Using a Natural Language Understanding System to Generate Semantic Web Content

Akshay Java, Sergei Nirenburg, Marjorie McShane, Timothy Finin, Jesse English, Anupam Joshi
University of Maryland, Baltimore County, Baltimore MD 21250
{aks1,sergei,marge,finin,english1,joshi}@umbc.edu
http://semnews.umbc.edu
http://ebiquity.umbc.edu/project/semnews

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 results 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.

I. Introduction

A core goal of semantic web technology is to bring progressively more meaning to 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 (because annotations must be made in a formal language of which, unlike natural languages, people are not native speakers). Developing interactive tools for annotation is a problematic undertaking because it is not known whether they will be in actual demand. A number of semantic web practitioners maintain that the desire to have their content available on the semantic web will compel a sufficient number of people to spend the time and effort on manual annotation. However, even if such a desire materializes, people will simply 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 (IE) techniques – a reasonably good level of capturing named entities with a somewhat less successful categorization of such entities (e.g., disambiguating Jordan between a first name and 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 capabilities of IE, 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-grain 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 IE 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 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 will also 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 paper describes our initial work on responding to the needs and leveraging the offerings of the web by merging knowledge-oriented NLP 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 [1] 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, 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 KR 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 resources (its 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 [2].

The remainder of this paper 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. Section IV provides an overview of the architecture of our implemented system and describes the approach used and major issues discovered in using it to map knowledge between OntoSem and OWL. Section V outlines some of the larger issues and challenges we expect to encounter. While this work is still in a preliminary stage, we offer some thoughts on how some components can be evaluated in Section VI. Section VII describes the SemNews application testbed and some general application scenarios we have explored to motivate and guide our research. Finally, we offer some concluding remarks in section IX.

II. Related Work

Among past projects that have addressed semantic annotation are the following:

- Gildea and Jurafsky [3] 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 (produced by the FrameNet project). When tasked to segment constituents and identify their semantic roles (with fillers being undisambiguated 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 goal of the Interlingual Annotation of Multilingual Text Corpora project [1] is to create a syntactic and semantic annotation representation methodology and test it out on six 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 light semantics essentially, thematic roles.
- In the ACE project [2], 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: e.g., 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. [4] report: 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.

[1] http://aitc.aitcnet.org/nsf/iamtc/

[2] http://www.ldc.upenn.edu/Projects/ACE/intro.html

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 [5] was to find the various named entities in text. Using OntoSem, we aim to not only 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 [6], [7]. 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. [8] 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. [9] 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 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 then mapped back into a Semantic Web representation.

TAP [10] 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. [11] developed an application that learned to extract information from talk announcements from training data using an algorithm based on Stalker [12]. 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 [13].

The Haystack Project has developed system [14] 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. 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 [15] 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.

III. ONTOSEM

Ontological Semantics (OntoSem) is a theory of meaning in natural language text [1]. 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* 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 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 below:

- 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.
- An OntoSem lexicon whose entries contain syntactic and semantic information (linked through variables) 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 can also describe word meaning extra-ontologically, for example, in terms of modality, aspect or time (see McShane and Nirenburg 2005 for in-depth discussion of the lexicon/ontology connection). The current English lexicon contains approximately 30,000 senses, including most closed-class items and many of the most frequent and polysemous verbs, as selected through corpus analysis. The base lexicon is expanded at runtime using an inventory of lexical (e.g., derivational-morphological) rules.

