources. This is still true: it would be unfeasible to process the enormous amounts of textual resources that are added to the web every day (let alone process all the existing web content). However, it is technically possible (and practically conceivable) to have specialised editors that process (in a transparent fashion) textual resources as the users publish them on the web, and add semantic annotations automatically extracted from the documents. In other words, the idea is to move the problem from the consumer of the information to the producer.

As Natural Language is the information access most users are comfortable with, we will also discuss possible ways to access the information encoded in the semantic annotations. Given a user question phrased in natural language, existing tools can convert it into the same kind of annotations as those stored in the documents. A new type of software agent (or search engine) might then be capable of retrieving those web pages whose annotations match those derived from the user question.

The approach presented in this paper is based on our previous work in the area of Question Answering, resulting in the ExtrAns system [22]. Specific research in the area of Question Answering has been promoted in the last few years in particular by the Question Answering track of the Text REtrieval Conference (TREC-QA) competi-

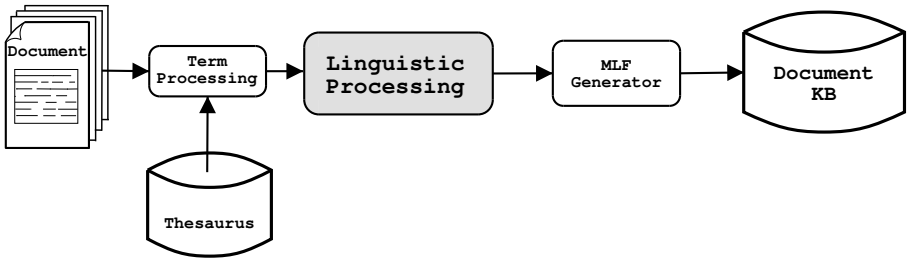

Fig. 1. Offline analysis of documents

tions [26]. ExtrAns uses a combination of robust natural language processing technology and dedicated terminology processing [19, 20] to create a Knowledge Base, containing a semantic representation for the propositional content of the documents [23]. Our research group has been working in the area of Question Answering for a few years, targeting different domains, such as the Aircraft Maintenance Manual (AMM) of a large aircraft [22] or a computer manual [15].

In a recently started EU project ("Parmenides") focusing on the integration of Information Extraction and Data Mining techniques, we aim at exploiting the work done in the ExtrAns system by moving from the system-specific semantic representation (Minimal Logical Forms) to a semantic representation based on W3C standards, like RDF. A secondary aim might be to explore possible synergies with the standardization effort of the ISO TC37/SC4 committe in the domain of linguistic annotations [21].

We will first briefly describe our past work resulting in the ExtrAns system (section 2), then describe the annotations that we aim at generating automatically in the Parmenides project (section 3). The following section (4) will describe in detail the approach that we propose in order to automatically create semantic annotation for textual web resources. Finally, in section (5) we explore advantages and disadvantages of the proposed methodologies, and describe our current work and suggestions for future development.

2 ExtrAns

In this section we briefly describe the linguistic processing performed in the ExtrAns systems, extended details can be found in [22]. An initial phase of syntactic analysis, based on the Link Grammar parser [24] is followed by a transformation of the dependency-based syntactic structures generated by the parser into a semantic representation based on Minimal Logical Forms, or MLFs [15]. As the name suggests, the MLF of a sentence does not attempt to encode the full semantics of the sentence. Currently the MLFs encode the semantic dependencies between the open-class words of the sentences (nouns, verbs, adjectives, and adverbs) plus prepositional phrases. The notation used has been designed to incrementally incorporate additional information

if needed. Thus, other modules of the NLP system can add new information without having to remove old information.

