Automating Ontological Annotation with WordNet

Antonio Sanfilippo and **Stephen Tratz** and **Michelle Gregory** and **Alan Chappell** and **Paul Whitney**
Christian Posse and **Patrick Paulson** and **Bob Baddeley**
Ryan Hohimer and **Amanda White**
Pacific Northwest National Laboratory
902 Battelle Blvd, Richland, PO Box 999
WA 99352, USA
{Antonio.Sanfilippo, Stephen.Tratz, Michelle.Gregory, Alan.Chappell, Paul.Whitney}@pnl.gov
{Christian.Posse, Patrick.Paulson, Bob.Baddeley, Ryan.Hohimer, Amanda.White}@pnl.gov

Abstract

Semantic Web applications require robust and accurate annotation tools that are capable of automating the assignment of ontological classes to words in naturally occurring text (ontological annotation). Most current ontologies do not include rich lexical databases and are therefore not easily integrated with word sense disambiguation algorithms that are needed to automate ontological annotation. WordNet provides a potentially ideal solution to this problem as it offers a highly structured lexical conceptual representation that has been extensively used to develop word sense disambiguation algorithms. However, WordNet has not been designed as an ontology, and while it can be easily turned into one, the result of doing this would present users with serious practical limitations due to the great number of concepts (synonym sets) it contains. Moreover, mapping WordNet to an existing ontology may be difficult and requires substantial labor. We propose to overcome these limitations by developing an analytical platform that (1) provides a WordNet-based ontology offering a manageable and yet comprehensive set of concept classes, (2) leverages the lexical richness of WordNet to give an extensive characterization of concept class in terms of lexical instances, and (3)

integrates a class recognition algorithm that automates the assignment of concept classes to words in naturally occurring text. The ensuing framework makes available an ontological annotation platform that can be effectively integrated with intelligence analysis systems to facilitate evidence marshaling and sustain the creation and validation of inference models.

1 Introduction

Ontological annotations identify real-world entities alongside properties and relations that characterize the entities' attributes and role in their textual context, with respect to a reference ontology. Adding these annotations to unstructured or semi-structured data is a basic requirement to make Semantic Web technologies work (Fensel et al. 2003, pp. 1-25; Klein et al. 2003). For example, the availability of ontologically annotated documents is crucial in enabling the shift from keyword-based queries and navigation by predefined links to semantic-driven search and navigation behaviors that can be effectively handled by automatic agents in Semantic Web applications (Maedche et al. 2003; Broekstra et al. 2003).

Ontologies such as Cyc[1] and SUMO[2] therefore represent a pivotal element for Semantic Web applications as they make available a knowledge representation language amenable to logical reasoning and a dictionary of classes and relations that Web Services can use to describe content and reason about it. However, linking words from naturally occurring text to entity and relationship classes in an ontology is often problematic. Ontologies do not usually integrate a rich enough set of lexical instances that exemplify the real-world entity and relationship tokens for their classes. Without such lexicons, gazetteers and thesauri, the automation of the ontological annotation process is impossible as there is no way of establishing how a word token (e.g. *gun*) can be related to an ontological class (e.g. #Weapon).

Manual ontological annotation may provide a viable solution in some limited application domains, but it is simply not a choice for applications which require processing large document collections. For example, imagine adding semantic tags to each newswire that a news service receives daily or, even worse, tackling the huge repositories of legacy newswire data. The daunting proportions of such an annotation task would constitute a pre-emptive bottleneck under both time and cost considerations. Ultimately, automatic ontological annotation is the only viable alternative. The minimal requirements to make such an alternative available are

- to establish reliable and cost-effective ways of linking lexical database entries to concept classes in an ontology, and
- to use word sense disambiguation algorithms that reliably relate words in naturally occurring text to those lexical database entries that have been linked to ontological classes.

The goal of this paper is to show how these two requirements can be satisfied by

- leveraging the hierarchical structure of WordNet to transform WordNet into an ontology where a relatively small number of top- and mid-level synonyms sets

[1] http://www.opencyc.org/.
[2] http://ontology.teknowledge.com/.

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 85–93.

are selected as concept classes, with all synonym sets defined as instances for such classes, and

- using WordNet-based word sense disambiguation algorithms to resolve ambiguities concerning the assignment of a word token (e.g. *conduct* in the context *conduct a nuclear program*) to its appropriate class (e.g. *manage* as opposed to *behave, perform*, or *transmit).*

The ensuing framework provides an ontological annotation platform that can be effectively integrated with intelligence analysis systems to facilitate evidence marshaling and sustain the creation and validation of inference models.

