Integrating Language Understanding Agents Into the Semantic Web*

Akshay Java, Tim Finin and Sergei Nirenburg
University of Maryland, Baltimore County, Baltimore MD 21250
{aks1,finin,sergei}@umbc.edu

Abstract

Many intelligent agents need knowledge and information to support their reasoning and problem solving. The World Wide Web is a vast, open, accessible and free source of knowledge, but virtually all of it is encoded as natural language text – a form difficult for most agents to directly understand. We describe initial work on adapting a mature language understanding agent to process Web text and publish its output in the Semantic Web language OWL. This approach adds knowledge on the Web in a form designed for agents to use. Moreover, language understanding agents can use the growing knowledge on the Semantic Web in their own language understanding tasks. Importing and exporting knowledge in the different knowledge representation formalisms used by these agents poses significant challenges. In particular we need to bridge the gap between the representation features of traditional non web-based representations and newer web-based formalisms such as OWL.

Introduction

A significant number of documents already exist on the Semantic Web in representations such as RDF and OWL (Ding *et al.* 2004). However, the vast majority of content on the Web remains as natural language text. It could be envisioned that specialized agents would exist that understand Natural Language text and share the information with other agents. Such agents would contribute to the Semantic Web, enriching the global *knowledge base*, and enable other Natural Language Processing (NLP) tools with ready knowledge to help them in their language understanding tasks. For example, an NLP tool can potentially use information present in a FOAF description to help in the disambiguation and reference resolution task. Additionally, there are a number of domain ontologies present on the Semantic Web. Information in these could be of great value for extending the ontology used by NLP tools.

Knowledge sharing is a critical factor to enable agents on the Semantic Web to use information extracted from NL text or to be able to provide information that can be used by NLP tools. This would require importing and exporting ontologies and facts from one representation formalism to another. One of the challenges is to bridge the gap between traditional, non web-based representations (frame based systems, etc) and newer web based representations such as OWL. Many NLP systems such as OntoSem (Nirenburg & Raskin 2005) use frame based representations to construct a model or ontology of the world. Such an ontology is then used to extract and represent meaning from Natural Language text.

In this paper we describe our initial efforts towards integrating OntoSem, and its large ontology and fact repositories into Semantic Web representations. We discuss some of the key challenges and propose some solutions towards overcoming them. In particular,

- We describe a way to transform an expressive KR system into web-based KR representations. Since Ontosem belongs to a general class of frame-based NLP system, we believe that the challenges and solutions described here are applicable to general KR systems as well.
- It is quite likely that for any KR system, transforming from one representation to another would always be lossfull. However for most applications it would suffice even if partial transformations are provided.
- By integrating such language understanding agents into the Semantic Web we make available a lot more information to agents, but it also has an implication that agents on the Semantic Web should be able to reason in presence of incomplete or sometimes even inaccurate annotations.

Finally, we briefly describe SemNews, a prototype application that we have developed as a testbed for this work. SemNews monitors RSS feeds of news articles, invokes OntoSem to understand their summaries, and publishes OntoSem's meaning representations in OWL on the Web.

Related Work

Recently, there has been a lot of interest in applying Information extraction technologies for the Semantic Web. However, few systems capable of deeper semantic analysis have been applied in Semantic Web related tasks. Information extraction tools work best when the types of objects that need to be identified are clearly defined, for example the objective in MUC (Grishman & Sundheim 1996) was to find the various named entities in text. Using OntoSem, we aim to not

*Partial support for this research was provided by Lockheed Martin Corporation.

only to provide such information, but also convert the text meaning representation of natural language sentences into Semantic Web representations.

A project closely related to our work was an effort to map the Mikrokosmos knowledge base to OWL (Beltran-Ferruz, ez Caler, & P.Gervas 2004; Beltran-Ferruz, Gonzalez-Caler, & P.Gervas 2004). Mikrokosmos is a precursor to OntoSem and was developed with the original idea of using it as an interlingua in machine translation related work. This project developed some basic mapping functions that can create the class hierarchy and specify the properties and their respective domains and ranges. In our system we describe how facets, numeric attribute ranges can be handled and more importantly we describe a technique for translating the sentences from their Text Meaning Representation to the corresponding OWL representation thereby providing semantically marked up Natural Language text for use by other agents.

Oliver et al. (Dameron, Rubin, & Musen 2005) describe an approach to representing the Foundational Model of Anatomy (FMA) in OWL. FMA is a large ontology of the human anatomy and is represented in a frame-based knowledge representation language. Some of the challenges faced were the lack of equivalent OWL representations for some frame based constructs and scalability and computational issues with the current reasoners.

Schlangen et al. (Schlangen, Stede, & Bontas 2004) describe a system that that combines a natural language processing system with Semantic Web technologies to support the content-based storage and retrieval of medical pathology reports. The NLP component was augmented with a background knowledge component consisting of a a domain ontology represented in OWL. The result supported the extraction of domain specific information from natural language reports which was them mapped back into a Semantic Web representation.