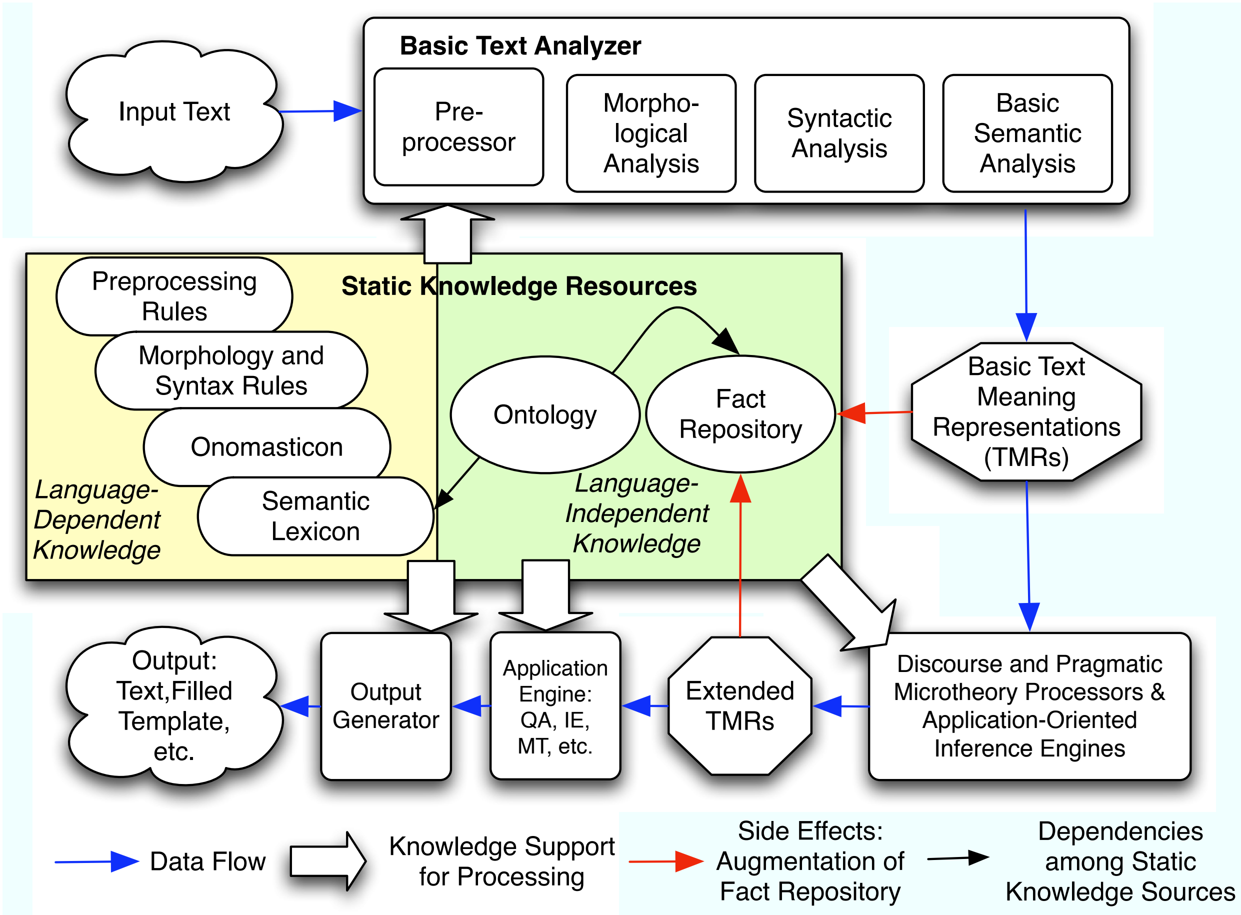


Fig. 1. High level view of OntoSem

- An onomasticon, or lexicon of proper names, which contains approximately 350,000 entries.
- A fact repository, which contains remembered instances of ontological concepts (e.g., SPEECH-ACT-3366 is the 3366th instantiation of the concept SPEECH-ACT in the memory of a text-processing agent). The fact repository is not used in the current experiment but will provide valuable semantically-annotated context information for future experiments.
- The OntoSem syntactic-semantic analyzer, which performs preprocessing (tokenization, named-entity and acronym recognition, etc.), morphological, syntactic and semantic analysis, and the creation of TMRs.
- The TMR language, which is the metalanguage for representing text meaning.

OntoSem knowledge resources have been acquired by trained acquirers using a broad variety of efficiency-enhancing tools 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, etc. OntoSems DEKADE environment [16] facilitates both knowledge acquisition and semi-automatic creation of gold standard TMRs, which can be also viewed as deep semantic text annotation.