2 Background

Several formalizations of WordNet as an OWL ontology have been developed during the last few years[3] and a WordNet Task Force has been created within the W3C Semantic Web Best Practices and Deployment Working Group[4] to support the deployment of WordNet and similarly structured lexica in RDF/OWL. One of the main problems with turning WordNet into an OWL ontology is the sheer number of resulting concept classes. WordNet 2.0 has some 115,000 synonym sets. If each synonym set is formalized as a concept class, the ensuing number of classes would just be too large and therefore impractical for a real-world application. Moreover, it is not clear whether such a large number of lexical concept classes is needed for applications such as semantic-based search and navigation. While it is important to have as wide a lexical coverage as possible, such an objective can be simply achieved by linking a large number of word senses (e.g. the 115,000 synonym sets in WordNet) to a more manageable number of concept classes.

Knight & Luk (1994) provide one of the earliest attempts at linking a large lexical database such as WordNet to an ontology derived from merging the PENMAN Upper Model and ONTOS (see also Hovy 1998). Such a mapping involves breaking WordNet into 200 hundred pieces and merging each manually into the merged PENMAN Upper Model and ONTOS ontology. Niles (2003) offers a more recent example of the same endeavor by developing a methodology to link SUMO classes to WordNet synonym sets manually; to date, the full WordNet 1.6 has been mapped to SUMO. Other examples are the Cyc-to-WordNet mapping that includes some 8,000 WordNet noun synsets, as reported in O'Hara et al. (2003) and the ongoing OntoWordNet Project at the Laboratory for Applied Ontology in the Italian National Research Council[5] (Cangemi et al., 2003).

These must all be regarded as important achievements as they greatly enhance the utility of influential ontologies. However, in spite of the considerable amount of work done, the accuracy of mapping methodologies developed so far is yet unknown. Minimally, an evaluation of mapping results would involve correlating choices made by several annotators for a representative subset of WordNet-SUMO mappings in order to compute inter-annotators' agreement. However, such an evaluation is yet to be performed. Moreover, regardless of their reliability, the mapping methodologies developed so far cannot be seen as providing a viable general solution for integrating ontologies with large lexical databases such as WordNet. Because of the great number of synonym sets, the task of mapping WordNet to existing ontologies is simply too costly and time-demanding to be carried out manually. In theory, the inheritance structure of WordNet can be used to reduce the number of nodes that are considered as mapping candidates, e.g. by selecting mapping candidates from the top layer of WordNet. In practice, this reduction requires a systematic and well-motivated methodology for establishing how far up the WordNet hierarchy we need to go to select the best mapping candidates, and none of the approaches used in mapping WordNet to existing ontologies to date have developed such a methodology.

Developing an effective methodology for mapping WordNet to an ontology is the first step to make the ontology useful. The next step is to establish which WordNet word sense is appropriate for a given word token in context, in the event several choices are possible, so as to automate the assignment of ontological classes to words in target documents. Suppose for example we are working with an ontology comprising several possible event classes for the verb lemma *conduct*: #manage, #perform, #behave and #transmit. If the ontology has already been mapped to WordNet, then each of these four classes would be linked to a different WordNet sense for the lemma *conduct*

- #manage:{*conduct#v#1, …* }
  - direct the course of; manage or control; *You cannot conduct business like this*
- #perform:{*conduct #v#2, …* }
  - lead, as in the performance of a composition; *Barenboim conducted the Chicago symphony for years*
- #behave:{*conduct#v#3, …* }
  - behave in a certain manner; *They conducted themselves well during these difficult times*
- #transmit:{*conduct#v#4, …* }
  - transmit or serve as the medium for transmission; *Many metals conduct heat*.

Automated ontological annotation in this case could leverage WordNet-based word sense disambiguation algorithms to establish which of these four classes is appropriate for the lemma *conduct* in the context *support the right of Iran to **conduct** a nuclear program for peaceful purposes*. Unfortunately, word sense disambiguation is a difficult task to perform successfully. The best word sense disambiguation results in the "all word" task for the Senseval3 evaluation[6]

[3] See the WordNet OWL ontology developed by the KID group `http://taurus.unine.ch/knowler/wordnet.html`.

[4] `http://www.w3.org/2001/sw/BestPractices/WNET/tf.html`.

[5] `http://www.loa-cnr.it/DOLCE.html`.

[6] `http://www.senseval.org/`

are at 0.652 precision/recall (Snyder & Palmer, 2004), with Kohomban & Lee (2005) reporting 0.661 for the same task and data set. Such results are only marginally better than baseline heuristics such as choosing the most frequent word sense in WordNet (0.609), and are just not reliable enough for most practical applications.

3 Defining a WordNet Ontology

Our main objective in constructing a WordNet-based ontology is to select a manageable number of classes that have sufficient conceptual depth to enable effective semantic inference and enough variety to yield the widest lexical coverage. The work we have carried out to date is primarily concerned with verbs and nouns, but the approach developed extends to other word classes in WordNet (adjectives and adverbs) in a straightforward manner.

