Using a Natural Language Understanding System to Generate Semantic Web Content

Akshay Java, University of Maryland, USA

Sergei Nirneburg, University of Maryland, USA

Marjorie McShane, University of Maryland, USA

Timothy Finin, University of Maryland, USA

Jesse English, University of Maryland, USA

Anupam Joshi, University of Maryland, USA

ABSTRACT

We describe our research on automatically generating rich semantic annotations of text and making it available on the Semantic Web. In particular, we discuss the challenges involved in adapting the OntoSem natural language processing system for this purpose. OntoSem, an implementation of the theory of ontological semantics under continuous development for over 15 years, uses a specially constructed NLP-oriented ontology and an ontological-semantic lexicon to translate English text into a custom ontology-motivated knowledge representation language, the language of text meaning representations (TMRs). OntoSem concentrates on a variety of ambiguity resolution tasks as well as processing unexpected input and reference. To adapt OntoSem's representation to the Semantic Web, we developed a translation system, OntoSem2OWL, between the TMR language into the Semantic Web language OWL. We next used OntoSem and OntoSem2OWL to support SemNews, an experimental Web service that monitors RSS news sources, processes the summaries of the news stories, and publishes a structured representation of the meaning of the text in the news story.

Keywords: information extraction; natural language processing; OWL; RDF; Semantic Web

INTRODUCTION

A core goal of the development of the Semantic Web is to bring progressively more meaning to the information published on the Web. An accepted method of doing this is by annotating the text with a variety of kinds of metadata. Manual annotation is time-consuming and error-prone. Moreover, annotations must be made in a formal language whose use may require considerable training and expertise. Developing interactive tools for annotation is a problematic undertaking because it is not known whether they will be

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

in actual demand. A number of Semantic Web practitioners maintain that the desire to have their content available on the Semantic Web will compel people to spend the time and effort on manual annotation. However, even if such a desire materializes, people simply will not have enough time either to annotate each sentence in their texts or annotate a subset at a semantic level that is sufficiently deep to be used by advanced intelligent agents that are projected as users of the Semantic Web alongside people.

The alternative on the supply side is, then, automatic annotation. Within the current state of the art, automatically produced annotations are roughly at the level attainable by the latest information extraction techniques—a reasonably good level of capturing named entities with a somewhat less successful categorization of such entities (e.g., deciding whether *Jordan* is used as the first name of an individual or a reference to the Hashemite kingdom). Extracting more advanced types of semantic information, for example, types of events (to say nothing about determining semantic arguments, ''case roles" in AI terminology), is not quite within the current information extraction capabilities, though work in this direction is ongoing. Indeed, semantic annotation is at the moment an active subfield of computational linguistics, where annotated corpora are intended for use by machine-learning approaches to building natural language processing capabilities.

On the demand side of the Semantic Web, a core capability is improving the precision of the Web search, which will be facilitated by detailed semantic annotations that are unambiguous and sufficiently detailed to support the search engine in making fine-grained distinctions in calculating scores of documents. Another core capability is to transcend the level of document retrieval and, instead, return as answers to user queries specially generated pragmatically and stylistically appropriate responses. To attain this capability, intelligent agents must rely on very detailed semantic annotations of texts. We believe that such annotations will be, for all intents and purposes, complete text meaning representations, not just sets of semantic or pragmatic markers (and certainly not templates filled with uninterpreted snippets of the input text that are generated by the current information extraction methods).

To attain such goals, Semantic Web agents must be equipped with sophisticated semantic analysis systems that process text found on the Web and publish their analyses on the Web as annotations in a form accessible to other agents, using standard Semantic Web languages such as RDF and OWL. The Semantic Web will, thus, be useful for both human readers and robotic intelligent agents. The agents will benefit from the existence of deep semantic annotations in their application-oriented information processing tasks and also will be able to derive such annotations from text. People will not directly access the annotation (metadata) level but will benefit from higher-quality and better formulated responses to their queries.

This article describes initial work on responding to the needs and leveraging the offerings of the Web by merging knowledge-oriented natural language processing with web technologies to produce both an automatic annotation-generating capability and an enhanced Web service oriented at human users. The ontological-semantic natural language processing system OntoSem (Nirenburg & Raskin, 2001) provided the basis for the automatic annotation effort. In order to test and evaluate the utility of OntoSem on the Semantic Web, we have developed SemNews (Java, Finin & Nirenburg, 2006; Java, Finin & Nirenburg, 2005), a prototype application that monitors RSS feeds of news stories, applies OntoSem to understand the text, and exports the computed facts back to the Web in OWL. A prerequisite for this system integration is a utility for translating knowledge formats between OntoSem's knowledge representation language and ontologies and those of the Semantic Web.

Since our goal is to continuously improve the service, the quality of OntoSem results and system coverage must be continuously enhanced. The Web, in fact, contains a wealth of textual information that, once processed, can enhance OntoSem's knowledge base (its

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

ontology, lexicon and fact repository, see below for a more detailed description). This is why the knowledge format conversion utility, OntoSem2OWL, has been developed to translate both ways between OWL and OntoSem's knowledge representation language. Our initial experiments on automatic learning of ontological concepts and lexicon entries are reported in English and Nirenburg (2007).

The remainder of this article is organized as follows. We start with a brief review of some related work on annotation in computational linguistics, and on mapping knowledge between a text understanding system and the Semantic Web representation. Next, we introduce the knowledge resources of OntoSem and illustrate its knowledge representation language. In the section, Mapping OntoSem to OWL, we provide an overview of the architecture of our implemented system and describe the approach used and major issues discovered in using it to map knowledge between OntoSem's knowledge representation system and the Semantic Web language OWL. We go on in the next section tooutline some of the larger issues and challenges we expect to encounter. We describe an approach to evaluating the use of OntoSem, the translation of its ontology to OWL, and the effectiveness of the SemNews system in the section that follows.. In the section, titled Applications, we describe the SemNews application testbed and some general application scenarios we have explored to motivate and guide our research. Finally, we offer some comments on the problem of extracting knowledge from the Web in the section titled, Using the Web for Knowledge Acquisition,and we conclude this article with closing remarks.

RELATED WORK

The general problem of automatically generating and adding semantic annotations to text has been the focus of research for many years. Most of the work has not used the Semantic Web languages for encoding these annotations. We briefly describe some of the work here and point out some similarities and differences with our own.