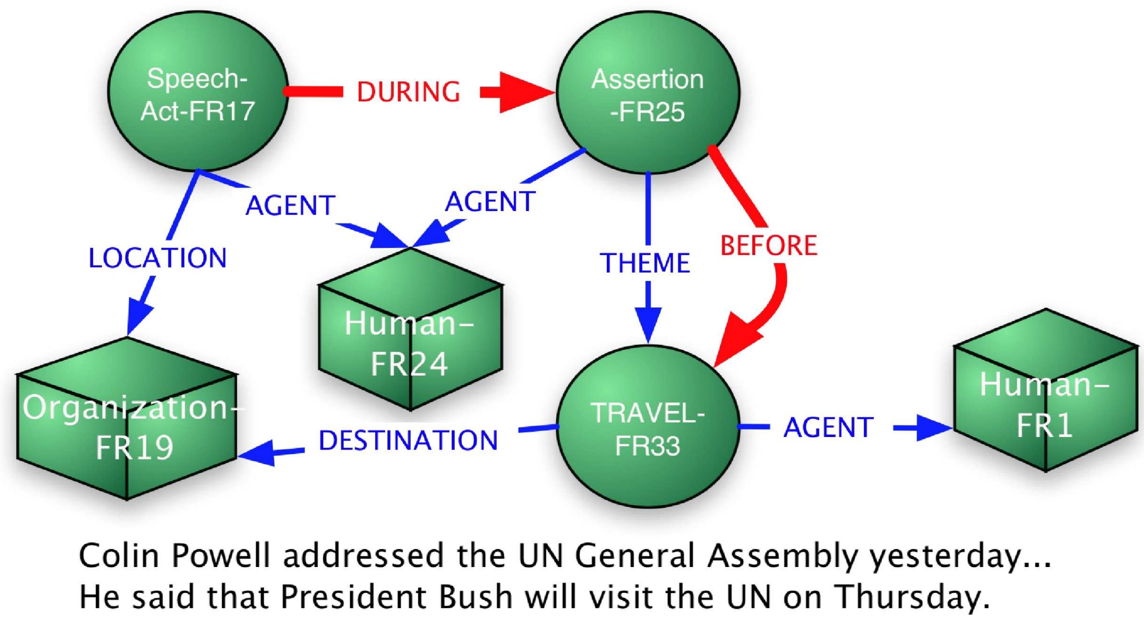


Fig. 2. OntoSem goes through several basic stages in converting a sentence into a text meaning representation (TMR).

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 basic steps in processing the sentence to extract the meaning representation is show in figure 1. The preprocessor deals with identifying sentence and word boundaries, part of speech tagging, recognition of named entities and dates, etc. 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 [17], machine translation [18] 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 knowledge base of OntoSem. As an example the sentence: " *He (Colin Powell) asked the UN to authorize the war*" is converted to the TMR shown in Figure 3. A more detailed description of OntoSem and its features is available in [19] and [20].

IV. Mapping OntoSem to OWL

We have developed **OntoSem2OWL** [21] 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

```
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.

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

the concepts in some other ontology [22]. 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)))
```

	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 I

TABLE SHOWING HOW OFTEN EACH OF THE CLASS RELATED CONSTRUCTS ARE USED

```
  (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.

A. 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 I shows the usage of each of these features in OntoSem.

B. 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"/>
```

	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 II

TABLE SHOWING HOW OFTEN EACH OF THE PROPERTY RELATED CONSTRUCTS ARE USED

```
      <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 II describes the typical usages of the property related constructs in OntoSem.

C. 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 III 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 [23].
- *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 [24]. 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 [25] 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 III that relatively they are used less frequently. Hence in our use cases, ignoring these facets does not lose a lot of information.

D. 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 [26] in OWL. The following is an example of how the property *age* could be represented in OWL using *xsd:restriction*:

	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 III

TABLE SHOWING HOW OFTEN EACH OF THE FACETS ARE USED

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

E. 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>
```

V. 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. Typically 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 nonmonotonic reasoning and has an open world assumption.

Knowledge sharing is a critical factor to enable agents on the Semantic Web to use this information extracted from NL text or be 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 [27] and KIF [28] 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 is also 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 [29] descriptions and data.

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

There were also 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, etc. 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.

VI. 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 [29], Dublin Core [30], OWL-S [31], OWL-time [32], 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 [33] 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 [34] and the WonderWeb OWL Ontology Validator [35].