TAP (R.V.Guha & McCool 2003) is an open source project lead by Stanford University and IBM Research aimed at populating the Semantic Web with information by providing tools that make the web a giant distributed Database. TAP provides a set of protocols and conventions that create a coherent whole of independently produced bits of information, and a simple API to navigate the graph. Local, independently managed knowledge bases can be aggregated to form selected centers of knowledge useful for particular applications.

Kruger et al. (Krueger *et al.* 2004) developed an application that learned to extract information from talk announcements from training data using an algorithm based on Stalker (Muslea, Minton, & Knoblock 2001). The extracted information was then encoded as markup in the Semantic Web language DAML+OIL, a precursor to OWL. The results were used as part of the ITTALKS system (Cost *et al.* 2002).

The Haystack Project has developed system (Hogue & Karger 2005) enabling users to train a browsers to extract Semantic Web content from HTML documents on the Web. Users provide examples of semantic content by highlighting them in their browser and then describing their meaning.

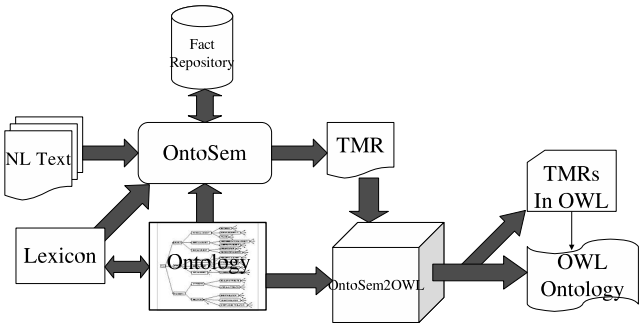


Figure 1: OntoSem2OWL is designed to translate OntoSem's ontology and *text meaning representations* (TMRs) from their native frame-based form into the Semantic Web language OWL. It can also translate TMRs in OWL back into their native representation.

Generalized wrappers are then constructed to extract information and encode the results in RDF. The goal is to let individual users generate Semantic Web content from text on web pages of interest to them.

The Cyc project has developed a very large knowledge base of common sense facts and reasoning capabilities. Recent efforts (Witbrock *et al.* 2004) include the development of tools for automatically annotating documents and exporting the knowledge in OWL. The authors also highlight the difficulties in exporting an expressive representation like CycL into OWL due to lack of equivalent constructs.

The OntoSem Ontology

Ontological Semantics (OntoSem) is a theory of meaning in natural language text (Nirenburg & Raskin 2001). The OntoSem environment is a rich and extensive tool for extracting and representing meaning in a language independent way. The OntoSem system is used for a number of applications such as machine translation, question answering, information extraction and language generation. It is supported by a *constructed world model* (Nirenburg & Raskin 2005) encoded as a rich ontology. The Ontology is represented as a directed acyclic graph using IS-A relations. It contains about 8000 concepts that have on an average 16 properties per concept. At the topmost level the concepts are: OBJECT, EVENT and PROPERTY.

The OntoSem ontology is expressed in a frame-based representation and each of the frames corresponds to a concept. The concepts are defined using a collection of slots that could be linked using IS-A relations. A slot consists of a PROPERTY, FACET and a FILLER.

```
ONTOLOGY ::= CONCEPT+
CONCEPT  ::= ROOT | OBJECT-OR-EVENT | PROPERTY
SLOT     ::= PROPERTY + FACET + FILLER
```

A property can be either an attribute, relation or ontology slot. An ontology slot is a special type of property that is used to describe and organize the ontology. The ontology is closely tied to the lexicon to make it language independent. There is a lexicon for each language and stored "meaning procedures" that are used to disambiguate word senses

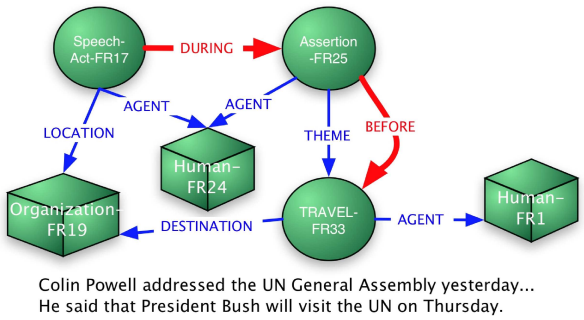


Figure 2: A schematic rendering of the TMR facts generated by OntoSem from the text *"Colin Powell addressed the UN General Assembly yesterday ... He said that President Bush will visit the UN on Thursday."*

ORGANIZATION-17

```
textpointer          UN-GENERAL-ASSEMBLY
word-num             3
HAS-NAME             "United Nations"
LOCATION-OF          SPEECH-ACT-16
```

SPEECH-ACT-16

```
textpointer      ADDRESS
word-num         1
LOCATION         ORGANIZATION-17
AGENT            HUMAN-15
TIME             (ABSOLUTE-TIME
                   (YEAR 2003)
                   (MONTH 9)
                   (DATE 23))
```