In defining an event ontology based on WordNet, we selected verb synonym sets that were less specific in meaning as event classes (e.g., {*communicate#2*, *intercommunicate#2*} vs. {*gesticulate#1*, *gesture#1*, *motion#1*}). In doing so, we chose the more frequent member of the synonym set to name the class, e.g. *communicate#2* for the synonym set {*communicate#2*, *intercommunicate#2*}. The verbs in the synonym sets chosen as event classes (e.g., *communicate#2*, *intercommunicate#2*) as well as their troponyms (e.g., {*gesticulate#1*, *gesture#1*, *motion#1*}, {*grimace#1*, *make_a_face#1*, *pull_a_face#1*}) were declared as instances. The ontology is being developed as an OWL ontology[7] using Protégé[8] as the ontology editor environment and Jena[9] as the Semantic Web framework in which to implement the ontology, handle reification, issue queries, and perform logical inference. An example of the resulting event ontology is shown in Figure 1, where verb senses associated with the folder icon indicate event classes while those associated with a bullet point are instances.

To assess the specificity level of synonym sets, we used frequency counts for WordNet synonym sets obtained from the British National Corpus (BNC) using the methodology established by Resnik (1995) as implemented by Pedersen, Banerjee and Patwardhan[10] (see also Pedersen et al. 2005, p. 15). Since the BNC is not annotated with WordNet word senses, concept counts were distributed across all possible senses of a word. Frequencies of the verb senses were computed by taking the count of a verb and splitting it among its senses and hypernyms; thus each sense and hypernym associated with a word type received an equal share of each count. For example, if there are two senses of a word, then each of the concepts associated with each sense is updated by 0.5 when we observe the word in a corpus.

Verb synonym sets that have hyponyms and whose frequency counts were above a given threshold were chosen as event classes. BNC frequency counts for verb synonym sets ranged from 0 to 2,060,415. We chose a frequency cut-off value of 10,000. The chosen synonym sets tended to be in the top- to mid-layer of the WordNet hierarchy and have a high number of hyponyms as they designated more general event concepts. Following this method, we created 1077 event classes out of a total of 24,632 verb senses. 386 top-level verb synonym sets had no hyponyms or were below the frequency cut-off value. 69 of these verb synonym sets were mapped to other verb synonym sets using the "similar sense" function in WordNet. The remaining 317 verb synonym sets represent rarer and more specific concepts, have very few or no hyponyms, and are therefore not well suited as ontology classes; however, we have included these in our event ontology for completeness. We are currently trying to find ways to integrate these 317 verb synonym sets in the event ontology as instances for some of the remaining 760 synonym sets that are more linguistically motivated as event classes.

Using the same approach with a BNC frequency threshold of 6100, we defined a noun ontology of 3005 nodes. The development of the noun ontology was comparatively simpler since WordNet only has 9 top-level synonym sets for nouns and they are all good candidates as ontology classes.

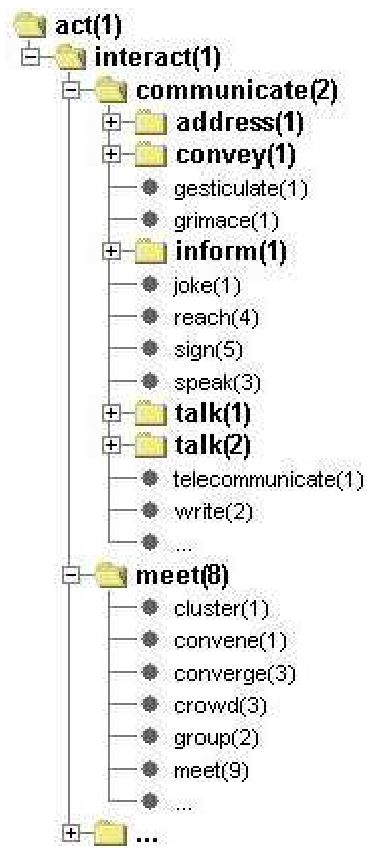

Figure 1: WordNet-based event ontology fragment.

[7] http://www.w3.org/TR/owl-ref.
[8] http://protege.stanford.edu.
[9] http://jena.sourceforge.net.
[10] http://search.cpan.org/dist/WordNetSimilarity/utils/BNCFreq.pl.

4 Automatic Word Class Recognition

Our main objective in targeting the disambiguation of word classes as opposed to individual word senses is to obtain results that significantly exceed current word sense disambiguation results. In our event ontology, nearly 25,000 verb senses mapped into 1077 verb classes. Such mapping significantly reduces the number of possible choices in assigning a concept to an ambiguous verb and should therefore simplify the disambiguation challenge. This hypothesis is supported by previously reported good performance for coarse grained word sense disambiguation systems (Yarowsky 1992).

Our approach is based on a supervised classification approach and we use SemCor[11] as training corpus. Currently, we employ the OpenNLP MaxEnt implementation[12] of the maximum entropy classification lgorithm (Berger et al. 1996) to develop word class recognition models. For each verb/noun lemma, we create a classifier that predicts which of the possible verb/noun classes for the lemma is most likely according to the context in which the lemma occurs.