We have chosen a computationally intensive approach, which allows a deeper linguistic analysis to be performed, at the cost of higher processing time. Such costs are negligible in the case of a single sentence (like a user query) but become rapidly impractical in the case of the analysis of a large document set. The approach we take is to analyse all the documents in an off-line stage (see figure 1) and store a representation of their contents (the MLFs) in a Knowledge Base. In an on-line phase, the MLF which results from the analysis of the user query is matched in the KB against the stored representations, locating those MLFs that best answer the query. At this point the system can locate in the original documents the sentences from which the MLFs where generated (see figure 2).

One of the most serious problems that we have encountered in processing technical documentation is the syntactic ambiguity generated by multi-word units, in particular technical terms. Any generic parser, unless developed specifically for the domain at hand, will have serious problems dealing with them. On the one hand, it is likely that they contain tokens that do not correspond to any word in the parser's lexicon, on the other, their syntactic structure is highly ambiguous (alternative internal structures, as well as possible undesired combinations with neighbouring tokens). In fact, it is possible to show that, when all the terminology of the domain is available, a much more efficient approach is to pack the multi-word units into single lexical tokens prior to syntactical analysis [5]. In our case, such an approach brings a reduction in the complexity of parsing of almost 50%.

During the process described above, terms are gathered into WordNet style synsets and organized into a taxonomy. During the analysis of documents and queries, if a term belonging to a synset is identified, it is replaced by its synset identifier, which then allows retrieval using any other term in the same synset. This amounts to an implicit 'terminological normalization' for the domain, where the synset identifier can be taken as a reference to the 'concept' that each of the terms in the synset describe [10]. In this way any term contained in a user query is automatically mapped to all its variants.

When an answer cannot be located with the approach described so far, the system is capable of 'relaxing' the query, gradually expanding the set of acceptable answers. A first step consists of including hyponyms and hyperonyms of terms in the query. If the query extended with this ontological information fails to find an exact answer, the system returns the sentence (or set of sentences) whose MLF is semantically closest with the MLF of the question. Semantic closeness is measured here in terms of overlap of logical forms; the use of flat expressions for the MLFs allows for a quick computation of this overlap after unifying the variables of the question with those of the answer candidate. The current algorithm for approximate matching compares pairs of MLF predicates and returns 0 or 1 on the basis of whether the predicates unify or not. An alternative that is worth exploring is the use of ontological information to compute a measure based on the ontological distance between words, i.e. by exploring its shared information content [18].

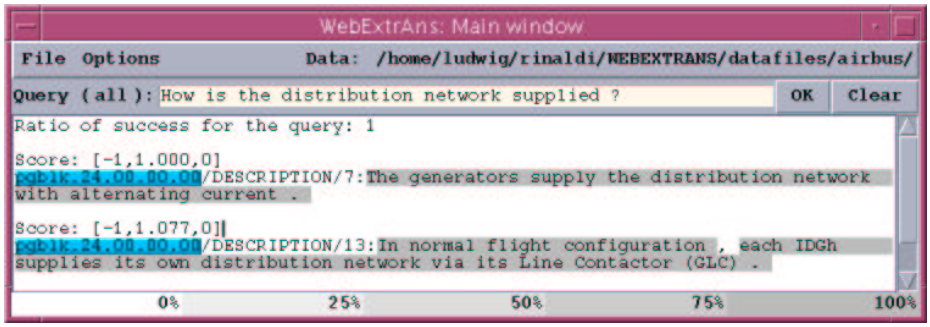

Fig. 2. Example of interaction with the system

3 Parmenides Annotations

It is by now widely accepted that some W3C standards (such as XML and RDF) provide a convenient and practical framework for the creation of field-specific markup languages (e.g. MathML, VoiceXML). However XML provides only a common "alphabet" for interchange among tools, the steps that need to be taken before there is any real sharing are still many (just as many human languages share the same alphabets, that does not mean that they can be mutually intelligible). A minimal approach is to create a common data model.