HUMAN-15

```
textpointer      *PERSON*
word-num         0
AGENT-OF         SPEECH-ACT-16
HAS-NAME         ((FIRST COLIN)
                  ((LAST POWELL))
FR-REFERENCE     HUMAN-FR24
```

HUMAN-FR24

```
HAS-ALIAS     (Powell
               "Colin L. Powell"
               "Colin Luther Powell"
               "Colin Powell")
```

Figure 3: OntoSem constructs this text meaning representation (TMR) for the sentence *"Colin Powell addressed the UN General Assembly yesterday"*.

and references. Thus keeping the concepts defined relatively few and making the ontology small. The English lexicon for example contains about 20,000 word senses. The ontology is also supported by an *Ontomasticon* (Nirenburg & Raskin 2005), which is a lexicon of proper names.

The OntoSem environment takes as input unrestricted text and performs different syntactic and semantic processing steps to convert it into a set of Text Meaning Representations (TMR). The TMR is a representation of the meaning of the text and is expressed using the various concepts defined in the ontology. The learned instances from the text are stored in a *fact repository* which essentially forms the knowledge base of OntoSem. As an example the sentence: *"Colin Powell addressed the UN General Assembly yesterday"* is converted to the TMR shown in Figure 3. A more detailed description of OntoSem and its features is available in (Nirenburg & Raskin 2005) and (ili).

Mapping OntoSem to OWL

We have developed **OntoSem2OWL** as a tool to convert OntoSem's ontology and TMRs encoded in it to OWL. This enables an agent to use OntoSem's environment to extract semantic information from natural language text. Ontology Mapping deals with defining functions that describe how concepts in one ontology are related to the concepts in some other ontology (Dejing Dou & Qi 2002). Ontology translation process converts the sentences that use the source ontology into their corresponding representations in the target ontology. In converting the OntoSem Ontology to OWL, we are performing the following tasks:

- Translating the OntoSem ontology deals with mapping the semantics of OntoSem into a corresponding OWL version.
- Once the ontology is translated the sentences that use the ontology are syntactically converted.
- In addition OntoSem is also supported by a fact repository which is also mapped to OWL.

OntoSem2OWL is a rule based translation engine that takes the OntoSem Ontology in its LISP representation and converts it into its corresponding OWL format. The following is an example of how a concept ONTOLOGY-SLOT is described in OntoSem:

```
(make-frame definition
  (is-a (value (common ontology-slot)))
  (definition (value (common "Human
       readable explanation for a concept")))
  (domain (sem (common all))))
```

Its corresponding OWL representation is:

```
<owl:ObjectProperty rdf:ID="definition">
 <rdfs:subPropertyOf>
   <owl:ObjectProperty rdf:about="#ontology-slot"/>
 </rdfs:subPropertyOf>
 <rdfs:label>
   "Human readable explanation for a concept"
 </rdfs:label>
 <rdfs:domain>
  <owl:Class rdf:about="#all"/>
 </rdfs:domain>
</owl:ObjectProperty>
```

We will briefly describe how each of the OntoSem features are mapped into their OWL versions: classes, properties, facets, attribute ranges and TMRs.

Handling Classes

New concepts are defined in OntoSem using *make-frame* and related to other concepts using the *is-a* relation. Each concept may also have a corresponding definition. Whenever the system encounters a *make-frame* it recognizes that this is a new concept being defined. OBJECT or EVENT are mapped to *owl:Class* while, PROPERTIES are mapped to *owl:ObjectProperty*. ONTOLOGY-SLOTS are special properties that are used to structure the ontology. These are also mapped to *owl:ObjectProperty*. Object definitions are created using *owl:Class* and the IS-A relation is mapped using *owl:subClassOf*. Definition property in OntoSem has the same function as *rdfs:label* and is mapped directly. The table 1 shows the usage of each of these features in OntoSem.

	case	times used	mapped using
1	total Class/Property make-frame	8199	owl:class or owl:ObjectProperty
2	Definition	8192	rdfs:label
3	is-a relationship	8189	owl:subClassOf

Table 1: Table showing how often each of the Class related constructs are used

Handling Properties

Whenever the level 1 parent of a concept is of the type PROPERTY it is translated to *owl:ObjectProperty*. Properties can also be linked to other properties using the IS-A relation. In case of properties, the IS-A relation maps to the *owl:subPropertyOf*. Most of the properties also contain the domain and the range slots. Domain defines the concepts to which the property can be applied and the ranges are the concepts that the property slot of an instance can have as fillers. OntoSem domains are converted to *rdfs:domain* and ranges are converted to *rdfs:range*. For some of the properties OntoSem also defines inverses using the INVERSE-OF relationship. It can be directly mapped to the *owl:inverseOf* relation.