Following Dang & Palmer (2005) and Kohomban & Lee (2005), we use contextual, syntactic and semantic information to inform our verb class disambiguation system.

- Contextual information includes the verb under analysis plus the three tokens found on each side of the verb, within sentence boundaries. Tokens included word as well as punctuation.
- Syntactic information includes grammatical dependencies (e.g. subject, object) and morpho-syntactic features such as part of speech, case, number and tense. We used the Connexor parser[13](Tapanainen and Järvinen, 1997) to extract syntactic information. A sample output of a Connexor parse is given in Table 1.
- Semantic information includes named entity types (e.g. person, location, organization) and hypernyms.
  - We used LCC's Cicero Lite[14] to extract named entity types, replacing the strings identified as named entities (e.g., Joe Smith) with the corresponding entity type (PERSON). We also substituted personal pronouns that unambiguously denote people with the entity type PERSON.
- Hypernyms were retrieved from WordNet. Differently from Dang & Palmer (2005), we only expanded the hypernym of sense 1 of lemmas, but we included the entire hypernym chain (e.g. motor, machine, device, instrumentality, artifact, object, whole, entity).

A sample of the resulting feature vectors which were used both for training and recognition is given in Table 2.

As the example in Table 2 indicates, combination of contextual, syntactic and semantic information types results in a large number of features. Inspection of the training data reveals that some features may be more important than others in establishing verb class assignment for each choice of verb lemma. We used a feature selection procedure to reduce the full set of features to the feature subset that is most relevant to verb class assignment for each verb lemma. This practice improved both the efficiency and effectiveness of our verb class disambiguation algorithm. The feature selection procedure we adopted consists in scoring each potential feature according to a particular feature selection metric, and then take the best k features. We choose the Information Gain selection metric, measuring the decrease in entropy when the feature is given vs. when it is absent. Yang and Pederson (1997) report that the Information Gain performed best in their multi-class benchmarks, and Foreman (2003) showed that it performed amongst the best for his 2-class problems. In the future we intend to improve the feature selection process by developing a better subset selection procedure based on Information Gain. The procedure will score subsets of features simultaneously rather than individual features, thereby identifying high value feature combinations.

To adapt this approach for noun disambiguation, we simply selected a slightly different set of features, as indicated below.

- The noun under analysis plus the three tokens found on each side of the noun within sentence boundaries, and all verbs within sentence boundaries. Tokens included words as well as punctuation.
- Morphological information about all tokens chosen (e.g. POS, Case, Number).
- The syntactic dependency of the noun, an indication of what dependents the noun has (e.g. "hasDet"), a specification of what verb is related to the noun, and the syntactic dependency of the words dependent on the noun (e.g. 'det:the').
- The hypernym chain of all nouns selected.

4.1 Evaluation

We evaluated our disambiguation algorithm in two distinct tasks: verb and noun class disambiguation and verb sense disambiguation. The first evaluation task demonstrates the utility of the disambiguation algorithm with specific reference to the ontological annotation challenge. The second task provides an evaluation of the disambiguation algorithm with reference to comparable results in the literature.

4.1.1 Verb and Noun Class Disambiguation

The goal of the class disambiguation task is to disambiguate a verb or a noun with reference to the classes in the ontology defined in section 2, rather than individual WordNet senses. The reason for collapsing verb and noun senses into verb and noun classes is to simplify the disambiguation task by modeling coarser-grained categories to better support ontological annotation. We used the SemCor corpus for this evaluation task. As described above, we created classifiers that predict for each verb or noun lemma which of the possible verb or noun classes is most likely for the lemma,

[11] http://www.cs.unt.edu/~rada/downloads.html.
[12] http://maxent.sourceforge.net/.
[13] http://www.connexor.com/.
[14] http://www.languagecomputer.com/solutions/information_extraction/cicero_lite.

Table 1: Connexor sample output for the sentence *"The engine throbbed into life"*.

ID#	**Word**	**Lemma**	**Grammatical Dependencies**	**Morphosyntactic Features**
1	the	the	det:>2	@DN> %>N DET
2	engine	engine	subj:>3	@SUBJ %NH N NOM SG
3	throbbed	throb	main:>0	@+FMAINV %VA V PAST
4	into	into	goa:>3	@ADVL %EH PREP
5	life	life	pcomp:>4	@<P %NH N NOM SG
6	.	.		