Gildea and Jurafsky (2002) created a stochastic system that labels case roles of predicates with either abstract (e.g., AGENT, THEME) or domain-specific (e.g., MESSAGE, TOPIC) roles. The system trained on 50,000 words of hand-annotated text that was produced by the FrameNet project (Baker, Fillmore & Lowe, 1998). When tasked to segment constituents and identify their semantic roles (with fillers being un-disambiguated textual strings, not machine-tractable instances of ontological concepts, as in OntoSem), the system scored in the 60s in precision and recall. Limitations of the system include its reliance on hand-annotated data and its reliance on prior knowledge of the predicate frame type (i.e., it lacks the capacity to disambiguate productively). Semantics in this project is limited to case-roles.

The *Interlingual Annotation of Multilingual Text Corpora* project (Farwell et al., 2004) had as its goal the creation of a syntactic and semantic annotation representation methodology, and tested it out on seven languages (English, Spanish, French, Arabic, Japanese, Korean, and Hindi). The semantic representation, however, is restricted to those aspects of syntax and semantics that developers believe can be consistently handled well by hand annotators for many languages. The current stage of development includes only syntax and a limited semantics—essentially, thematic roles.

In the ACE project[1], annotators carry out manual semantic annotation of texts in English, Chinese and Arabic to create training and test data for research task evaluations. The downside of this effort is that the inventory of semantic entities, relations, and events is very small, and, therefore, the resulting semantic representations are coarse-grained, for example, there are only five event types. The project description promises more fine-grained descriptors and relations among events in the future. Another response to the clear insufficiency of syntax-only tagging is offered by the developers of PropBank, the Penn Treebank semantic extension. Kingsbury et al. (2002), who report:

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

It was agreed that the highest priority, and the most feasible type of semantic annotation, is coreference and predicate argument structure for verbs, participial modifiers and nominalizations, and this is what is included in PropBank.

Recently, there has been interest in exploiting information extraction techniques to text to produce annotations 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 only provide such information, but also to 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, Gonzalez-Caler & Gervas, 2004a; 2004b). Mikrokosmos (Beale, Nirenburg & Mahesh, 1995) is a precursor to OntoSem and was developed with the intent 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. Another translation effort involving Mikrokosmos produced the *Omega Ontology* (Philpot, Hovy & Pantel, 2005) by merging the content of Mikrokosmos with Wordnet with additional information sources.

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

TAP (R.V. Guha & McCool, 2003) is an open source project led 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.

Krueger, Nilsson, Oates and Finin (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 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. Generalized wrappers are then constructed to extract information and encode

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

the results in RDF. The goal is to let individual users generate Semantic Web content of interest to them from text on Web pages. More recently, the project has developed a Firefox plug-in Solvent that can be used to write screen scrapers to produce RDF data from Web pages.

The On-to-Knowledge project (Fensel, Harmelen & Akkermans, 2000) provides an ontology-based system for knowledge management. It uses Ontology-based Inference Layer (OIL) support for description logics (DL) and frame-based systems over the Web. OWL itself is an extension derived from OIL and DAML. The OntoExtract and OntoWrapper sub-system in On-to-knowledge were responsible for processing unstructured and structured text. These systems were used to automatically extract ontologies and express them in Semantic Web representations. At the heat of OntoExtract is an natural language processing system that processes text to perform lexical and semantic analysis. Finally, concepts found in free text are represented as an ontology.

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.

Finally, we mention the KIM platform (Kiryakov, Popov, Terziev, Manov & Ognyanoff, 2004) for automatic semantic annotation, indexing, and retrieval of documents. This system uses the GATE (Cunningham, 2002) language engineering system backed by structured ontologies in OWL to produce annotations.