In case there are multiple concepts defined for a particular domain or range, OntoSem2OWL handles it using *owl:unionOf* feature. For example:

```
(make-frame controls
 (domain
  (sem (common physical-event
               physical-object
               social-event
               social-role)))
  (range (sem (common actualize
                      artifact
                      natural-object
                      social-role)))
  (is-a (value (common relation)))
  (inverse (value (common controlled-by)))
  (definition
    (value (common
     "A relation which relates concepts to
      what they can control"))))
```

is mapped to

```
<owl:ObjectProperty rdf:ID= "controls">
 <rdfs:domain>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#physical-event"/>
      <owl:Class rdf:about="#physical-object"/>
      <owl:Class rdf:about="#social-event"/>
      <owl:Class rdf:about="#social-role"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:domain>
 <rdfs:range>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#actualize"/>
<owl:Class rdf:about="#artifact"/>
<owl:Class rdf:about="#natural-object"/>
<owl:Class rdf:about="#social-role"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:range>
 <rdfs:subPropertyOf>
   <owl:ObjectProperty rdf:about="#relation"/>
 </rdfs:subPropertyOf>
 <owl:inverseOf rdf:resource="#controlled-by"/>
 <rdfs:label>
  "A relation which relates concepts to
   what they can control"
 </rdfs:label>
</owl:ObjectProperty>
```

The table 2 describes the typical usages of the property related constructs in OntoSem.

Handling Facets

OntoSem uses facets as a way of restricting the fillers that can be used for a particular slot. In OntoSem there are six facets that are created and one,*inv* that is automatically generated. The table 3 shows the different facets and how often they are used in OntoSem.

- *SEM and VALUE*: These are the most commonly used facets. OntoSem2OWl handles these identically and are maps them using *owl:Restriction* on a particular property. Using *owl:Restriction* we can locally restrict the type of values a property can take unlike *rdfs:domain* or *rdfs:range* which specifies how the property is globally restricted (McGuinness & van Harmelen 2004).
- *RELAXABLE-TO*: This facet indicates that the value for the filler can take a certain type. It is a way of specifying "typical violations". One way of handling RELAXABLE-TO is to add this information in an annotation and also add this to the classes present in the *owl:Restriction*.
- *DEFAULT*: OWL provides no clear way of representing defaults, since it only supports monotonic reasoning and this is one of the issues that have been expressed for future extensions of OWL language (Horrocks, Patel-Schneider, & van Harmelen 2003). These issues need to be further investigated in order to come up with an appropriate equivalent representation in OWL. One approach is to use rule languages like SWRL (Horrocks *et al.* 2004) to express such defaults and exceptions. Another approach would be to elevate facets to properties. This can be done by combining the property-facet to make a new property. Thus a concept of an apple that has a property color with the default facet value 'red' could be translated to a new property in the owl version of the frame where the property name is color-default and it can have a value of red.
- *DEFAULT-MEASURE*: This facet indicates what the typical units of measurements are for a particular property. This can be handled by creating a new property named MEASURING-UNITS or adding this information as a rule.
- *NOT*: This facet specifies that certain values are not permitted in the filler of the slot in which this is defined. *NOT* facet can be handled using the *owl:disjointWith* feature.
- *INV*: This facet need not be handled since this information is already covered using the inverse property which is mapped to *owl:inverseOf*.

Although DEFAULT and DEFAULT-MEASURE provides useful information, it can be noticed from 3 that relatively they are used less frequently. Hence in our use cases, ignoring these facets does not lose a lot of information.

	case	frequency	mapped using
1	domain	617	rdfs:domain
2	domain with not facet	16	owl:disjointWith
3	range	406	rdfs:range
4	range with not facet	5	owl:disjointWith
5	inverse	260	owl:inverseOf

Table 2: Table showing how often each of the Property related constructs are used

	case	frequency	mapped using
1	value	18217	owl:Restriction
2	sem	5686	owl:Restriction
3	relaxable-to	95	annotation
4	default	350	**not handled**
5	default-measure	612	**not handled**
6	not	134	owl:disjointWith
7	inv	1941	not required

Table 3: Table showing how often each of the facets are used

Handling Attribute Ranges

Certain fillers can also take numerical ranges as values. For instance the property *age* can take a numerical value between 0 and 120 for instance. Additionally $<, >, <>$ could also be used in TMRs. Attribute ranges can be handled using XML Schema (xml 2004) in OWL. The following is an example of how the property *age* could be represented in OWL using *xsd:restriction*:

```
<xsd:restriction base="integer">
   <xsd:minInclusive value="0">
   <xsd:maxExclusive value="120">
</xsd:restriction>
```

Converting Text Meaning Representations

Once the OntoSem ontology is converted into its corresponding OWL representation, we can now translate the text meaning representations into statements in OWL. In order to do this we can use the namespace defined as the OntoSem ontology and use the corresponding concepts to create the representation. The TMRs also contain additional information such as ROOT-WORDS and MODALITY. These are used to provide additional details about the TMRs and are added to the annotations. In addition TMRs also contain certain triggers for 'meaning procedures' such as TRIGGER-REFERENCE and SEEK-SPECIFICATION. These are actually procedural attachments and hence can not be directly mapped into the corresponding OWL versions.