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 [36] 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.

VII. 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 [3] [37] is a semantic news service that monitors different RSS news feeds and provides structured representations of the meaning of news 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 which 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 are also 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 (NLP) 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 (TMR). 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 [38]. This is helpful in correcting cases where the analyzers are not able to correctly annotate parts of the sentence. Corrections

[3] http://semnews.umbc.edu

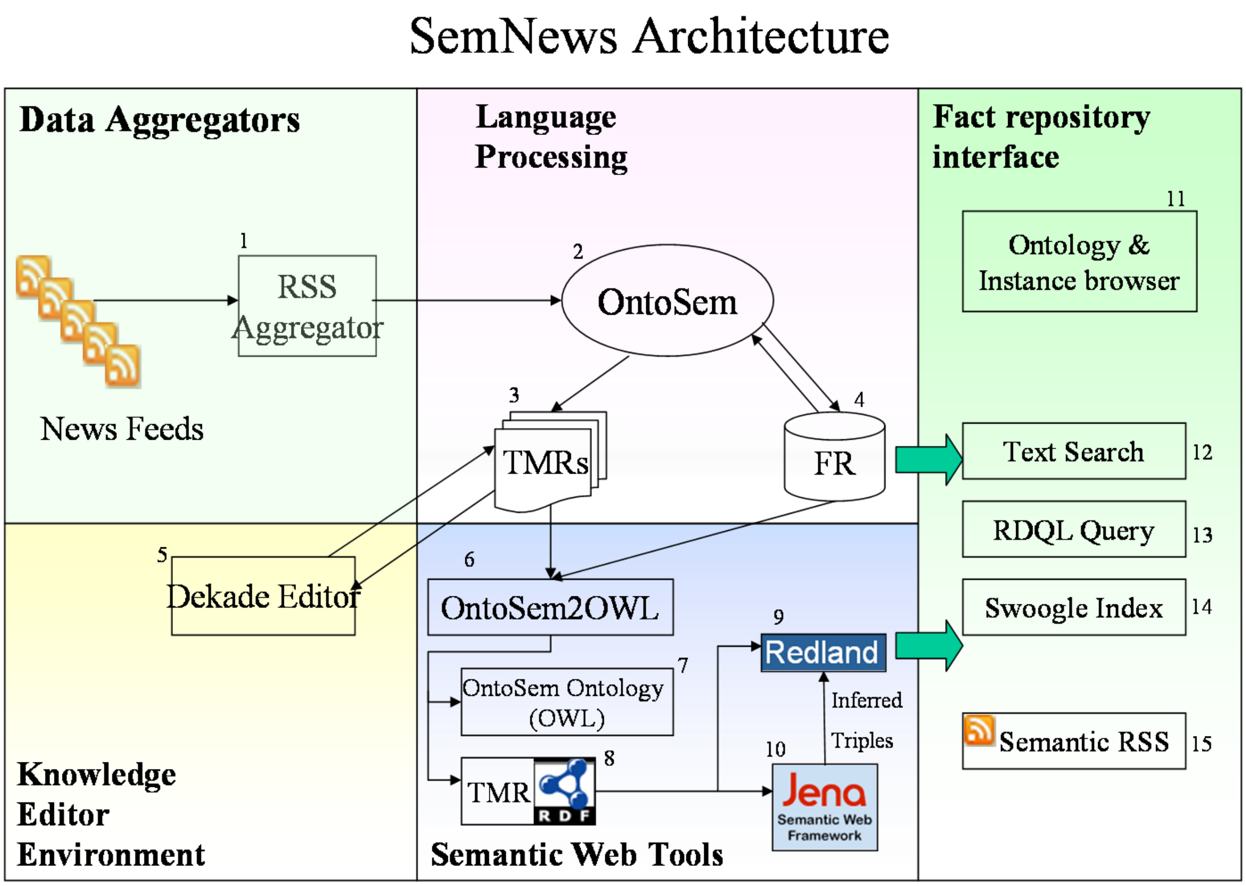


Fig. 4. The SemNews application, which serves as a testbed for our work, has a simple architecture. RSS (1) from multiple sources is aggregated and then processed by the OntoSem (2) text processing environment. This results in the generation of TMRs (3) and updates to the fact repository (4). The Dekade 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). There are also multiple viewers for searching and browsing the fact repository and triple store.