ONTOSEM

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*, that is, a structured model of the classes of objects, properties, relations and constraints that might be described in text, encoded as a rich ontology. The ontology is represented as a directed acyclic graph using IS-A relations. It contains about 8,000 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
```

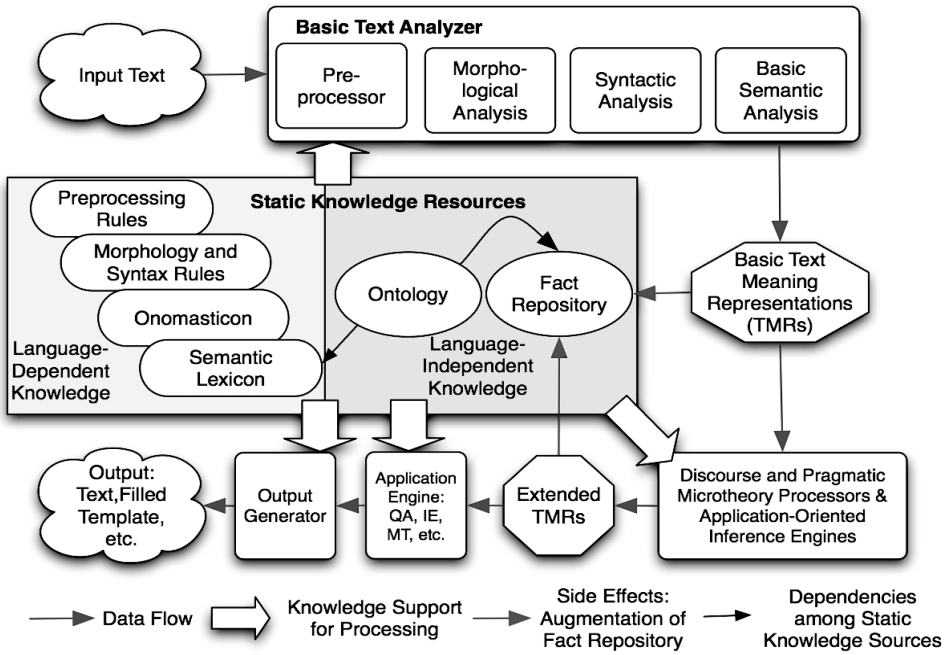
A property can be either an attribute, relation or ontology slot. An ontology slot is a special type of property that is used to describe and organize the ontology. The ontology is closely tied to the lexicon with language independence achieved through the use of multiple lexicons, one for each language with stored *meaning procedures*' that are used to disambiguate word senses and references. Thus, keeping the concepts defined relatively few and making the ontology small. Text analysis relies on extensive static knowledge resources, some of which are described in the following paragraphs.

The key knowledge base is the OntoSem language-independent ontology, which currently contains around 8,500 concepts, each of which is described by an average of 16 properties. The ontology is populated by concepts that we expect to be relevant cross-linguistically. The current experiment was run on a subset of the ontology containing about 6,000 concepts.

The OntoSem lexicon is a knowledge resource whose entries contain syntactic and semantic information (linked through variables),

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 1. 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.



as well as calls for procedural semantic routines when necessary. The semantic zone of an entry most frequently refers to ontological concepts, either directly or with property-based modifications, but also can describe words meaning extra-ontologically, for example, in terms of modality, aspect or time (see McShane, Nirenburg, Beale & O'Hara, 2005 for an in-depth discussion of the lexicon/ontology connection). The current English lexicon contains approximately 30,000 senses selected through corpus analysis over its many years of development in application to a number of domains. It includes most closed-class items[2] and frequent and polysemous[3] verbs. The base lexicon is expanded at runtime using an inventory of lexical (e.g., derivational-morphological) rules.

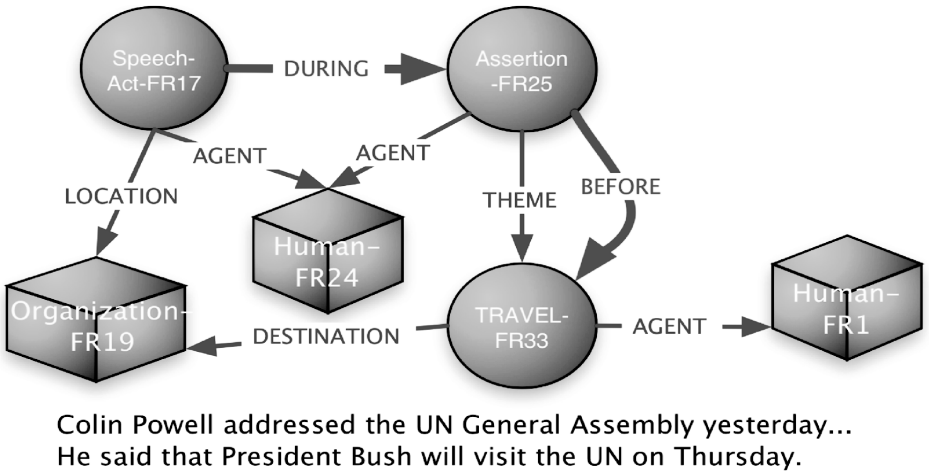
OnsoSem's onomasticon is its lexicon of proper names and contains approximately 350,000 entries. These include names of individual people (Colin Powell), governments (Belgium), organizations (United Nations), groups (Tamil Tigers) and places (Upper Manhattan).

OntoSem maintains a fact repository that contains *remembered instances* of ontological concepts, for example, SPEECH-ACT-3366 is the 3,366th instantiation of the concept SPEECH-ACT in the memory of a text-processing agent. Figure 2 shows the representation generated for the text "Collin Powell addressed the UN General Assembly yesterday. He said that President Bush will visit the UN on Thursday."' The fact repository is not used in the current experiment but will provide valuable semantically annotated context information for future experiments.

OntoSem's knowledge representation and reasoning system is the *Text Meaning Representation* (TMR) language. This is a frame-based

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 2. OntoSem constructs this text meaning representation (TMR) for the sentence "He (Colin Powell) asked the UN to authorize the war."



system with specific features to support the production of semantic descriptions from natural language text. Such descriptions are referred to as TMRs— text meaning representations.

The OntoSem syntactic-semantic analyzer is a knowledge resource that performs preprocessing (tokenization, named-entity and acronym recognition, etc.), morphological, syntactic and semantic analysis, and the creation of TMRs.

OntoSem knowledge resources have been constructed, enhanced, and maintained by trained acquirers using a broad variety of efficiency-enhancing tools, including graphical editors, enhanced search facilities, capabilities of automatically acquiring knowledge for classes of entities on the basis of manually acquired knowledge for a single representative of the class, for example, OntoSem's DEKADE environment (McShane et al., 2005) facilitates both knowledge acquisition and semi-automatic creation of *gold standard* TMRs, which also can be viewed as deep semantic text annotation.

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 (TMRs). The basic steps in processing the sentence to extract the meaning representation is shown in Figure 1. The preprocessor deals with identifying sentence and word boundaries, part of speech tagging, recognition of named entities, dates, and so forth. The syntactic analysis phase identifies the various clause level dependencies and grammatical constructs of the sentence. The TMR is a representation of the meaning of the text and is expressed using the various concepts defined in the ontology. The TMRs are produced as a result of semantic analysis, which uses knowledge sources such as lexicon, onomasticon, and fact repository to resolve ambiguities and time references. TMRs have been used as the substrate for question-answering (Beale, Lavoie, McShane, Nirenburg & Korelsky, 2004), machine translation (Beale, Nirenburg & Mahesh, 1995) and knowledge extraction. Once the TMRs are generated, OntoSem2OWL converts them to an equivalent OWL representation.

The learned instances from the text are stored in a *fact repository*, which essentially forms the instance-level knowledge base of OntoSem. As an example the sentence: "*He (Colin Powell) asked the UN to authorize the*

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

war" is converted to the TMR shown in Figure 3. A more detailed description of OntoSem and its features is available in Nirenburg and Raskin (2005) and from the Institute for language and information technologies (n.d.).

MAPPING ONTOSEM TO OWL

We have developed **OntoSem2OWL** (Java, Finin & Nirenburg, 2005) as a tool to convert OntoSem's ontology and TMRs encoded in it to the Web Ontology language 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 (Dou, McDermott, & Qi, 2005). 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 also is supported by a fact repository, which also is 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>
```

Figure 3. OntoSem constructs this text meaning representation (TMR) for the sentence "He (Colin Powell) asked the UN to authorize the war."

```
REQUEST-ACTION-69
  AGENT                HUMAN-72
  THEME                ACCEPT-70
  BENEFICIARY          ORGANIZATION-71
  SOURCE-ROOT-WORD     ask
  TIME                 (< (FIND-ANCHOR-TIME))

ACCEPT-70
  THEME                WAR-73
  THEME-OF             REQUEST-ACTION-69
  SOURCE-ROOT-WORD     authorize

ORGANIZATION-71
  HAS-NAME             United-Nations
  BENEFICIARY-OF       REQUEST-ACTION-69
  SOURCE-ROOT-WORD     UN

HUMAN-72
  HAS-NAME             Colin Powell
  AGENT-OF             REQUEST-ACTION-69
  SOURCE-ROOT-WORD     he      ; reference resolution has been carried out

WAR-73
  THEME-OF             ACCEPT-70
  SOURCE-ROOT-WORD     war
```

He asked the UN to authorize the war.

