Evaluating the Performance of the OntoSem Semantic Analyzer

Sergei NIRENBURG, Stephen BEALE and Marjorie MCSHANE
Institute for Language and Information Technologies (ILIT)
University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250 USA
sergei@umbc.edu, sbeale@umbc.edu, marge@umbc.edu

Abstract

This paper describes an innovative evaluation regimen developed for the text meaning representations (TMRs) produced by the Ontological Semantic (OntoSem) general purpose syntactic-semantic analyzer. The goal of evaluation is not only to determine the quality of TMRs for given texts, but also to assign blame for various classes of errors, thus suggesting directions for continued work on both knowledge resources and processors. The paper includes descriptions of the OntoSem processing environment, the evaluation regime itself and results from our first evaluation effort.

1   Introduction

In this paper we describe the evaluation regimen for a general-purpose syntactic-semantic analyzer, OntoSem, under continuous development at the Institute for Language and Information Technologies (ILIT) of the University of Maryland Baltimore County. Its top-level architecture is illustrated in Figure 1. The knowledge in the fact repository and the ontology serves not only OntoSem itself but also provides a knowledge substrate to be used in a variety of reasoning applications. At present, the acquisition of the ontology and the semantic lexicon is carried out by human acquirers using interactive tools. The acquisition of the fact repository is mixed, with some of it carried out manually and some of it resulting from the operation of the fact extractor on the results of semantic analysis.

The approach to semantic analysis in OntoSem is described in some detail in, e.g., Nirenburg and Raskin 2004, Nirenburg et al. 2003, Beale et al 2003. Our description here will be necessarily brief.

Text analysis in OntoSem relies on the results of a battery of pre-semantic text processing modules. The preprocessor module deals with mark-up in the input text, finds boundaries of sentences and words, and recognizes dates, numbers, named entities and acronyms. Morphological analysis accepts a string of word forms as input and for each word form outputs a record containing its citation form in the lexicon and a set of morphological features and their values that correspond to the word form from the text. Once the
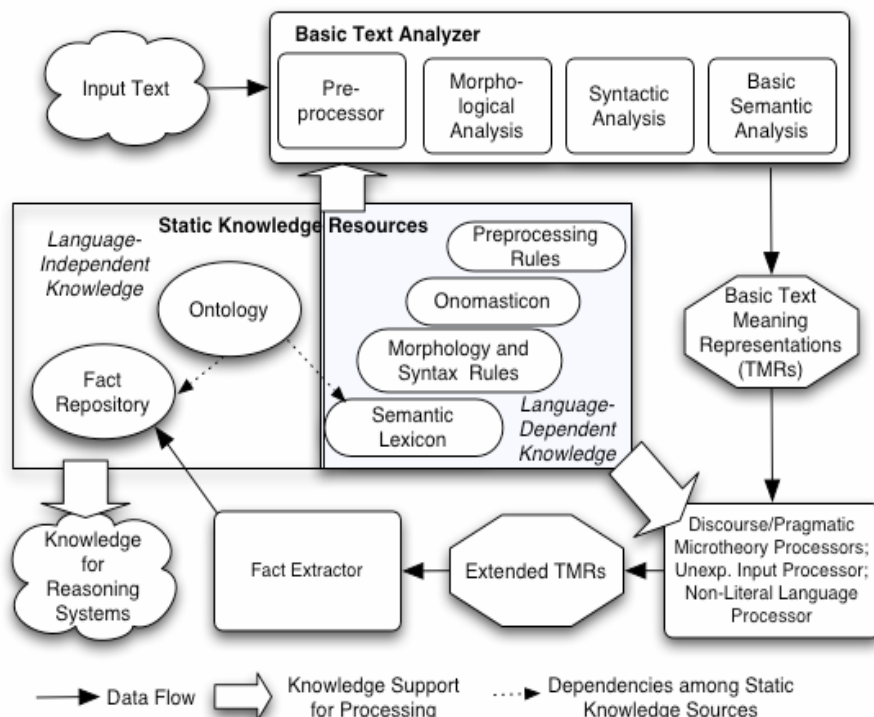

Figure 1. The overall architecture of the OntoSem semantic analyzer. The evaluation regimen described in this paper evaluates the production of basic TMRs. Other processing will be evaluated in follow-up work.

morphological analyzer has generated the citation forms for word forms in a text, the system can

activate the relevant lexical entries in its lexicons, including the onomasticon (a lexicon of proper names). The task of syntactic analysis in ontological semantics is, essentially, to determine clause-level dependency structures for an input text and assign grammatical categories to clause constituents (that is, establish subjects, direct objects, obliques and adjuncts).