Table 2: Feature vector for the sentence *"The engine throbbed into life"*.

the	pre:2:the, pre:2:pos:DET, det:the, det:pos:DET, hassubj:det:
engine	pre:1:instrumentality, pre:1:object, pre:1:artifact, pre:1:device, pre:1:engine, pre:1:motor, pre:1:whole, pre:1:entity, pre:1:machine, pre:1:pos:N, pre:1:case:NOM, pre:1:num:SG,subj:instrumentality,subj:object, subj:artifact, subj:device, subj:engine, subj:motor, subj:whole, subj:entity, subj:machine, subj:pos:N, hassubj:, subj:case:NOM, subj:num:SG,
throbbed	haspre:1:,haspre:2:,haspost:1:, haspost:2:, haspost:3:, self:throb, self:pos:V, main:,throbbed, self:tense:PAST
into	post:1:into, post:1:pos:PREP, goa:into, goa:pos:PREP,
life	post:2:life, post:2:state, post:2:being, post:2:pos:N, post:2:case:NOM, post:2:num:SG, hasgoa:, pcomp:life, pcomp:state, pcomp:being, pcomp:pos:N, hasgoa:pcomp:, goa:pcomp:case:NOM, goa:pcomp:num:SG
.	post:3:.

according to the context in which the lemma occurs. Our baseline is given by selecting the verb/noun class linked to the sense for the lemma that has the lowest word sense number (e.g. the highest frequency).

We ran two experiments. In the first experiment, we ignored verb/noun classes with 4 or fewer instances. The selection of training and testing data sets from SemCor was done for each choice of verb/noun sense: 80% for training and 20% for testing. For example, if a verb had two senses, W1 and W2, with 10 SemCor examples for W1 and 3 for W2, we would ignore W2 and select from SemCor 8 samples for training and 2 for testing. In the second experiment, we included verb/noun classes with at least 3 instances and chose the same training to testing ratio, rounding up to X+1 when 20% of the testing data set was greater or equal to X.5, where X is an integer.

The results shown in Table 3 demonstrate that we do reasonably better than the baseline, especially in experiment 1 for verbs and experiment 2 for nouns. Moreover, 0.71–0.69 precision/recall for verbs seems to be a good result as verbs are known to be harder to disambiguate. For example, Snyder and Palmer (2004) report that verbs scored the lowest in inter-annotator agreement at 67.8% during the preparation of the evaluation data for Senseval3, followed by nouns at 74.9% and adjectives at 78.5%.

4.1.2 Verb Sense Disambiguation

Due to the uniqueness of the word class disambiguation task, comparable results are currently not available in the literature. In order to compare the performance of our system with that of other approaches, we trained our algorithm for word sense disambiguation and used the Senseval3 English All Words task test data. For training, we used verb instances in SemCor. If a verb occurring in the Senseval3 test data was not present in the SemCor training set, we assumed the most frequent sense. Since our system was built using WordNet 2.0 and Senseval3 uses WordNet 1.7.1, we mapped the output of our system to the corresponding WordNet 1.7.1 senses. Using the scoring software and results files available from senseval.org, we calculated the results for verbs relative to the two top performers in the Senseval3 English All Words task for comparison purposes: GAMBL (Decadt et al. 2004) and SenseLearner (Mihalcea 2004). The baseline was calculated by assuming the most frequent sense for each verb.

Table 3: Precision/recall results for verb and noun class disambiguation on SemCor data.

	Verbs	**Nouns**
Experiment 1 (5 SemCor examples or more)	0.711	0.837
Baseline for experiment 1	0.645	0.800
Experiment 2 (3 SemCor examples or more)	0.691	0.811
Baseline for experiment 2	0.630	0.773

Table 4 provides the precision scores for baseline and the three systems—see Snyder & Palmer (2004) for a description of the scoring system.

Table 4: Results for verb sense disambiguation on Senseval3 data.

System	Precision	Fraction of Recall
Our system	61%	22%
GAMBL	59.0%	21.3%
SenseLearner	56.1%	20.2%
Baseline	52.9%	19.1%

Overall, our disambiguation system yields better precision and recall scores. To verify the statistical significance of these results, we used a standard proportions comparison test (see Fleiss 1981, p. 30). According to this test, the precision of our system is significantly better than the baseline ($p = 0.000765$) and marginally better than SenseLearner ($p = 0.028$). The test does not detect a statistically significant difference between the scores reported by our system and GAMBL ($p = 0.21$).

5 Related Work

Considerable amount of effort has been devoted to the development of automatic annotation methodologies for the Semantic Web during the last few years. Most of the approaches proposed exploit information extraction techniques such as the recognition of named entities, relationships and events. For example, Kogut & Holmes (2001) present a system that generates DAML annotations for most proper nouns and common relationships from web pages using AeroText^{TM}, a commercial information extraction tool. Dingli et al. (2003) and Ciravegna & Wilks (2003) propose an adaptive information extraction approach where information from structured sources is used to train learning algorithms capable of automating the annotation of domain specific web pages. These approaches work well for the semantic annotation of named entities and for specific application domains where the vocabulary is somewhat limited and lexical ambiguity is a relatively low concern. With more generic content (e.g. newswires), semantic annotation requires additional tools and resources capable of providing large lexical coverage and a more fine grained identification of word meaning. For example, Witbrock et al. (2004) describe a system which uses a lexicon of about 24,620 lexemes (nouns, verbs and adjectives) and 5,429 semantic translation patterns to produce initial Cyc OWL annotations of arbitrary text documents automatically. The need to engage large scale semantic knowledge resources such as WordNet and word sense disambiguation algorithms capable of discriminating among contextually appropriate word meanings with reference to such resources is also discussed in Buitelaar and Declerck (2004).