Sentence: *Ohio Congressman Arrives in Jordan*

TMR

```
(COME-1740
 (TIME (VALUE (COMMON (FIND-ANCHOR-TIME))))
 (DESTINATION (VALUE (COMMON CITY-1740)))
 (AGENT (VALUE (COMMON POLITICIAN-1740)))
 (ROOT-WORDS (VALUE (COMMON (ARRIVE))))
 (WORD-NUM (VALUE (COMMON 2)))
 (INSTANCE-OF (VALUE (COMMON COME))))
```

TMR in OWL

```
<ontosem:come rdf:about="COME-1740">
 <ontosem:destination
          rdf:resource="#CITY-1740"/>
 <ontosem:agent
    rdf:resource="#POLITICIAN-1740"/>
</ontosem:come>
```

TMR

```
(POLITICIAN-1740
  (AGENT-OF (VALUE (COMMON COME-1740)))
  ;; Politician with some relation to Ohio. A
  ;; later meaning procedure should try to find
  ;; that the relation is that he lives there.
  (RELATION (VALUE (COMMON PROVINCE-1740)))
  (MEMBER-OF (VALUE (COMMON CONGRESS)))
  (ROOT-WORDS (VALUE (COMMON (CONGRESSMAN))))
  (WORD-NUM (VALUE (COMMON 1)))
  (INSTANCE-OF (VALUE (COMMON POLITICIAN))))
```

TMR in OWL

```
<ontosem:politician rdf:about="POLITICIAN-1740">
  <ontosem:agent-of rdf:resource="#COME-140"/>
  <ontosem:relation rdf:resource="#PROVINCE-1740"/>
  <ontosem:member-of rdf:resource="#congress"/>
</ontosem:politician>
```

TMR

```
(CITY-1740
  (HAS-NAME (VALUE (COMMON "JORDAN")))
  (ROOT-WORDS (VALUE (COMMON (JORDAN))))
  (WORD-NUM (VALUE (COMMON 4)))
  (DESTINATION-OF (VALUE (COMMON COME-1740)))
  (INSTANCE-OF (VALUE (COMMON CITY))))
```

TMR in OWL

```
<ontosem:city rdf:about="CITY-1740">
  <ontosem:has-name>JORDAN</ontosem:has-name>
  <ontosem:destination-of rdf:resource="#COME-1740"/>
</ontosem:city>
```

Preliminary Evaluation

There are several dimensions along which this research could be evaluated. Our translation model involves translating ontologies and instances (facts) in both directions: from OntoSem to an OWL version of the OntoSem Ontology and from the OWL version of OntoSem into OntoSem. For the translation to be truly useful, it should also involves the translation between the OWL version of OntoSem's ontologies and facts and the ontologies in common use on the Semantic Web (e.g., FOAF (foa), Dublin Core (Miller & Brickley 2002), OWL-S (owl 2004), OWL-time (Hobbs & Pan 2004), etc.).

Since our current work has concentrated on the initial step of translating from OntoSem to OWL, we will enumerate some of the issues from that perspective. Translating in the

opposite direction raises similar, though not identical, issues. The chief translation measures we have considered are as follows:

- **Syntactic correctness.** Does the translation produce syntactically correct RDF and OWL? The resulting documents can be checked with appropriate RDF and OWL validation systems.
- **Semantic validity.** Does the translation produce RDF and OWL that is semantically well formed? An RDF or OWL file can be syntactically valid yet contain errors that violate semantic constrains in the language. For example, an OWL class should not be disjoint with itself if it has any instances. Several OWL validation services make some semantic checks in addition to syntactic ones. A full semantic validity check is quite difficult and, to our knowledge, no system attempts one, even for decidable subsets of OWL.
- **Meaning preservation.** Is the meaning of the generated OWL representation identical to that of the OntoSem representation? This is a very difficult question to answer, or even to formulate, given the vast differences between the two knowledge representation systems. However, we can easily identify some constructs, such as defaults, that clearly can not be captured in OWL, leading to a loss of information and meaning when going from OntoSem to OWL.
- **Feature minimization.** OWL is a complex representation language, some of whose features make reasoning difficult. A number of levels of complexity can be identified (e.g., the OWL *species: Lite, DL and Full*). In general, we would like the translation service to not use a complex feature unless it is absolutely required. Doing so will reduce the complexity of reasoning with the generated ontology.
- **Translation complexity.** What are the speed and memory requirements of the translation. Since, in general, a translation might require reasoning, this could be an issue.

Since our project is still in an early stage, we report on some preliminary evaluation metrics covering the basic OntoSem to OWL translation.