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

```
 <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 also may 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 also are 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. Table 1 shows the usage of each of these features in OntoSem.

Handling Properties

Whenever the level-one parent of a concept is of the type PROPERTY, it is translated to *owl:ObjectProperty*. Properties also can 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>
```

Table 1. This table shows how often each of the Class-related constructs were used in OntoSem's ontology

	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

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

```
</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>
```

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. 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 maps them using an *owl:Restriction* on a particular property. With an *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 (OWL Web ontology language for services, n.d.).
- **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, Patel-Schneider, Boley, Tabet, Grosof & Dean, 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

Table 2. shows some of the key properties used in OntoSem along with the frequency of use and the RDFS or OWL construct used to encode them

	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

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

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* facets can be handled using the *owl:disjointWith* feature.
- **INV:** This facet need not be handled since this information already is covered using the inverse property, which is mapped to *owl:inverseOf*.

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

Handling Attribute Ranges

Certain fillers also can take numerical ranges as values. For instance the property *age* can take a numerical value between 0 and 120 for instance. Additionally comparison functions such as <, >, and <> also could be used in TMRs. Attribute ranges can be handled using XML Schema (Fallside & Walmsley, 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 actually are procedural attachments and, hence, cannot be directly mapped into the corresponding OWL versions.

Sentence: *Ohio Congressman Arrives in Jordan*

TMR

```
(COME-1740
  (TIME (VALUE (COMMON (FIND-ANCHOR-
    TIME))))
```

Table 3 shows how often each of the facets was used in OntoSem's ontology.

	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

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

```
   (DESTINATION (VALUE (COMMON CITY-
      1740)))
   (AGENT (VALUE (COMMON POLITICIAN-
      1740)))
   (ROOT-WORDS (VALUE (COMMON (AR-
      RIVE))))
  (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 CON-
      GRESS)))
  (ROOT-WORDS (VALUE (COMMON (CON-
      GRESSMAN))))
 (WORD-NUM (VALUE (COMMON 1)))
  (INSTANCE-OF (VALUE (COMMON POLITI-
      CIAN))))
```