The existence of a common standard brings many other advantages, like the ability to automatically compare the results of different tools which provide the same functionality, from the very basic (e.g. tokenization) to the most complex (e.g. discourse representation). Some of the NIST-supported competitive evaluations (e.g. MUC) greatly benefited from the existence of scoring tools, which could automatically compare the results of each participant against a gold standard. Another clear benefit of agreed standards is that they will increase interoperability among different tools. It is not enough to have publicly available APIs to ensure that different tools can be integrated. In fact, if their representation languages (their "data vocabulary") are too divergent, no integration will be possible (or at least it will require a considerable mapping effort).

In this section we will briefly describe the XML-based annotation scheme proposed for the Parmenides project (for more details see [21]). In general terms the project is concerned with organisational knowledge management, specifically, by developing an ontology driven systematic approach to integrating the entire process of information gathering, processing and analysis. The annotation scheme is intended to work as the projects' *lingua franca*: all the modules will be required to accept as input and generate as output documents conformant to the (agreed) annotation scheme. The specification will be used to create data-level compatibility among all the tools involved in the project.

Each tool might choose to use or ignore part of the information defined by the markup: some information might not yet be available at a given stage of processing or might not be required by the next module. Facilities will be provided for filtering

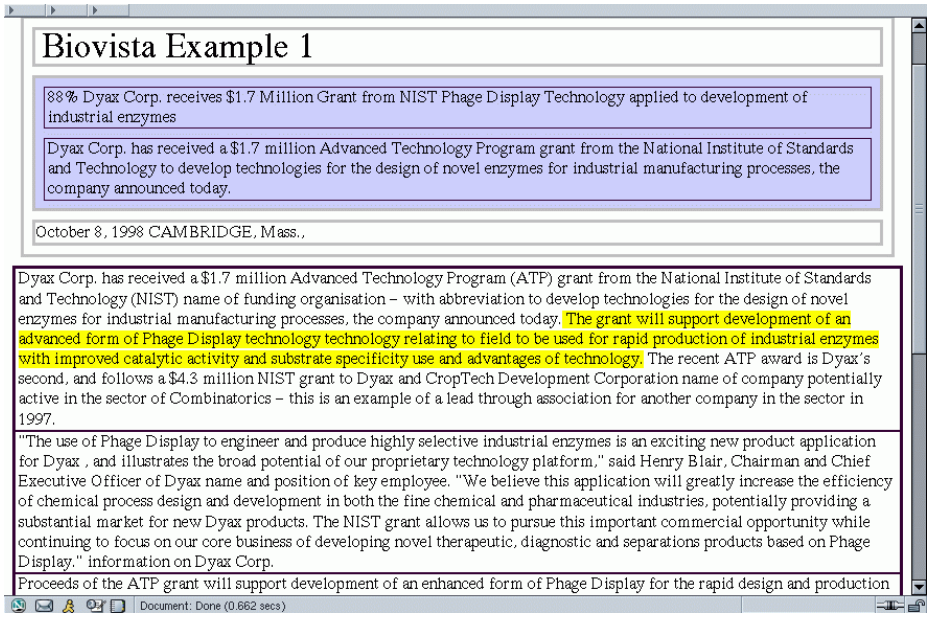

Fig. 3. Visualization of Structural Annotations

annotations according to a simple configuration file. This is in fact one of the advantages of using XML: many readily available off-the-shelf tools can be used for parsing and filtering the XML annotations, according to the needs of each module.

Parmenides aims at using consolidated Information Extraction techniques, such as Named Entity Extraction, and therefore this work builds upon well-known approaches, such as the Named Entity annotation scheme from MUC7 [3]. Other sources that have been considered include the GENIA tagset [7], TEI [25] and the GDA tagset [12]. Crucially, attention will also be paid to temporal annotations, with the aim of using extracted temporal information for detection of trends (using Data Mining techniques). Therefore we have investigated all the recently developed approaches to such a problem, and have decided for the adoption of the TERQAS tagset [9, 17]. The domain of interests (e.g. Biotechnology) are also expected to be terminology-rich and therefore require proper treatment of terminology.

There are currently three methods of viewing the document which offer differing ways to visualize the annotations. These are all based on transformation of the same XML source document, using XSLT and CSS (and some Javascript for visualization of attributes).