6 Building an Ontological Annotation Environment for Intelligence Analysis

We are currently using the word class disambiguation algorithm described in this paper to develop an Ontological Annotation Tool (OAT) capable of supporting the extraction of evidence from document sets for intelligence analysis applications such as the analysis of competing hypotheses (Sanfilippo et al. 2005). As shown in Figure 2, OAT represents extracted evidence in the form of semantic graphs. These semantic graphs are the combined result of an event extraction process based on dependency parsing with Connexor and the word class disambiguation algorithm described in this paper. We use OWL (Web Ontology Language) to represent semantic graphs. OWL facilitates the description of data classes in a way that supports automated reasoning about the class membership of given instances. OWL class descriptions can specify subsumption relationships and the properties associated with members of a given class. Descriptions can also restrict class membership by property values.

The results of parsing are semantically interpreted by the verb and entity classes (see Figure 3). The verb class comprises 1077 verb classes defined in terms of the upper-level verb synonym sets selected from WordNet and their subsumption relations, as described above in Section 2. Each verb instance is tied to one or more instances of the entity class: these instances correspond to event participants. Verb and entity instances have additional information that ties them to associated text within source documents. Our entity class is currently based on the entity types supported by the Cicero Lite named entity recognition system.

After the initial text parsing is completed, verb disambiguation is performed to determine the correct verb classification for events. This is recorded in the knowledge base by assigning the verb class to the event instance. An example of the OWL output produced is shown in Table 5.

OAT uses the Jena Ontology API to create models that describe the results of document parsing and disambiguation. These models are viewable by the user. The granularity of events displayed can be controlled by moving up and down the event hierarchy and by the types of restrictions placed on entity instances tied to the events. The use of OWL and the Jena API will allow us to support user-defined restrictions on the participants of events which are considered intelligence targets.

7 Conclusions

If ontologies are to support Semantic Web applications, a reliable system to relate words in naturally occurring text to ontological classes must be made available. In this paper, we have shown that such a system can be developed by defining a WordNet-based ontology that offers a manageable set of concept classes, provides an extensive characterization of concept class in terms of lexical instances, and integrates an automated class recognition algorithm. Our current verb class disambiguation algorithm demonstrates strong performance, and better results yet are expected for noun and adjective classes. Once completed, our WordNet-based ontology can be used as such or mapped to other ontologies to provide ontological annotation functionality. Because of the substantial reduction of WordNet synonym sets considered as mapping candidates, our approach can also reduce the costs and improve the results in the alignment of WordNet with existing ontologies. The ensuing framework