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 (TMR), 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 [39].

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 could actually 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 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 6 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 7. Additionally the system can also 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. 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, 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 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.

A. 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 is also available in RSS and ATOM feeds, while some use the Dublin Core ontology. Some NLP and statistical tools such as SemTag[40] and the TAP[10] 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 are also exported in OWL and stored in a triple store.

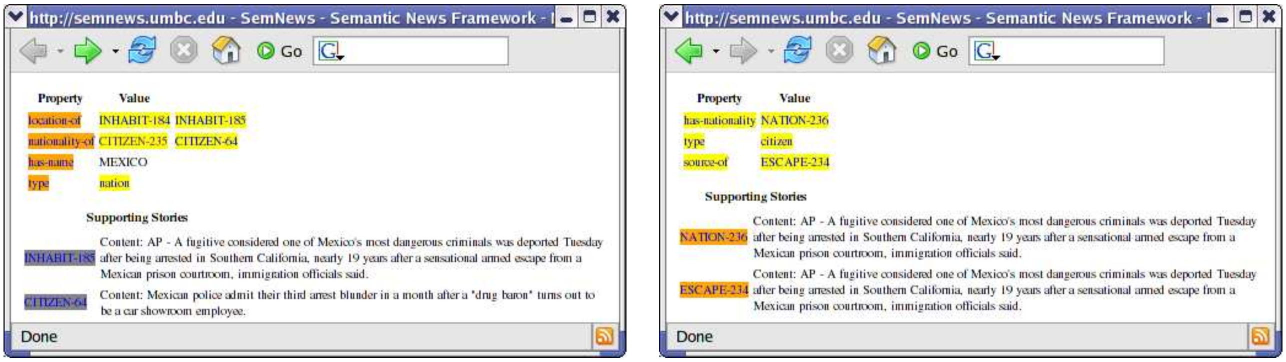


Fig. 5. Fact repository explorer for the named entity 'Mexico'. Shows that the entity has a relation 'nationality-of' with CITIZEN-235. Fact repository explorer for the instance CITIZEN-235 shows that the citizen is an agent-of an ESCAPE-EVENT.

B. 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 6 but also explore the facts found in different documents about that named entity. As shown in VII-B, 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.

C. Provenance and Trust

Provenance involves identifying source of information and tracking the history of where the information came from. 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 report 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.