The set of Parmenides annotations is organized into three levels:

- **Structural Annotations**
- **Lexical Annotations**
- **Semantic Annotations**

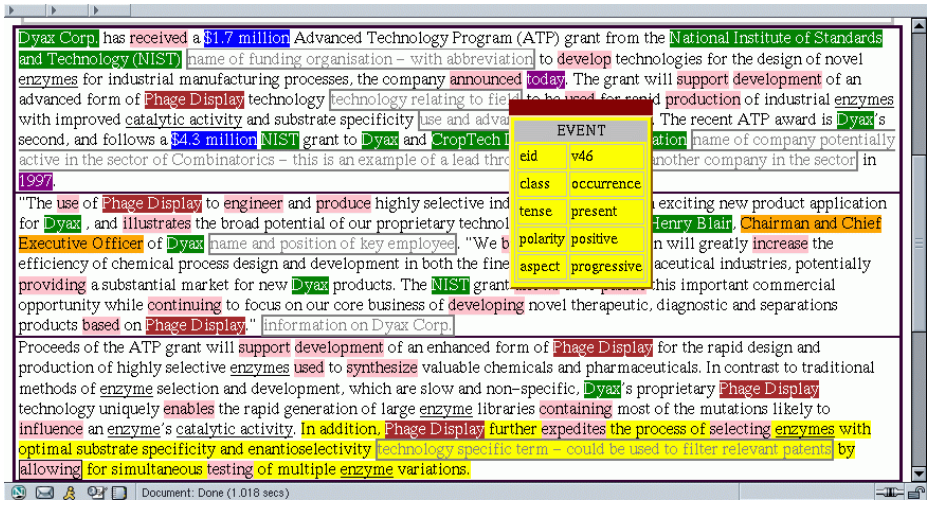

Fig. 4. Visualization of Lexical Annotations and their attributes

Structural annotations are used to define the physical structure of the document, it's organization into head and body, into sections, paragraphs and sentences. Lexical annotations identify lexical units that have some relevance for the Parmenides project. Semantic annotations are meant to represent the propositional content of the document (the "meaning"). While structural annotations apply to large text spans, lexical annotations aplly to smaller text spans (sub-sentence) and semantic annotations are not directly associated to a specific text span, however, they are linked to text units by co-referential identifiers. All annotations are required to have an unique ID and thus will be individually addressable, this allows semantic annotations to point to the lexical annotations to which they correspond. Semantic Annotations themselves are given a unique ID, and therefore can be elements of more complex annotations.

The structure of the documents will be marked using an intuitively appropriate scheme based on the TEI recomendations [25]. Broadly speaking, structural annotations are concerned with the organization of documents into sub-units, such as section, title, paragraphs and sentences. Figure (3) demonstrates the annotation visualization tool displaying the documents structure (using nested boxes).

Lexical Annotations are used to mark any text unit (smaller than a sentence), which can be of interest in Parmenides. They include (but are not limited to): Named Entities in the classical MUC sense, new domain-specific Named Entities, Terms, Temporal Expressions, Events. When visualizating the set of Lexical Tags in a given annotated document, clicking on specific tags displays the attribute values (see figure (4)).

The relations that exist between lexical entities are expressed through the semantic annotations. So lexically identified people can be linked to their organisation and job title, if this information is contained in the document (see figure (6)).

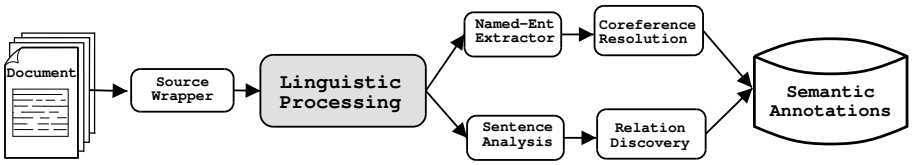

Fig. 5. From Documents to Semantic Annotations

4 From Documents to Semantic Annotations

In this section we will describe the approach taken in the Parmenides project towards the automatic creation of Semantic Annotation starting from existing documents (see figure 5). Documents are assumed to be gathered from a variety of sources, and thus will present different formats. The first step of processing is going to be a conversion from the source-specific document format to the agreed Parmenides format. This conversion is based on a set of source-specific wrappers [13], which transforms the original document into the XML structural annotations previously described.