OntoSem2OWL uses the Jena Semantic Web Framework (McBride 2001) internally to build the OWL version of the Ontology. The ontologies generated were successfully validated using two automated RDF validators: the W3C's RDF Validation Service (w3c) and the WonderWeb OWL Ontology Validator (won).

There were a total of about 8000 concepts in the original OntoSem ontology. The total number of triples generated in the translated version was just over 100,000. These triples included a number of blank nodes – RDF nodes representing objects without identifiers that are required due to RDF's low-level triple representation.

Because the generated ontologies required the use of the OWL's *union* and *inverseOf* features, the results fall in the *OWL full* class in terms of the the level of expressivity.

Using the Jena API it takes about 10-40 seconds to build the model, depending upon the reasoner employed. The computation of transitive closure and basic RDF Schema inferencing takes approximately ten seconds on a typical workstation. The OWL Micro reasoner takes about 40 seconds while OWL Full reasoner fails, possibly due to the large search space. The OntoSem ontology in its OWL representation can be successfully loaded into the SWOOP (Kalyanpur, Parsia, & Hendler 2005) OWL editor for browsing, editing and further validation.

Based on our preliminary results, we found that OntoSem2OWL is able to translate most of the OntoSem ontology into a form that is syntactically valid and, in so far as current validators can tell, free of semantic problems. There are some problems in representing defaults and correctly mapping some of the facets, however these are used relatively less frequently.

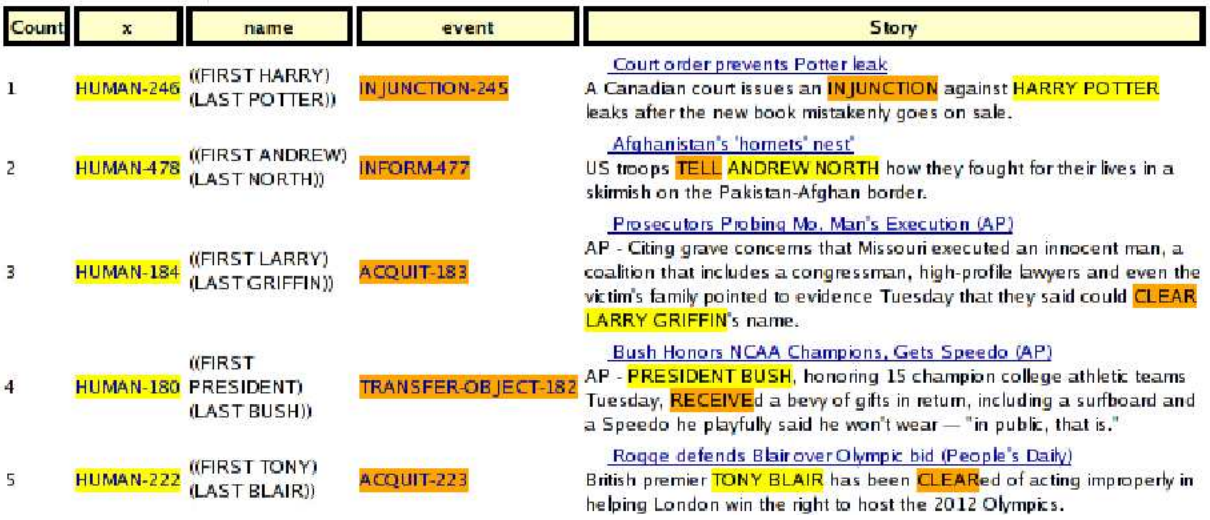
An Application Testbed

One of the motivations for integrating language understanding agents into the Semantic Web is to enable applications to use the information that is published in free text along with other semantic web data. SemNews (Sem ; Java, Finin, & Nirenburg 2006) is a semantic news framework that monitors different RSS News Sources and provides a structured representation of the meaning of the news. The RSS descriptions of the news articles are processed by OntoSem resulting in a TMR which is then converted into OWL. The OWL TMR for each document is stored in a Redland-based triple store, allowing other applications and users perform semantic queries over the documents. This enables them to search for information that would otherwise not be easy to find using simple keyword based search. The TMRs are published as RDF documents which are available to agents and added a special document collection which is indexed by the Swoogle Semantic Search engine (Ding *et al.* 2004).

Developing SemNews provided a perspective on some of the general problems of integrating a mature language processing system like OntoSem into a Semantic Web oriented application. While doing a complete and faithful translation of knowledge from OntoSem's native meaning representation language into OWL is not feasible, we found the problems to be manageable in practice for several reasons.

First, OntoSem's knowledge representation features that were most problematic for translation are not used with great frequency. For example, the default values, relaxable range constraints and procedural attachments were used relatively rarely in OntoSem's ontology. Thus shortcomings in the OWL version of OntoSem's ontology are limited and can be circumscribed. We are also optimistic that most Semantic Web content will be amenable to translation into OntoSem's representation. It's likely that the majority of Semantic Web content will be encoded with relatively simple ontologies that use only RDF and RDFS and do not use OWL. Many of the OWL ontologies may be partionable into portions which do not use difficult to translation features and those that do.