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 "JOR-
      DAN")))
    (ROOT-WORDS (VALUE (COMMON (JOR-
      DAN))))
   (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>
```

CHALLENGES

There are a number of challenges in trying to map a frame-based system like OntoSem to OWL. This section discusses some of the important issues that pertain to mapping of any frame-based system to Web representation such as OWL.

One of the challenges in building such a system is to bridge the gap between the knowledge representation features that are used by natural language processing systems and Semantic Web technologies. Some NLP systems such as OntoSem are supported by 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. Since OntoSem is used for natural language processing applications, it has a way of expressing defaults and exceptions. However there is no clear way of mapping defaults to OWL since OWL does not support non-monotonic reasoning and has an open world assumption. However, this is not as significant as it may appear, since these features are primarily used when creating classes and relations between them. For instance, a default for analyzing US-based news stories may be that if no specific nationality of a person is given, they are US Nationals. Systems such as SemNews will mostly deal with instance data where these features are not used, so the fact that information about the instance is represented in a system that does not handle defaults does not affect the ability of OntoSem to handle that data.

Knowledge sharing is a critical factor to enable agents on the Semantic Web to use this information extracted from NL text or be

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

able to provide information that can be used by NLP tools. This requires mapping across different ontologies and translating sentences from one representation to another. KQML (Finin, Fritzson, McKay & McEntire, 1994) and KIF (Genesereth, 1991) were two such attempts that developed protocols to enable sharing of large scale *knowledge bases*. Our system maps the OntoSem ontology to OWL and thus makes the framework sharable with other agents on the Web.

Ambiguity also is an issue when dealing with NL text. Human language can have ambiguity at both syntactic and semantic level. An example often discussed is *anaphora resolution*, which is the problem of identifying and resolving different references to the same named entity. OntoSem provides ways for handling such references and resolves these references, not just within a single document but across all the facts in its repository. This could have interesting applications in the Semantic Web domain, especially in resolving ambiguities inherent in FOAF (The friend of a friend project, n.d.) descriptions and data.

While some of the basic mapping rules have been developed, more needs to be done to identify and represent cardinalities, transitive, symmetric and inverse functional properties. These issues are being investigated.

There also were interesting challenges while mapping a large ontology such as OntoSem. Although we needed the capabilities of OWL Full to represent a more complete subset of OntoSem's features, the result was too large for OWL Full reasoners to process. One suggestion is to build mappings at different levels of expressivity, for example, we could have different versions of the OntoSem ontology for OWL Lite, DL and Full. Another approach would be to investigate the possibility of partitioning the ontology into different smaller ontologies.

OntoSem uses procedural attachments with concepts in the ontology and also in the TMRs. These are useful in performing tasks such as reference resolution, finding the relative time reference, and so forth. An important implication of the translation process is that currently it does not support any of these procedural attachments. It would be interesting to look into ways in which this information could be additionally incorporated either into the reasoner or the knowledge base of the agent itself.

EVALUATION

There are several dimensions along which this research can be evaluated. One possible evaluation is that of the underlying NLP system OntoSem. This has been reported on in (Nirenburg, Beale & McShane, 2004) and (Nirenburg & Kavi, 1996). We briefly summarize these below. However, we note that the primary purpose of this paper is not to report on OntoSem. The focus here is to build a system that can bridge between the Semantic Web and semantically rich NLP systems, such as OntoSem, and so the evaluation is a measure of how well we can translate.

The primary output of OntoSem is in the form of TMRs as described above. To evaluate these semantic representations of natural language, gold standards were created throughout the evaluation process, and a collection of *correctness* metrics were then applied to these gold standards along with the automatically created TMRs. The intermediate outputs of the various stages of OntoSem's processing (preprocessing, syntactic analysis, and semantic analysis) were inspected and corrected to produce gold standards; the amount of correction required was tabulated to produce an evaluation.

The emphasis on evaluation was placed on the primary function of OntoSem, semantic analysis. Four calculation metrics were combined to produce an overall evaluative figure for any given textual analysis (Nirenburg et al., 2004):

- **Match/Mismatch of TMR elements.** How many TMR elements did OntoSem properly identify, and how many were in error?
- **Weighted Score for WSD Complexity.** Of the mismatched TMR elements, how ambiguous was the mismatched lexical entry?

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

- **Weighted Score for WSD Distance.** Of the mismatched TMR elements, how similar are they to the target element, using the OntoSearch distance metric (Onyshkevych, 1997)?
- **Semantic Dependency Determination.** How many semantic dependencies in the TMR were properly matched, and how many were in error?

Combining the results of these four metrics, an automatically generated TMR can be compared to a gold standard TMR, producing an evaluation of OntoSem's performance.

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 also should involve 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); (The friend of a friend project, n.d.), Dublin Core (Miller & Brickley, 2001), OWL-S (OWL Web ontology language, 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 cannot 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.

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 (RDF validation service, n.d.) and the WonderWeb OWL Ontology Validator (Wonderweb owl ontology validator, n.d.).

There were a total of about 8,000 concepts in the original OntoSem ontology. The total number of triples generated in the translated

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

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.

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

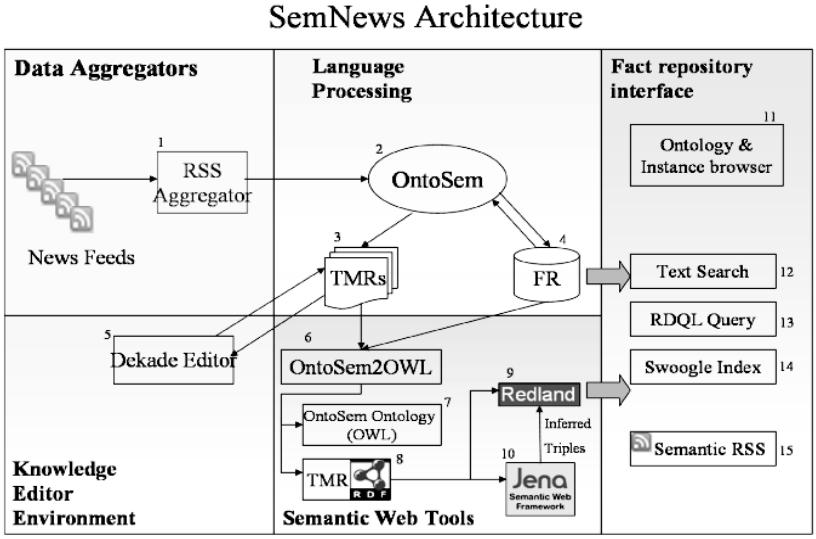
Using the Jena API it typically takes between 10 and 40 seconds to build the model, depending upon the reasoner employed. The computation of transitive closure and basic RDF schema differencing takes approximately 10 seconds on a typical workstation. The OWL Micro reasoner takes about 40 seconds, while 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.

APPLICATIONS

One of the motivations for integrating language understanding agents into the Semantic Web is to enable applications to use the information published in free text along with other Semantic Web data. SemNews[4] (Java, Finin & Nirenburg, 2006) is a semantic news service that monitors different RSS news feeds and provides structured representations of the meaning of news

Figure 4. The SemNews application, which serves as a testbed for our work, has a simple architecture. RSS (1) from multiple sources is aggregated and then processed by the OntoSem (2) text processing environment. This results in the generation of TMRs (3) and updates to the fact repository (4). The Dekade environment (5) can be used to edit the ontology and TMRs. OntoSem2OWL (6) converts the ontology and TMRs to their corresponding OWL versions (7,8). The TMRs are stored in the Redland triple store (9) and additional triples inferred by Jena (10).



Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

articles found in them. As new articles appear, SemNews extracts the summary from the RSS description and processes it with OntoSem. The resulting TMR is then converted into OWL. This enables us to *semantacize* the RSS content and provide live and up-to-date content on the Semantic Web. The prototype application also provides a number of interfaces that allow users and agents to query over the meaning representation of the text as expressed in OWL.

Figure 4 shows the basic architecture of SemNews. The RSS feeds from different news sources are aggregated and parsed. These RSS feeds also are rich in useful meta-data such as information on the author, the date when the article was published, the news category, and tag information. These form the explicit meta-data that is provided by the publisher. However there is a large portion of the RSS field that is essentially plain text and does not contain any semantics in them. It would be of great value if this text available in description and comment fields, for example, could be *semantacized*. By using Natural Language Processing tools such as OntoSem we can convert natural language text into a structured representation thereby adding additional metadata in the RSS fields. Once processed, it is converted to its Text Meaning Representation. OntoSem also updates its fact repositories to store the information found in the sentences processed. These facts extracted help the system in its future text analysis tasks.

An optional step of correction of the TMRs could be performed by means of the Dekade environment (English, 2006). This is helpful in correcting cases where the analyzers are not able to correctly annotate parts of the sentence. Corrections can be performed at both the syntactic processor and the semantic analyzer phase. The Dekade environment could also be used to edit the OntoSem ontology and lexicons or static knowledge sources.

As discussed in the previous sections, the meaning in these structured representations, also known as Text Meaning Representations, can be preserved by mapping them to OWL/RDF. The OWL version of a document's TMRs is stored in a Redland-based triple store, allowing other applications and users to 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 also indexed by the Swoogle Semantic Web Search system (Ding et al., 2004).

The following are some examples of queries that go beyond simple keyword searches.

- **Conceptually searching for content.** Consider the query—*Find all stories that have something to do with a place and a terrorist activity.*. Here the goal is to find the content or the story, but essentially by means of using ontological concepts rather than string literals. So for example, since we are using the ontological concepts here, we actually could benefit from resolving different kinds of terror events such as bombing or hijacking to a terrorist-activity concept.
- **Context-based querying.** Answering the query—*Find all the events in which 'George Bush' was a speaker*—involves finding the context and relation in which a particular concept occurs. Using named entity recognition alone, one can only find that there is a story about a named entity of the type person/human, however it is not directly perceivable as to what role the entity participated in. Since OntoSem uses deeper semantics, it not only identifies the various entities but also extracts the relations in which these entities or instances participate, thereby providing additional contextual information.
- **Reporting facts.** To answer a query like—*Find all politicians who traveled to Asia*— requires reasoning about people's roles and geography. Since we are using ontological concepts rather than plain text and we have certain relations like meronomy/part-of we could recognize that Colin Powel's trip to China will yield an answer.
- **Knowledge sharing on the Semantic Web.** Knowledge sharing is critical for

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

agents to reason on the Semantic Web. Knowledge can be shared by means of using a common ontology or by defining mappings between existing ontologies. One of the benefits of using a system like SemNews is that it provides a mechanism for agents to populate various ontologies with live and updated information. While FOAF has become a very popular mechanism to describe a person's social network, not everyone on the Web has a FOAF description. By linking the FOAF ontology to OntoSem's ontology, we could populate additional information and learn new instances of *foaf:person*, even though these were not published explicitly in foaf files but as plain text descriptions in news articles.

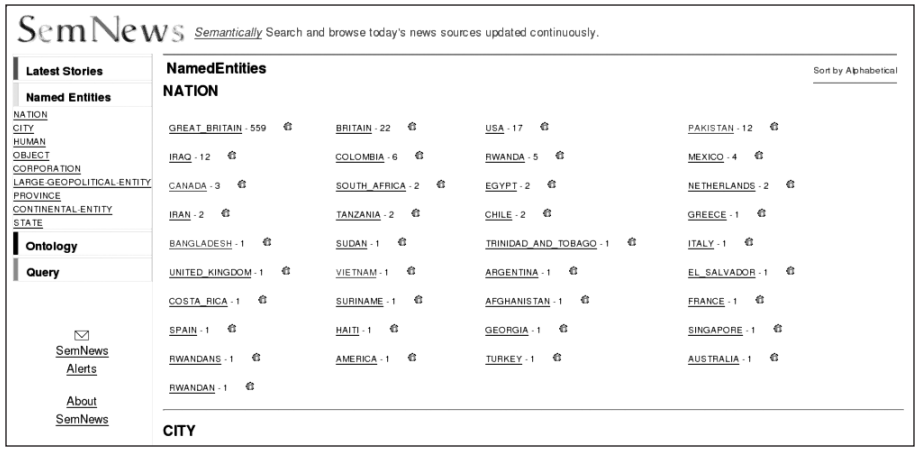
The SemNews environment also provides a convenient way for the users to query and browse the fact repository and triple store. Figure 5 shows a view that lists the named entities found in the processed news summaries. Using an ontology viewer the user can navigate through the news stories conceptually while viewing the instances that were found. The fact repository explorer provides a way to view the relations between different instances and see the news stories in which they were found. An advanced user may also query the triple store directly, using RDQL query language as shown in Figure 6. Additionally the system also can publish the RSS feed of the query results allowing users or agents to easily monitor new answers. This is a useful way of handling standing queries and finding news articles that satisfy a structured query.

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.

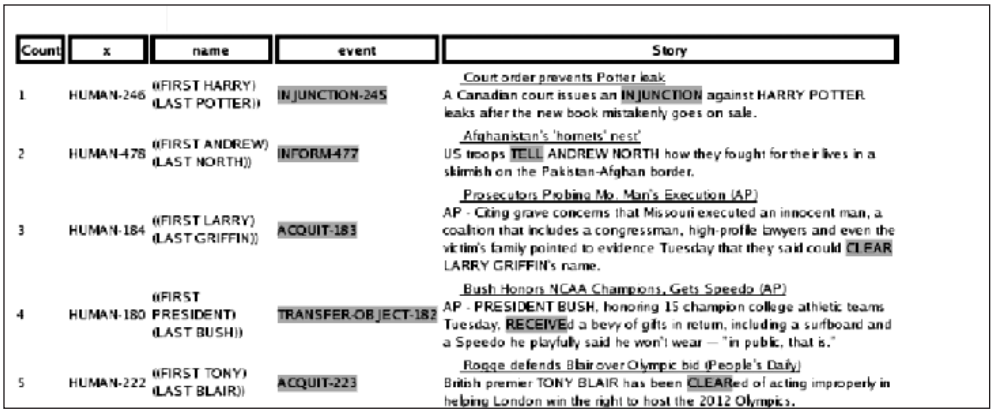
Second, the goal is not just to support translation between OntoSem, and a complete and faithful OWL version of OntoSem. It is unlikely

Figure 5 Various types of named entities can be identified and explored in SemNews.



Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

*Figure 6 This SemNews interface shows the results for the query—**Find all humans and what are they the beneficiaryof***



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.

that most Semantic Web content producers or consumers will use OntoSem's ontology. Rather, 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.

Language understanding agents could not only empower Semantic Web applications but also could create a space where humans and NLP tools would be able to make use of existing structured or semi-structured information available. The following are a few of the example application scenarios.

Semantic Annotation and Metadata Generation

The growing popularity of folksonomies and social bookmarking tools such as del.icio.us have demonstrated that light-weight tagging systems are useful and practical. Metadata also is available in RSS and ATOM feeds, while some use the Dublin Core ontology. Some NLP and statistical tools such as SemTag (Dill,

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Eiron, Gibson, Gruhl & Guha, 2003) and the TAP (Guha et al., 2003) project aim to generate semantically annotated pages from already existing documents on the Web. Using OntoSem in the SemNews framework, we have been able to demonstrate the potential of large scale semantic annotation and automatic metadata generation. Figure 3 shows the graphical representation of the TMRs, which also are exported in OWL and stored in a triple store.

Gathering Instances

Ontologies for the Semantic Web define the concepts and properties that the agents could use. By making use of these ontologies along with instance data agents can perform useful reasoning tasks. For example, an ontology could describe that a country is a subclass of a geopolitical entity and that a geopolitical entity is a subclass of a physical entity. Automatically generating instance data from natural language text and populating the ontologies could be an important application of such technologies. For example, in SemNews you can not only view the different named entities as shown in Figure 5, but you also can explore the facts found in different documents about that named entity. As shown in Figure 7, we could start browsing from an instance of the entity type 'NATION' and explore the various facts that were found in the text about that entity. Since OntoSem also handles referential ambiguities, it would be able to identify that an instance described in one document is the same as the instance described in another document.