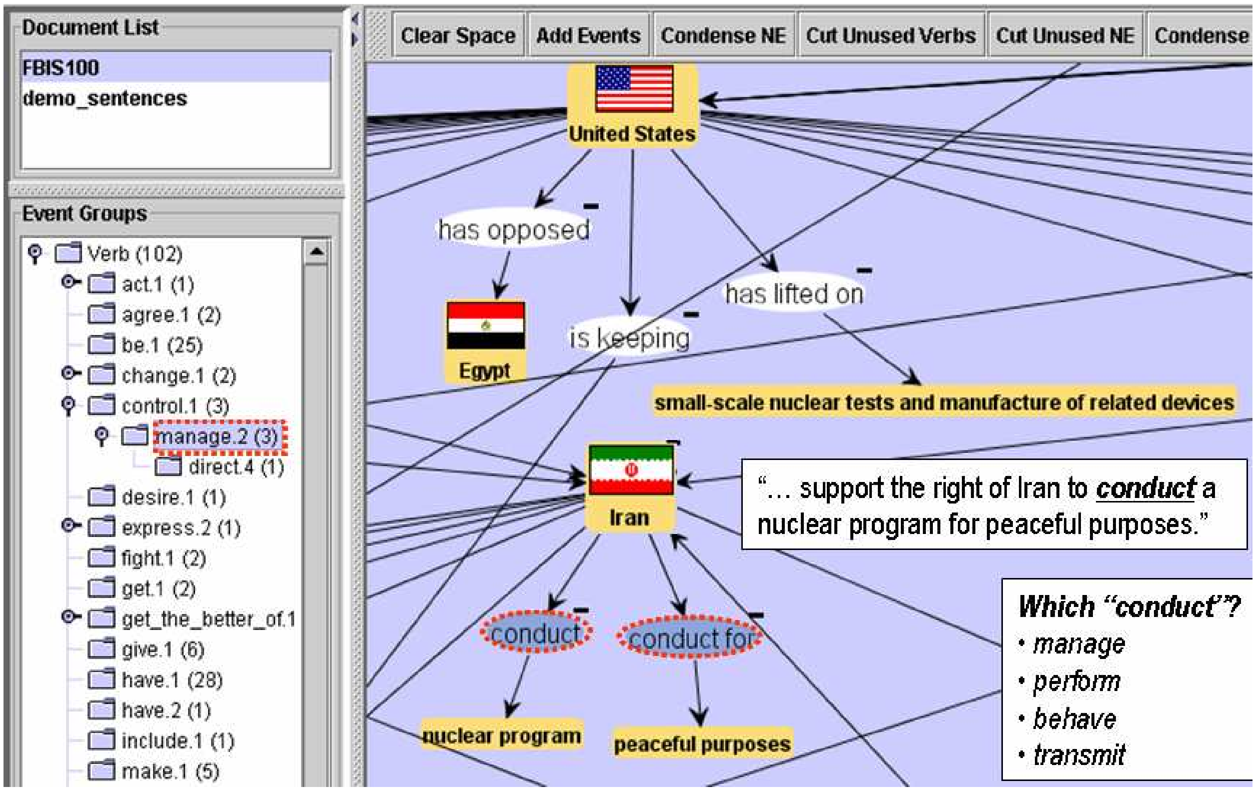

Figure 2: OAT sample.

makes available an ontological annotation platform that can be effectively integrated with intelligence analysis systems to facilitate evidence marshaling and sustain the creation and validation of inference models.

Acknowledgments

The work described in this paper was developed within the context of ADVISE (*Analysis, Dissemination, Visualization, Insight, and Semantic Enhancement*)—one of several research thrusts in the *Threat Assessment Portfolio* at the Department of Homeland Security (DHS), Science and Technology Directorate. We would like to thank Elsa Augustenborg and Rich Quadrel for continuous feedback and support.

References

Broekstra, J., A. Kampman, F. van Harmelen (2003) Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. In Fensel, D., J. Hendler, H. Lieberman, and W. Wahlster (eds.) *Spinning the Semantic Web*. MIT Press, Cambridge, MA, 2003, pp. 197–222.

Berger, A., S. Della Pietra and V. Della Pietra (1996) A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, volume 22, number 1, pages 39–71.

Buitelaar, P. and T. Declerck (2003) Linguistic Annotation for the Semantic Web. In S. Handschuh, S. Staab (eds.) *Annotation for the Semantic Web*, IOS Press, Amsterdam, The Netherlands.

Ciravegna, F. and Y. Wilks (2003) Designing Adaptive Information Extraction for the Semantic Web in Amilcare. In S. Handschuh, S. Staab (eds.) *Annotation for the Semantic Web*, IOS Press, Amsterdam, The Netherlands.

Dang, H. T. and M. Palmer (2005) The Role of Semantic Roles in Disambiguating Verb Senses. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor MI, June 26–28, 2005.

Decadt, B., V. Hoste, W. Daelemans and A. Van den Bosch (2004) GAMBL, genetic algorithm optimization of memory-based WSD. *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain.

Dingli, A., F. Ciravegna, and Y. Wilks (2003) Automatic Semantic Annotation using Unsupervised Information Extraction and Integration. In *Proceedings of the K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation*, Sanibel, Florida, October 26, 2003.

Fensel, D., J. Hendler, H. Lieberman, and W. Wahlster (eds.) *Spinning the Semantic Web*. MIT Press, Cambridge, MA, 2003.

Fleiss, J. L. *Statistical Methods for Rates and Proportions*. 2nd ed. New York: John Wiley & Sons; 1981.
Foreman, G. (2003) An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3, pages 1289–1305.

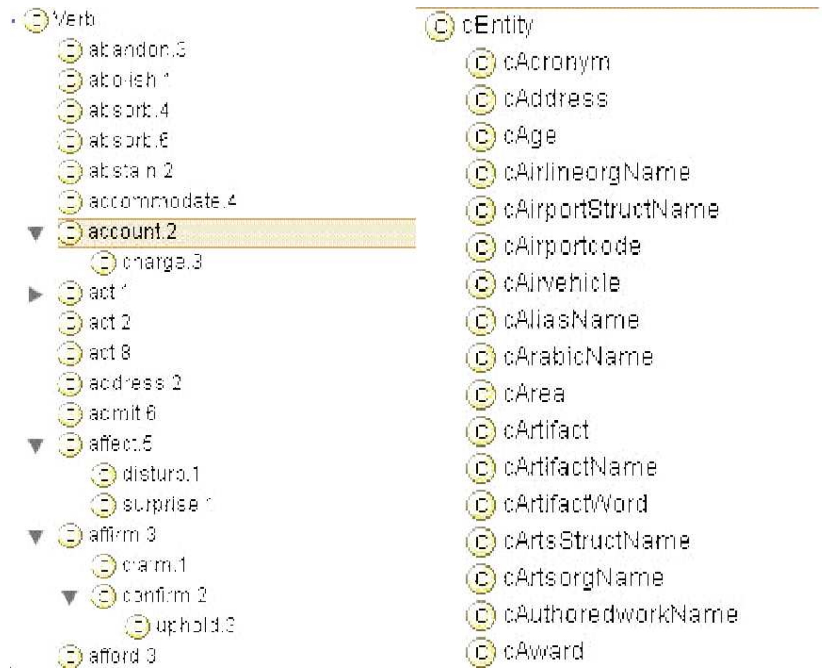

Figure 3: Verb and Entity classes in OAT.