Second, the goal is not just to support translation between OntoSem and a complete an faithful OWL version of OntoSem. It is unlikely that most Semantic Web content producers or consumers will use OntoSem's ontology. Rather,



Count	x	name	event	Story
1	HUMAN-246	((FIRST HARRY) (LAST POTTER))	INJUNCTION-245	Court order prevents Potter leak A Canadian court issues an INJUNCTION against HARRY POTTER leaks after the new book mistakenly goes on sale.
2	HUMAN-478	((FIRST ANDREW) (LAST NORTH))	INFORM-477	Afghanistan's 'hornets' nest' US troops TELL ANDREW NORTH how they fought for their lives in a skirmish on the Pakistan-Afghan border.
3	HUMAN-184	((FIRST LARRY) (LAST GRIFFIN))	ACQUIT-183	Prosecutors Probing Mo. Man's Execution (AP) AP - Citing grave concerns that Missouri executed an innocent man, a coalition that includes a congressman, high-profile lawyers and even the victim's family pointed to evidence Tuesday that they said could CLEAR LARRY GRIFFIN's name.
4	HUMAN-180	((FIRST PRESIDENT) (LAST BUSH))	TRANSFER-OBJECT-182	Bush Honors NCAA Champions, Gets Speedo (AP) AP - PRESIDENT BUSH, honoring 15 champion college athletic teams Tuesday, RECEIVEd a bevy of gifts in return, including a surfboard and a Speedo he playfully said he won't wear — "in public, that is."
5	HUMAN-222	((FIRST TONY) (LAST BLAIR))	ACQUIT-223	Rogge defends Blair over Olympic bid (People's Daily) British premier TONY BLAIR has been CLEARed of acting improperly in helping London win the right to host the 2012 Olympics.

Figure 4: SemNews. Shows results for query "Find all humans and what are they the theme-of"

we expect common consensus ontologies like FOAF, Dublin Core, and SOUPA to emerge and be widely used on the Semantic Web. The real goal is thus to mediate between OntoSem and a host of such consensus ontologies. We believe that these translations between OWL ontologies will of necessity be inexact and thus introduce some meaning loss or drift. So, the translation between OntoSem's native representation and the OWL form will not be the only lossy one in the chain.

Third, the SemNews application generates and exports facts, rather than concepts. The prospective applications coupling a language understanding agent and the Semantic Web that we have examined share this focus on importing and exporting instance level information. To some degree, this obviates many translation issues, since these mostly occur at the concept level. While we may not be able to exactly express OntoSem's complete concept of a book's author in the OWL version, we can translate the simple instance level assertion that a known individual is the author of a particular book and further translate this into the appropriate triple using the FOAF and Dublin Core RDF ontologies.

Finally, with a focus on importing and exporting instances and assertions of fact, we can require these to be generated using the native representation and reasoning system. Rather than exporting OntoSem's concept definitions and a handful of facts to OWL and then using an OWL reasoner to derive the additional facts which follow, we can require OntoSem to precompute all of the relevant facts. Similarly, when importing information from an OWL representation, the complete model can be generated and just the instances and assertions translated and imported.

Conclusion

Natural Language processing agents can provide a great service by analyzing text published on the Web and publishing annotations which capture aspects of the text's meaning. Their output will enable many more agents to benefit from the knowledge and facts expressed in the text. Similarly, language processing agents need a wide variety of knowledge and facts to correctly understand the text they process. Much of the needed knowledge may be found on the Web already encoded in RDF and OWL and thus easy to import.

One of the key problems to be solved in order to integrate language understanding agents into the Semantic Web is the problem of translating knowledge and information from their native representation systems and the Semantic Web languages. We have described initial work aimed at preparing the the OntoSem language understanding system to be integrated into applications on the Web. OntoSem is a large scale, sophisticated natural language understanding system that uses a custom frame-based knowledge representation system with an extensive ontology and lexicon. These have been developed over many years and are adapted to the special needs of text analysis and understanding.

We have described a translation system that is being used to translate OntoSem's ontology into the Semantic Web language OWL. While the translator is not able to handle all of OntoSem's representational features, it is able to translate a large and useful subset. The translator has been used to develop SemNews as a prototype of a system that reads summaries of web news stories and publishes OntoSem's understanding of their meaning in OWL.

References

Beltran-Ferruz, P.; ez Caler, P.; and P.Gervas. 2004. Converting frames into OWL: Preparing Mikrokosmos for linguistic creativity. In *LREC Workshop on Language Resources for Linguistic Creativity*.

Beltran-Ferruz, P.; Gonzalez-Caler, P.; and P.Gervas. 2004. Converting Mikrokosmos frames into description logics. In *RDF/RDFS and OWL in Language Technology: 4th ACL Workshop on NLP and XML*.