The next step of processing involves addition of basic linguistic information: documents are tokenized, morphologically analyzed and tagged. At this stage sentence boundaries are also detected. This phase completes the creation of the structural annotation, going down to the lowest levels: the sentence and the token. An example of the resulting annotation can be seen below:

```
<tok id="t15" pos="CD" lem="@card@">
100
</tok>
<tok id="t16" pos="NNS" lem="calorie">
calories
</tok>
```

A Named Entity Extractor [1] is then used to detect persons, organizations, locations and numerical amounts. Together with a terminology extraction tool [6] this module creates the base Lexical Annotations.

At this stage however many different references to the same conceptual objects are not resolved. For instance different occurrences of the string “Dyax” will be considered as different lexical entities. The same problem would happen with cases like “Bill Clinton”, “Clinton”, “The former president of the United States”. An anaphora resolution module [16] is thus used to detect coreferent lexical entities. Simple string based match might suffice in some cases of named entities, however in more complex cases complex pronominal resolution algorithms are needed. This results in a set of equivalence classes (which contain coreferent lexical entities). Each class is then mapped into conceptual entities. The result of this process is illustrated in the upper half of figure 6.

A more thorough linguistic analysis is now needed to obtain relations among the entities discovered in the previous stage. First, each sentences is parsed using the Link Grammar parser, this is followed by a step of disambiguation and another step of

(pronominal) anaphora resolution, as described in detail in section 2. The result of this phase of analysis is a representations of the propositional content of the sentences, as minimal logical forms.

From the minimal logical forms, it is possible to detect relations, on the basis of axioms associated to lexical items, for instance the sentence "A works for B" gives rise to the logical form `work(A,B)`, while the sentence "A is employed by B" gives rise to the logical form `employ(B,A)`. Specific axioms associated to the lexical entries for "work" and "employ" allow to obtain the following mappings:

```
work(A,B)   -> worksFor(A,B)
employ(B,A) -> worksFor(A,B)
```

Now what is left to do is only to transform the results obtained into a standard formalism. The Conceptual Entities obtained after co-reference resolution can be mapped directly into RDF resources, as illustrated below.

```
<Organization rdf:ID="obj1"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns="http://www.parmenides.org/ontology/organization#"
      xml:base="http://www.parmenides.org/docbase/pardoc12566">
  <name>Dyax Corp</name>
  <activity>Biotechnology</activity>
</Organization>

<Organization rdf:ID="obj2"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns="http://www.parmenides.org/ontology/organization#"
      xml:base="http://www.parmenides.org/docbase/pardoc12566">
  <name>NIST</name>
  <activity>Government Agency</activity>
</Organization>
```

Relations can then be added to them as attributes, for instance to the RDF resource for "obj5" (Charles R. Wescott), it is possible to add the attribute "worksFor", as illustrate below.

```
<Person rdf:ID="obj5"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://www.parmenides.org/ontology/person#"
    xml:base="http://www.parmenides.org/docbase/pardoc12566">
 <name>Charles R. Wescott</name>
 <role>Senior Scientist</role>
 <worksFor rdf:resource=
   "http://www.parmenides.org/docbase/pardoc12566#obj1"/>
</Person>
```

The RDF annotations obtained with the process described above can be added automatically to documents being published on the web (in a totally transparent fashion).