D. 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

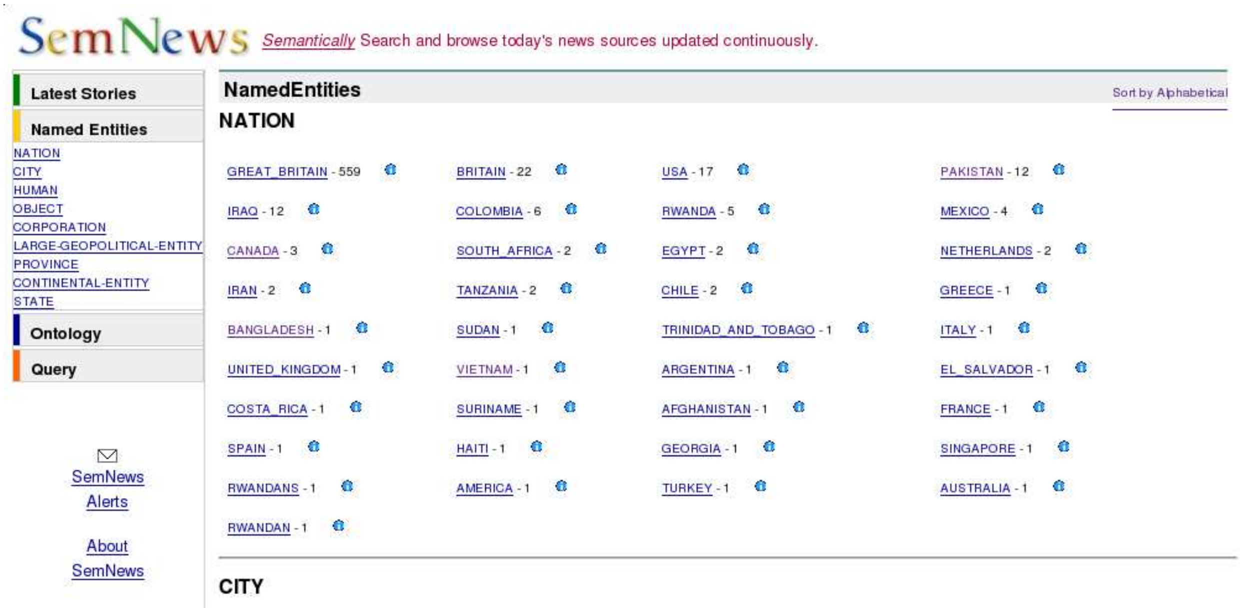


Fig. 6. Various types of named entities can be identified and explored in SemNews.

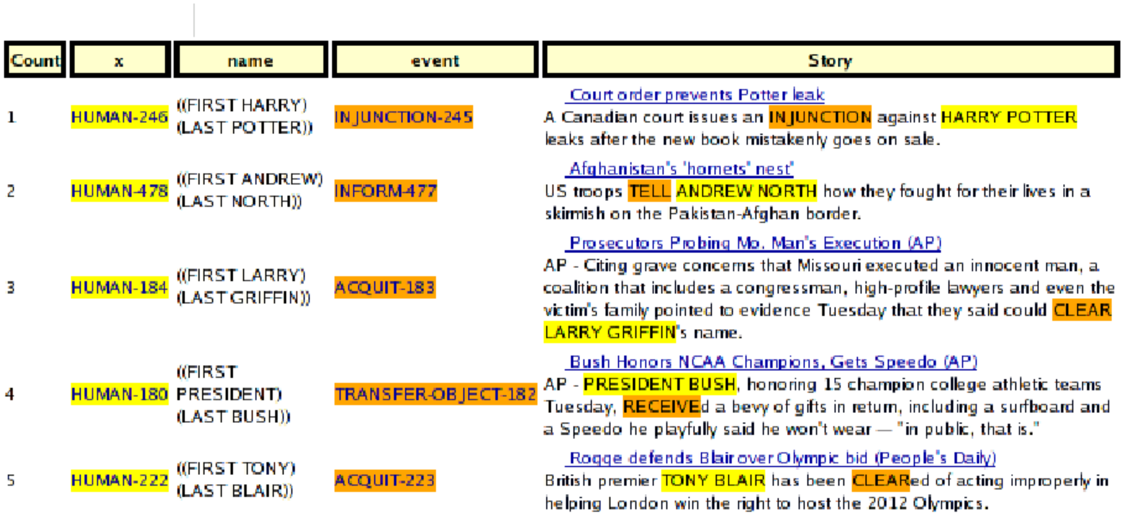
that could not only reason based on explicit information available in them, but also use information extracted form natural language text to augment their reasoning. One of the implications of using the information extracted from natural language text in reasoning applications is that agents on the Semantic 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 would also help in deciding the degree of confidence that the reasoning engine may have in the using certain assertions for reasoning.

E. 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 4 different varieties of fish that have been defined. We could now use a semantic search engine such as Swoogle [39] to find new types of fish and suggest some of the properties that could be used in order to describe fish in the ontology.

F. 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



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.

Fig. 7. This SemNews interface shows the results for query "Find all humans and what are they the beneficiary-of"

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

TABLE IV

COMPARISON OF SELECTED PROPERTIES OF HUMAN TO PROPERTIES FOR THE CONCEPT 'HOBBIT' AUTOMATICALLY LEARNT FROM THE WEB

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.

VIII. Using the Web for Knowledge Acquisition

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 can also use 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 [2]. As an example the following table shows some of the properties for the concept *'Hobbit'* learnt by querying the web.

The Second method is to import concepts and instance data available on the Semantic Web.

Automating knowledge acquisition for use in automatic reasoning systems in a variety of applications has long been recognized as the Holy Grail of AI. The long-term goal of our ongoing research is indeed 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 NLP, that is, the very process through which the agent learns from text.