Cost, R. S.; Finin, T.; Joshi, A.; Peng, Y.; Nicholas, C.; Soboroff, I.; Chen, H.; Kagal, L.; Perich, F.; Zou, Y.; and Tolia, S. 2002. ITtalks: A Case Study in the Semantic Web and DAML+OIL. *IEEE Intelligent Systems Special Issue*.

Dameron, O.; Rubin, D. L.; and Musen, M. A. 2005. Challenges in converting frame-based ontology into OWL: the Foundational Model of Anatomy case-study. In *American Medical Informatics Association Conference AMIA05*.

Dejing Dou, D. M., and Qi, P. 2002. Ontology translation by ontology merging and automated reasoning. In *Proc. EKAW Workshop on Ontologies for Multi-Agent Systems*.

Ding, L.; Finin, T.; Joshi, A.; Pan, R.; Cost, R. S.; Peng, Y.; Reddivari, P.; Doshi, V. C.; ; and Sachs, J. 2004. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*.

The friend of a friend(foaf) project. `http://www.foaf-project.org/`.

Grishman, R., and Sundheim, B. 1996. Message understanding conference-6: a brief history. In *Proceedings of the 16th Conference on Computational Linguistics*, 466–471.

Hobbs, J. R., and Pan, F. 2004. An ontology of time for the semantic web. *ACM Transactions on Asian Language Processing (TALIP)* 3(1):66–85. Special issue on Temporal Information Processing.

Hogue, A., and Karger, D. R. 2005. Thresher: Automating the unwrapping of semantic content from the world wide web. In *Proceedings of the Fourteenth International World Wide Web Conference*.

Horrocks, I.; Patel-Schneider, P. F.; Boley, H.; Tabet, S.; Grosof, B.; and Dean, M. 2004. Swrl: A semantic web rule language combining owl and ruleml. World Wide Web Consortium Specification.

Horrocks, I.; Patel-Schneider, P. F.; and van Harmelen, F. 2003. From shiq and rdf to owl: the making of a web ontology language. *J. Web Sem.* 1(1):7–26.

Institute for language and information technologies. `http://ilit.umbc.edu/`.

Java, A.; Finin, T.; and Nirenburg, S. 2006. Text understanding agents and the Semantic Web. In *Proceedings of the 39th Hawaii International Conference on System Sciences*.

Kalyanpur, A.; Parsia, B.; and Hendler, J. 2005. A tool for working with web ontologies. In *In Proceedings of the International Journal on Semantic Web and Information Systems*, volume 1.

Krueger, W.; Nilsson, J.; Oates, T.; and Finin, T. 2004. *Automatically Generated DAML Markup for Semistructured Documents*. Lecture Notes in Artificial Intelligence. Springer.

McBride, B. 2001. Jena: Implementing the RDF model and syntax specification. In *Proceedings of the WWW2001 Semantic Web Workshop*.

McGuinness, D. L., and van Harmelen, F. 2004. Owl web ontology language overview. `http://www.w3.org/TR/2004/REC-owl-features-20040210/#s3.4`.

Miller, D. B. E., and Brickley, D. 2002. Expressing simple dublin core in RDF / XML. Dublin Core Metadata Initiative Recommendation.

Muslea, I. A.; Minton, S.; and Knoblock, C. 2001. Hierarchical wrapper induction for semistructured information services. *Journal of Autonomous Agents and Multi-Agent Systems* 4(1/2):93–114.

Nirenburg, S., and Raskin, V. 2001. Ontological semantics, formal ontology, and ambiguity. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, 151–161. New York, NY, USA: ACM Press.

Nirenburg, S., and Raskin, V. 2005. *Ontological semantics*. MIT Press.

2004. OWL web ontology language for services (OWL-S). A W3C submission. `http://www.w3.org/Submission/2004/07/`.

R.V.Guha, and McCool, R. 2003. TAP: A semantic web toolkit. *Semantic Web Journal*.

Schlangen, D.; Stede, M.; and Bontas, E. P. 2004. Feeding owl: Extracting and representing the content of pathology reports. In *RDF/RDFS and OWL in Language Technology: 4th ACL Workshop on NLP and XML*.

Semnews application. `http://semnews.umbc.edu/`.

RDF validation service. `http://www.w3.org/RDF/Validator/`.

Witbrock, M.; Panton, K.; Reed, S.; Schneider, D.; Aldag, B.; Reimers, M.; and Bertolo, S. 2004. Automated OWL Annotation Assisted by a Large Knowledge Base. In *Workshop Notes of the 2004 Workshop on Knowledge Markup and Semantic Annotation at the 3rd International Semantic Web Conference ISWC2004*.

Wonderweb owl ontology validator. `http://phoebus.cs.man.ac.uk:9999/OWL/Validator`.

2004. Xml schema part 0: Primer. World Wide Web Consortium Specification. see `http://www.w3.org/TR/xmlschema-0/`.