Provenance and Trust

Provenance involves identifying source of information and tracking the history of where the information came. Trust is a measure of the degree of confidence one has for a source of information. While these are somewhat hard to quantify and are a function of a number of different parameters, there can be significant indicators of trust and provenance already present in the text and could be extracted by the agent. News reports typically describe some of the provenance information, as well as other metadata that can effect trust, such as temporal information. This type of information would be important in applications where agents need to make decisions based on the validity of certain information.

Reasoning

While currently reasoning on the Semantic Web is enabled by using the ontologies and Semantic Web documents, there could be potentially vast knowledge present in natural language. It would be useful to build knowledge bases that could not only reason based on explicit information available in them, but also could use information extracted from natural language text to augment their reasoning. One of the implications of using the information extracted from natural language text in reasoning applications is that agents on

Figure 7 Fact repository explorer for the named entity 'Mexico'. Shows that the entity has a relation 'nationality-of' with CITIZEN-235. Fact repository explorer for the instance CITIZEN-235 shows that the citizen is an agent-of an ESCAPE-EVENT.

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

the Semantic Web would need to reason in presence of inconsistent or incomplete annotations as well. Reasoning could be supported from not just Semantic Web data and natural language text but also based on provenance. Developing measures for provenance and trust also would help in deciding the degree of confidence that the reasoning engine may have in the using certain assertions for reasoning.

Ontology Enrichment

Knowledge acquisition is one of the most expensive steps in developing large scale Semantic Web applications. Even within the framework of OntoSem, the OntoSem ontology has been developed and perfected over years of research in linguistics, NLP and knowledge representation. In order to make the task of a knowledge engineer easier, we could possibly use the existing ontologies on the Semantic Web to suggest new concepts, relations, or even properties. As an example, consider the concept of fish, in OntoSem there are about four different varieties of fish that have been defined. We could now use a semantic search engine such as Swoogle (Ding et al., 2004) to find new types of fish and suggest some of the properties that could be used in order to describe fish in the ontology.

Natural Language Interface to Semantic Web

While the Semantic Web is primarily for use by machines, and the information available on it is in machine understandable format, the end goal is still to assist the human users in their tasks. Using technologies from question answering and language generation, it would be helpful to provide capabilities through which users can interact with their agent through natural language, thus reducing the cognitive load in formulating the task in a machine-readable format.

USING THE WEB FOR KNOWLEDGE ACQUISTION

In this paper we have reported on SemNews, a system, which uses OntoSem and the various processors and knowledge repositories available, along with the Web, to enhance the Semantic Web with knowledge learned through text analysis of RSS news feeds. However, we also can use the Web as a source for knowledge acquisition. This automated knowledge acquisition can be done in a few ways. First, when OntoSem encounters an unexpected input we can query the Web for documents related to such unknown lexical or ontological concepts. By processing the documents containing this concept, we can learn its meaning. Using the Web as a corpus, we have been able to automatically generate ontological concepts to some degree of accuracy, when given a target word (English et al., 2007). As an example, Table 4 shows some of the properties for the concept *'Hobbit'* learned by querying the Web.

From this data we learn that both humans and hobbits (whatever they might be) live, create things, and participate in elections. They both also can be rescued and killed. This kind of data can be used to generate a hypothesis

Table 4. This table compares selected properties of the concept 'HUMAN' to properties for the concept 'HOBBIT' automatically from the Web

Ontological Property	Values in HUMAN	Values in HOBBIT
AGENT-OF	LIVE, CREATE-ARTIFACT, ELECT, READ	LIVE, CREATE-ARTIFACT, ELECT
THEME-OF	RESCUE, MARRY, KILL	RESCUE, KILL
HAS-OBJ-AS-PART	HEAD	na

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

that hobbits are a class of sentient beings not unlike humans.

The second method is to import concepts and instance data available on the Semantic Web. The long-term goal of our ongoing research is indeed to automate knowledge acquisition and learning by reading. Specifically, we are working toward creating a system (an intelligent agent) that will be able to extract from text formal representations ready for use in automatic reasoning systems. These structures will reflect both instances and types of events, objects, relations, and agents' attitudes in the real world. The reasoning that such agents will be able to perform will support both general problem solving and, specifically, knowledge-based natural language processing, that is, the very process through which the agent learns from text.

Importing and integrating new conceptual knowledge from the Semantic Web remains a challenging problem. It involves addressing not only the ontology mapping problem, but also the difficulties in translating knowledge from OWL to a different knowledge representation system like OntoSem. Importing instance-level data, however, can be done with a very pragmatic approach and will result in information that can be extremely useful to a language processing system. For example, the *Semantic Wikipedia* project (Volkel, Krotzsch, Vrandecic, Haller & Studer, 2006) exposes some of the information found in Wikipedia in RDF using a small set of ontologies. This can be easily mined, for example, to enrich OntoSem's onomasticon by mapping key classes and properties into the OWL version of OntoSem and ultimately to OntoSem's native representation system.

In any case, the benefit is clear. Using the Web, and the Semantic Web as a corpus, OntoSem can learn new concept instances, ontological concepts, and lexical entries by reading. As the effect of increased static knowledge resources on OntoSem is that of producing better TMRs, the benefit is circular. The more OntoSem learns, the better it becomes at learning. By using a fully open corpus, containing material on nearly everything imaginable, we will soon be able to *close the loop.*

CONCLUSION

Natural language processing agents can provide a service by analyzing text documents on the Web and publishing Semantic Web annotations and documents that 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 translating knowledge and information from their native representation systems to 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, OntoSem2OWL 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 on the Web encoded in OWL.

REFERENCES

Baker, C., Fillmore, C., & Lowe, J. (1998). The Berkeley FrameNet Project. *Proceedings of the*

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

36th Conference on Computational Linguistics-Volume 1. Morristown, NJ, USA (pp. 86-90). Association for Computational Linguistics..