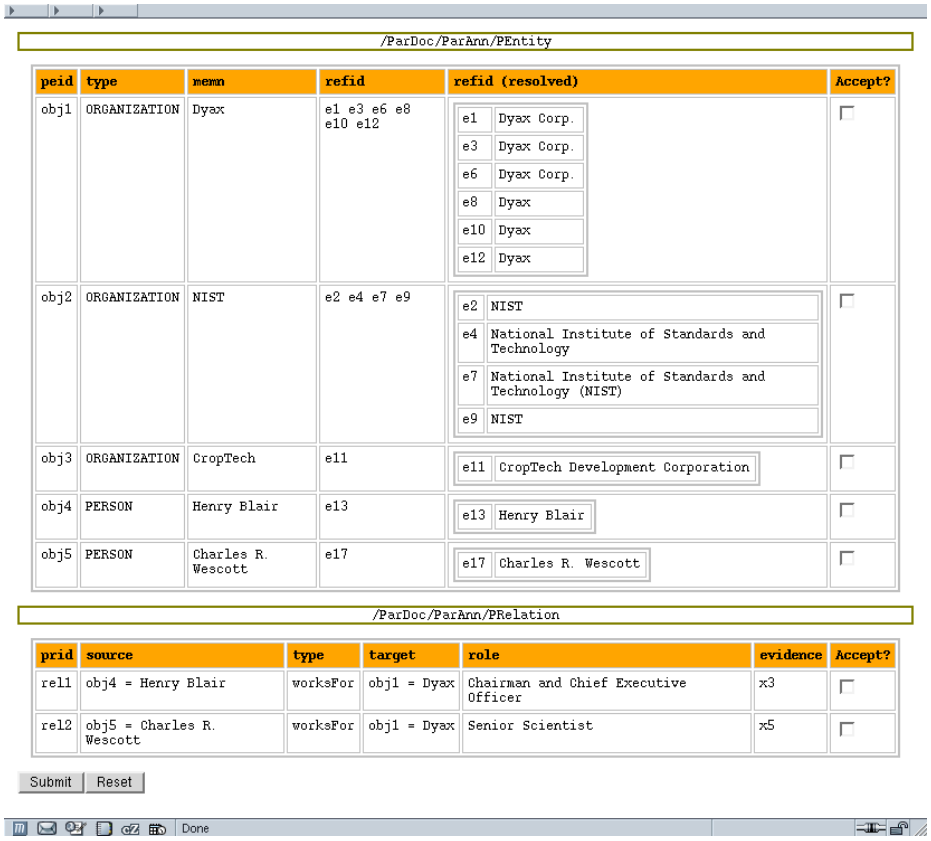

Fig. 6. Visualization of Semantic Annotations

In this way, they will become immediately accessible to automated software agents, which will be able to make more meaningful access to the document to which they are associated.

5 Current and Future Work

The method described in the previous section can work in an unsupervised manner, and help populate the semantic web with initial RDF resources. However, the NLP tools that are used to create them, might introduce various elements of errors, thus potentially decreasing the value of the resulting annotations. Missing annotations might be considered an acceptable price to pay, however conceptually wrong annotations might introduce dangerous contradictory information. In any case, the annotations, even if not always 100% reliable, might help in creating the “critical mass” that is so desperately needed in order to kick-start practical deployment of Semantic Web formalism and tools.

We are also considering a partially revised approach, in which a user can approve or reject the annotations suggested by the system. We have developed a simple graphical interface (based on XSLT transformations of the original XML documents and annotations), that allow a user to inspect the proposed resources and either accept or reject them. Consider again figure 6, what so far we have not explained is that this representation is also a web form, where the user can accept or reject individual objects and relations suggested by the system. The input from the user is processed by a conventional form processing script, which will then add to the published document only the annotations approved by the user. We think this approach would be particularly helpful in avoiding the introduction of contradictory or false knowledge, without posing too great a burden on the user.