Table 5: Class disambiguation OWL output for the verb convene in the context *In May 1992, at Qaddafi's instigation, 1,500 People's Congresses convened in Libya and abroad...* The verb convene is assigned WordNet sense 1 and thus recognized as an instance of the verb class `meet.8` (see Figure 1) which corresponds to the WordNet synonym set comprising the verb senses `meet#8`, `gather#2`, `assemble#2`, `forgather#1` and `foregather#1`.

```
<txtmark:cEvent rdf:about="http://nvac
.pnl.gov/sid/owl/data/libya_government
.htm#convene_2913">

<txtmark:dStartIndex rdf:datatype=
"http://www.w3.org/2001/XMLSchema#int">
1169
</txtmark:dStartIndex>

<verbs:pLemma>convene</verbs:pLemma>

<verbs:pWordNetSense rdf:datatype=
"http://www.w3.org/2001/XMLSchema#int">
1</verbs:pWordNetSense>

<verbs:pText rdf:datatype="http://
www.w3.org/2001/XMLSchema#string">
convened in</verbs:pText>

<rdf:type} rdf:resource="http://
nvac.pnl.gov/sid/owl/verbs\#meet.8"/>
...
</txtmark:cEvent>
```


Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. Sweetening WordNet with DOLCE. *AI Magazine* 24(3): Fall 2003, pp. 13–24.

Hovy, E. (1998). Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. Granada, Spain.

Klein, M., J. Broekstra, D. Fensel, F. van Harmelen, and I. Horrocks (2003). *Spinning the Semantic Web, chapter 4: Ontologies and schema languages on the Web*, pages 95-139. The MIT Press, Cambridge, Massachusetts.

Knight, K. and S. K. Luk. 1994. Building a Large-Scale Knowledge Base for Machine Translation. *Proceedings of the AAAI Conference*.

Kogut, P., Holmes, W. (2001) AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In *K-CAP 2001 Workshop KnowledgeMarkup & Semantic Annotation*.

Kohomban, U. and W. Lee (2005) Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual meeting of the Association for Computational Linguistics*, Ann Arbor, MI.

Maedche, A., S. Staab, N. Stojanovic, R. Studer, and Y. Sure. SEmantic PortAL: The SEAL approach. In Fensel, D., J. Hendler, H. Lieberman, and W. Wahlster (eds.) *Spinning the Semantic Web.* MIT Press, Cambridge, MA, 2003, pp. 317–354.

Mihalcea, R. and E. Faruque (2004) SenseLearner: Minimally supervised word sense disambiguation for all words in open text. *SENSEVAL-3: Third International Wortk-*

shop on the Evaluation of Systems for the Semantic Analysis of Text. Barcelona, Spain.

Niles, I. and Pease, A. (2003) Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE '03)*, Las Vegas, Nevada, June 23–26, 2003.

O'Hara, T., N. Salay, M. Witbrock, et al. (2003) Inducing criteria for mass noun lexical mappings using the Cyc Knowledge Base and its Extension to WordNet. In *Proceedings of the Fifth International Workshop on Computational Semantics*, Tilburg, 2003.

Pedersen, T., S. Banerjee and S. Patwardhan (2005) Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. University of Minnesota Supercomputing Institute Research Report UMSI 2005/25, March. Available at `http://www.msi.umn.edu/general/Reports/rptfiles/2005-25.pdf`.

Resnik, P. (1995) Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Vol. 1, 448–453, Montreal, August 1995.

Snyder, B. and M. Palmer. 2004. The English all-words task. *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain.

Sanfilippo, A., B. Baddeley, A. J. Cowell, M. L. Gregory, R. Hohimer and S. Tratz (2005) Building a Human Information Discourse Interface to Uncover Scenario Content. In *Proceeding to the 2005 International Conference on Intelligence Analysis*. 2–6 May, 2005 McLean, VA.

Tapanainen, P. and Timo Järvinen (1997) A nonprojective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71, Washington D.C. Association for Computational Linguistics.

Witbrock, M., K. Panton, S.L. Reed, D. Schneider, B. Aldag, M. Reimers and S. Bertolo (2004) 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*, Hiroshima, Japan, November 2004, pp. 71–80.

Yang, Y. and J. O. Pedersen (1997) A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning* (ICML), pages 412–420, 1997.

Yarowsky, D. (1992) Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of COLING-92*, pages 454–460.