In either 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.

IX. 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

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

[2] J. English and S. Nirenburg, "Ontology learning from text using automatic ontological-semantic text annotation and the web as the corpus." in *Submitted to AAAI-07 Spring Symposium on Machine Reading. (in submission)*, 2007.

[3] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," in *Computational Linguistics*, pp. 245–288.

[4] P. Kingsbury, M. Palmer, and M. Marcus, "Adding semantic annotation to the penn treebank," in *Proceedings of the Human Language Technology Conference (HLT'02)*, 2002.

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

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

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

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

[9] D. Schlangen, M. Stede, and E. P. Bontas, "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*, July 2004.

[10] R.V.Guha and R. McCool, "TAP: A semantic web toolkit," *Semantic Web Journal*, October 2003.

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

[12] I. A. Muslea, S. Minton, and C. Knoblock, "Hierarchical wrapper induction for semistructured information services," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 4, no. 1/2, pp. 93–114, 2001.

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

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

[15] M. Witbrock, K. Panton, S. Reed, D. Schneider, B. Aldag, M. Reimers, and S. Bertolo, "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*, November 2004.

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

[17] B. L. M. M. S. N. Beale, S. and T. Korelsky, "Question answering using ontological semantics."

[18] S. Beale, S. Nirenburg, and K. Mahesh, "Semantic analysis in the mikrokosmos machine translation project," 1995.

[19] S. Nirenburg and V. Raskin, "Ontological semantics," 2005.

[20] "Institute for language and information technologies," http://ilit.umbc.edu/.

[21] A. Java, T. Finin, and S. Nirenburg, "Integrating language understanding agents into the semantic web," in *Proceedings of the AAAI Fall Symposium on Agents and the Semantic Web*, T. Payne and V. Tamma, Eds., November 2005.

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

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

[24] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, "From shiq and rdf to owl: the making of a web ontology language." *J. Web Sem.*, vol. 1, no. 1, pp. 7–26, 2003.

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

[26] "Xml schema part 0: Primer," World Wide Web Consortium Specification, 2004, see http://www.w3.org/TR/xmlschema-0/.

[27] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an Agent Communication Language," in *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, N. Adam, B. Bhargava, and Y. Yesha, Eds. Gaithersburg, MD, USA: ACM Press, 1994, pp. 456–463. [Online]. Available: citeseer.csail.mit.edu/article/finin95kqml.html

[28] M. Genesereth, "Knowledge interchange format," in *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1991, pp. 599–600.

[29] "The friend of a friend(foaf) project," http://www.foaf-project.org/.

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

[31] "OWL web ontology language for services (OWL-S)," A W3C submission, 2004, http://www.w3.org/Submission/2004/07/.

[32] J. R. Hobbs and F. Pan, "An ontology of time for the semantic web," *ACM Transactions on Asian Language Processing (TALIP)*, vol. 3, no. 1, pp. 66–85, March 2004, special issue on Temporal Information Processing.

[33] B. McBride, "Jena: Implementing the RDF model and syntax specification," in *Proceedings of the WWW2001 Semantic Web Workshop*, 2001.

[34] "RDF validation service," http://www.w3.org/RDF/Validator/.

[35] "Wonderweb owl ontology validator," http://phoebus.cs.man.ac.uk:9999/OWL/Validator.

[36] A. Kalyanpur, B. Parsia, and J. Hendler, "A tool for working with web ontologies," in *In Proceedings of the International Journal on Semantic Web and Information Systems*, vol. 1, no. 1, January-March 2005. [Online]. Available: \url{http://www.mindswap.org/papers/Swoop-Journal.pdf}

[37] A. Java, T. Finin, and S. Nirenburg, "Text understanding agents and the Semantic Web," in *Proceedings of the 39th Hawaii International Conference on System Sciences*, Kauai HI, January 2006.

[38] J. English, "Dekade ii: An environment for development and demonstration in natural language processing," 2006.

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

[40] S. Dill, N. Eiron, D. Gibson, D. Gruhl, and R. Guha, "Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation," in *Proceedings of WWW2003*, 2003.