A further (planned) development, would allow detailed editing of the resources within the same browser page. Clearly the users can in any case inspect the annotations with a conventional XML editor, but we believe that such an approach would not be feasible for large quantities of documents or for non-experienced users. We would fall back in the loop that we described at the beginning of the paper. A simple user-friendly interface like the one shown in the figure (possibly extended with more powerful editing capabilities) might provide a significant boost in allowing users to create RDF resources in a semi-automated fashion.

An aspect that we have not explored so far (but which is within our future targets) is the generalisation of resources from one document to a collection. All the resources described with the methodology illustrated in this paper are document specific, thus when we talk of *"Dyax"*, we should really say, the company *"Dyax as mentioned in the Parmenides document 12566"*, which might not be the same as the company *"Dyax"* mentioned in another document. So the complete reference to a particular instance of Dyax could be:

`http://www.parmenides.org/docbase/pardoc12566#obj1`

An aggregator tool should be able to detect the existence of the same company within different documents, and thus create a new, document-independent resource, to which all the individual mentions of "Dyax" in different documents point to, such as:

`http://www.parmenides.org/organization#Dyax`

Finally, a very natural extension of the work described here is the use of NL also for querying. This is in the spirit of our original ExtrAns system (as described in section 2), which was developed specifically for that purpose. Although this is not a direct target of the Parmenides project, we believe that the very same techniques can be extended to the Semantic Web area. We are actively looking for a chance to explore this very intriguing idea.

6 Conclusion

Despite the still experimental level of the current implementation, we are confident that the ideas described in this paper provide a powerful (and extremely useful) contribution to the future developments of the Semantic Web.

We are certain that we will witness in the near future a deeper convergence of the Semantic Web and the Natural Language Processing communities, towards the common goal of easing the information access bottleneck to web resources.

Acknowledgments

The originalExtrAns project (1996-2000) was funded by the Swiss National Science Foundation (contracts 1214-45448.95 and 1213-53704.98). Later work on the AMM manual (2000-2002) was privately funded by the Gebert Rüf Foundation (contract GRS-043/98). The Parmenides project is funded by the European Commission (contract No. IST-2001-39023) and by the Swiss Federal Office for Education and Science (BBW/ OFES).

The authors wish to thank all the Parmenides partners for helpful comments and insights. Special thanks to Biovista (http://www.biovista.com/) for their contribution to this work in supplying sample data and domain specific knowledge relating to corporate intelligence in biotechnology. Any remaining errors are the sole responsibility of the listed authors.

References

1. William J Black, Fabio Rinaldi, and David Mowatt. FACILE: Description of the NE system used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, 1998.
2. Dan Brickley and R.V. Guha. RDF vocabulary description language 1.0: RDF Schema. Technical report, W3C working draft, World Wide Web Consortium, April 2002. A reference for RDFS.
3. Nancy Chinchor. MUC-7 Named Entity Task Definition, Version 3.5, 1997. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html.
4. DAML+OIL, 2001. http://www.daml.org/.
5. James Dowdall, Michael Hess, Neeme Kahusk, Kaarel Kaljurand, Mare Koit, Fabio Rinaldi, and Kadri Vider. Technical terminology as a critical resource. In *International Conference on Language Resources and Evaluations (LREC-2002), Las Palmas*, pages 1897–1903, 29–31 May 2002. [1]
6. Katerina T Frantzi and Sophia Ananiadou. The C/NC value domain inpedented method for multi-word term extraction. *Journal of Natural Language Processing*, 6(3):145–180, 1999.
7. GENIA. Genia project home page, 2003. http://www-tsujii.is.s.u-tokyo.ac.jp/~genia.
8. Nicola Guarino. Formal ontologies in information systems. In N. Guarino, editor, *Proceedings of FOIS'98*, pages 3–15, Trento, June 1998. IOS Presss, Amsterdam.
9. Bob Ingria and James Pustejovsky. TimeML Specification 1.0 (internal version 3.0.9), July 2002. http://www.cs.brandeis.edu/%7Ejamesp/arda /time/documentation/TimeML-Draft3.0.9.html.
10. Kyo Kageura. *The Dynamics of Terminology, A descriptive theory of term formation and terminological growth*. Terminology and Lexicography, Research and Practice. John Benjamins Publishing, 2002.
11. Boris Katz, Jimmy Lin, and Dennis Quan. Natural language annotations for the semantic web. In *Proceedings of the International Conference on Ontologies, Databases, and Application of Semantics (ODBASE2002)*, October 2002.