Beale, S., Lavoie, B., McShane, M., Nirenburg, S., & Korelsky, T. (2004). Question answering using ontological semantics. *Proceedings of ACL-2004 Workshop on Text Meaning and Interpretation* (pp. 41-48). Association for Computational Linguistics.

Beale, S., Nirenburg, S., & Mahesh, K. (1995). Semantic analysis in the mikrokosmos machine translation project. *Proceedings of the 2nd Symposium on Natural Language Processing* (pp. 297-307).

Beltran-Ferruz, P., Gonzalez-Caler, P., & P. Gervas (2004a). Converting frames into OWL: Preparing Mikrokosmos for linguistic creativity. *LREC Workshop on Language Resources for Linguistic Creativity.*

Beltran-Ferruz, P., Gonzalez-Caler, P., & Gervas, P. (2004b, July). Converting Mikrokosmos frames into description logics. *RDF/RDFS and OWL in Language Technology: 4th ACL Workshop on NLP and XML*. Association for Computational Linguistics.

Cost, R. S., Finin, T., Joshi, A., Peng, Y., Nicholas, C., Soboroff, I., Chen, H., Kagal, L., Perich, F., Zou, Y., & Tolia, S. (2002). ITtalks: A case study in the Semantic Web and DAML+OIL. *IEEE Intelligent Systems, 17* (1), 40-47.

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities, 36* (2), 223-254.

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

Dill, S., Eiron, N., Gibson, D., Gruhl, D., & Guha, R. (2003). Semtag and seeker: Bootstrapping the Semantic Web via automated semantic annotation. *Proceedings of the Twelfth International Conference on World Wide Web*. New York, NY, USA. (pp. 178-186). ACM Press.

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

Dou, D., McDermott, D., & Qi, P. (2005). *Ontologies for agents: Theory and experiences*, Chap. Ontology translation by ontology merging and automated reasoning (pp. 73-94). Birkhäuser Basel.

English, J. (2006). DEKADE II: An environment for development and demonstration in natural language processing. (Master's thesis, University of Maryland, Baltimore County).

English, J., & Nirenburg, S. (2007). Ontology learning from text using automatic ontological-semantic text annotation and the Web as the corpus. *AAAI Spring Symposium on Machine Reading*. AAAI Press.

Fallside, D. C., & Walmsley, P. (2004). *XML schema part 0: Primer* second edition (W3C recommendation). W3C. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

Farwell, D., Helmreich, S., Dorr, B., Habash, N., Reeder, F., Miller, K., Levin, L., Mitamura, T., Hovy, E., Rambow, O., et al. (2004). Interlingual annotation of mMultilingual text corpora. *Proceedings of the North American Chapter of the Association for Computational Linguistics Workshop on Frontiers in Corpus Annotation* (pp. 55-62). Association for Computational Linguistics.

Fensel, D., van Harmelen, F., & Akkermans, H. (2000). Ontoknowledge: Ontology-based tools for knowledge management. *Proceedings of the eBusiness and eWork Conference*. Springer Verlag.

Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994). KQML as an Agent Communication Language. In N. Adam, B. Bhargava, & Y. Yesha (Eds.), *Proceedings of the Third International Conference on Information and Knowledge Management* (CIKM'94). Gaithersburg, MD, USA. (pp. 456-463). ACM Press.

Genesereth, M. (1991). Knowledge interchange format. *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning* (pp. 599-600). Morgan Kaufmann.

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics, 28* (3), 245-288.

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

Hobbs, J. R., & 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., & Karger, D. R. (2005, May). Thresher: Automating the unwrapping of semantic content from the Wworld Wide Web. *Proceedings of the Fourteenth International World Wide Web Conference*.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). Swrl: A Ssemantic Web rule language combining owl and ruleml. *World Wide Web Consortium Member Submission.*

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

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

Java, A., Finin, T., & Nirenburg, S. (2005, November). Integrating language understanding agents into the semantic web. In T. Payne, & V. Tamma (Eds.), *Proceedings of the AAAI Fall Symposium on Agents and the Semantic Web*. AAAI Press.

Java, A., Finin, T., & Nirenburg, S. (2006, January). Text understanding agents and the Semantic Web. *Proceedings of the 39th Hawaii International Conference on System Sciences*. Kauai, HI.

Kalyanpur, A., Parsia, B., & Hendler, J. (2005). A tool for working with Web ontologies. *International Journal on Semantic Web and Information Systems, 1* (1), 36-49.

Kingsbury, P., Palmer, M., & Marcus, M. (2002). Adding semantic annotation to the penn treebank. *Proceedings of the Human Language Technology Conference.*

Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Journal of Web Semantics, 2* (1), 49-79.

Krueger, W., Nilsson, J., Oates, T., & Finin, T. (2004). Agent mediated knowledge management, Chap. *Automatically Generated DAML Markup for Semistructured Documents*. (LNCS 4150).Springer.

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

McShane, M., Nirenburg, S., Beale, S., & O'Hara, T. (2005, June). Semantically rich human-aided machine annotation. *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*. Ann Arbor, Michigan. (pp. 68-75). Association for Computational Linguistics.

Miller, D. B. E., & Brickley, D. (2001). Expressing simple dublin core in RDF / XML. Dublin Core *Metadata Initiative Recommendation*.

Muslea, I. A., Minton, S., & 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., Beale, S., & McShane, M. (2004). Evaluating the performance of the ontosem semantic analyzer. *Proceedings of the ACL Workshop on Text Meaning Representation*. Association for Computational Linguistics.

Nirenburg, S., & Raskin, V. (2001). Ontological semantics, formal ontology, and ambiguity. FOIS '01: *Proceedings of the International Conference on Formal Ontology in Information Systems*. New York, NY, USA. (pp. 151-161). ACM Press.

Nirenburg, S., & Raskin, V. (2005). Ontological semantics. *MIT Press*.

Onyshkevych, B. (1997). Ontosearch: Using an ontology as a search space for knowledge based text processing. (PhD thesis, Carnegie Mellon University).

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

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

Philpot, A., Hovy, E., & Pantel, P. (2005). The Omega Ontology. *Proceedings of the ONTOLEX Workshop at the International Conference on Natural Language Processing.* Jeju Island, South Korea.(pp. 59-66).

R.V. Guha, & McCool, R. (2003). TAP: A Semantic Web toolkit. **Journal of Web Semantics, 1** (1), 81-87.

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

Schlangen, D., Stede, M., & Bontas, E. P. (2004, July). Feeding owl: Extracting and representing the content of pathology reports. *RDF/RDFS and OWL in Language Technology: 4th ACL Workshop on NLP and XML*. Association for Computational Linguistics.

Sergei Nirenburg, K. M. (1996). Measuring semantic coverage. *17th International Conference on Computational Linguistics*, COLING-96. Association for Computational Linguistics.

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

van Harmelen, F., & McGuinness, D. L. (2004). *OWL web ontology language overview (W3C recommendation)*. W3C. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

Volkel, M., Krotzsch, M., Vrandecic, D., Haller, H., & Studer, R. (2006). Semantic Wikipedia. *Proceedings of the 15th International Conference on World Wide Web* (pp. 585-594).

Witbrock, M., Panton, K., Reed, S., Schneider, D., Aldag, B., Reimers, M., & Bertolo, S. (2004, November). Automated OWL Annotation Assisted by a Large Knowledge Base. *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.

ENDNOTES

1 http://www.ldc.upenn.edu/Projects/ACE/.

2 Closed classes are lexical classes with a relatively fixed set of words to which new ones normally are not added, such as (for English) prepositions, conjunctions, and pronouns.

3 A polysemous word is one with multiple senses whose meanings are related. An Example in English is *mole*, which can be a mammal that lives in underground burrows or a spy.

4 http://semnews.umbc.edu.

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Copyright © 2007, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.