[1] Available at http://www.cl.unizh.ch/CLpublications.html

12. Hasida Kôiti. The GDA Tag Set. http://www.i-content.org/GDA/tagset.html.
13. Nicholas Kushmerick, Daniel S. Weld, and Robert B. Doorenbos. Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence (IJCAI97)*, pages 729–737, 1997.
14. Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. Technical report, W3C, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.
15. Diego Mollá, Rolf Schwitter, Michael Hess, and Rachel Fournier. ExtrAns, an answer extraction system. *T.A.L. special issue on Information Retrieval oriented Natural Language Processing*, pages 495–522, 2000. [1]
16. Diego Mollá, Rolf Schwitter, Fabio Rinaldi, James Dowdall, and Michael Hess. Anaphora resolution in Extrans. In *The 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization*, Venice, June 2003. [1]
17. James Pustejovsky, Roser Sauri, Andrea Setzer, Robert Gaizauskas, and Bob Ingria. TimeML Annotation Guideline 1.00 (internal version 0.4.0), July 2002. http://www.cs.brandeis.edu/ jamesp/arda/time/documentation/TimeML-Draft3.0.9.html.
18. Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1998.
19. Fabio Rinaldi, James Dowdall, Michael Hess, Kaarel Kaljurand, and Magnus Karlsson. The Role of Technical Terminology in Question Answering. In *Proceedings of TIA-2003, Terminologie et Intelligence Artificielle*, pages 156–165, Strasbourg, April 2003. [1]
20. Fabio Rinaldi, James Dowdall, Michael Hess, Kaarel Kaljurand, Mare Koit, Kadri Vider, and Neeme Kahusk. Terminology as Knowledge in Answer Extraction. In *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering (TKE02)*, pages 107–113, Nancy, 28–30 August 2002. [1]
21. Fabio Rinaldi, James Dowdall, Michael Hess, Kaarel Kaljurand, Andreas Persidis, Babis Theodoulidis, Bill Black, John McNaught, Haralampos Karanikas, Argyris Vasilakopoulos, Kelly Zervanou, Luc Bernard, Gian Piero Zarri, Hilbert Bruins Slot, Chris van der Touw, Margaret Daniel-King, Nancy Underwood, Agnes Lisowska, Lonneke van der Plas, Veronique Sauron, Myra Spiliopoulou, Marko Brunzel, Jeremy Ellman, Giorgos Orphanos, Thomas Mavroudakis, and Spiros Taraviras. Parmenides: an opportunity for ISO TC37 SC4? In *The ACL-2003 workshop on Linguistic Annotation, July 2003, Sapporo, Japan.*, 2003. [1]
22. Fabio Rinaldi, James Dowdall, Michael Hess, Diego Mollá, and Rolf Schwitter. Towards Answer Extraction: an application to Technical Domains. In *ECAI2002, European Conference on Artificial Intelligence, Lyon*, pages 460–464, 21–26 July 2002. [1]
23. Fabio Rinaldi, James Dowdall, Michael Hess, Diego Mollá, Rolf Schwitter, and Kaarel Kaljurand. Knowledge-Based Question Answering. In *Proceedings of KES-2003, Knowledge-Based Intelligent Information and Engineering Systems*, Oxford, September 2003. Accepted for publication.[1]
24. Daniel D. Sleator and Davy Temperley. Parsing English with a link grammar. In *Proc. Third International Workshop on Parsing Technologies*, pages 277–292, 1993.
25. TEI Consortium. The text encoding initiative, 2003. http://www.tei-c.org/.
26. Ellen M. Voorhees. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378, 2001.