Semantic analysis proper uses the information (mutual constraints) in the active lexicon entries, the ontology and the results of earlier processing to carry out, at the first stage, word sense disambiguation and establish basic semantic dependencies in the text. The results are recorded
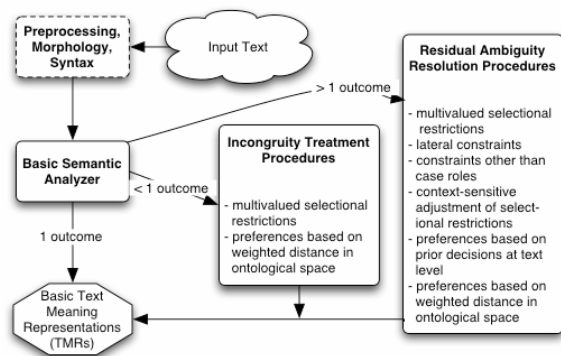

Figure 2. Creation of basic TMRs. The basic semantic analyzer relies largely on matchin selectional restrictions. This can lead to incongruity when the listed constraints are too strong, or to residual ambiguity if they are too weak to filter out all but one candidate. To resolve these problems, OntoSem uses both static knowledge (multivalued selectional restrictions and lateral constraints among co-arguments of a predicate) and context-generated heuristics, including information in the nascent TMR and measuring distances among any two concepts in the ontological search space.

in *basic* text meaning representations (TMRs).

At the next stage, the analyzer determines the values of the various modalities, aspect, time, speech acts, speaker attitudes etc., to produce *extended* TMRs. At both stages, the analyzer has to deal with ambiguity, incongruity between the input and expectations recorded in the static knowledge sources, unknown words, and non-literal language. Figure 2 summarizes the types of heuristics that the analyzer uses at the first stage. While all of procedures using them have been implemented (see Nirenburg et al. 2003), the version of the analyzer we evaluated involved only a subset of them. We plan to evaluate the analyzer with all the available recovery procedures in the near future.

The **OntoSem ontology** provides a metalanguage for describing the meaning of the lexical units in a language as well as for the specification of meaning encoded in TMRs. The ontology contains specifications of concepts corresponding to classes of things and events in the world. It is a collection of frames, or named collections of property-value pairs, organized into a hierarchy with multiple inheritance. The expressive power of the ontology and the TMR is enhanced by multivalued fillers for properties, implemented using the value "facets" DEFAULT, SEM, VALUE, and RELAXABLE-TO, among others. At the time of this writing, the ontology contains about 5,500 concepts (events, objects and properties), with, on average, 16 properties each.

The **OntoSem lexicon** contains not only semantic information, it also supports morphological and syntactic analysis. Semantically, it specifies what concept, concepts, property or properties of concepts defined in the ontology must be instantiated in the TMR to account for the meaning of a given lexical unit of input. At the time of writing, the latest version of the English semantic lexicon includes over 12,000 handcrafted entries. These entries cover some of the most complex lexical material in the language – "closed-class" grammatical lexemes such as conjunctions, prepositions, pronouns, auxiliary and modal verbs, etc., as well as about 3,000 of the most frequent verbs. We illustrate the structure of the lexicon entry on the example of the first verbal sense of *alert* (presented in a simplified format):

```
alert-v1
  example  "He  alerted  us  to  the
           danger"
  morph     regular
 syn-struc
  root     root $var0 cat v
  subject  root $var1 cat n
  object   root $var4 cat n opt +
  pp-adjunct
         root $var2
           cat prep root to opt +
           object
              root $var3 cat n
 sem-struc
  WARN       ;an ontological concept
   agent    value ^$var1 sem HUMAN
   theme    value ^$var3
   beneficiary  value ^$var4
   instrument   value ^$var1
              sem (or
                 ARTIFACT EVENT)
  ^$var2   null-sem +
```

In the lexicon, variables (e.g., $var2) support syntax-semantics dependency linking; the caret "^"

is read "the meaning of." In this example, if ^$var1 is HUMAN or a descendant of HUMAN, it occupies the semantic role of AGENT (*he alerted us...*), whereas if it is ARTIFACT or EVENT (or a descendant of any of those concepts) it is INSTRUMENT (*the bell alerted us..., his behavior alerted us...*). For lack of space, we will not be able to discuss all the representational and descriptive devices used in the lexicon or the variety of the ways in which semantic information in the lexicon and the ontology can interact. See Nirenburg and Raskin 2004 for discussion.

The English **Onomasticon** (lexicon of proper names) currently contains over 350,000 entries that are semantically linked to ontological concepts by way of the fact repository. Onomasticon entries are indexed by name (e.g., *New York*), while the entries in the fact repository are identified by appending a unique number to the name of the ontological concept of which they are instances (e.g., *Detroit* might be listed as CITY-213).

The **TMR** (automatically generated but shown here in a simplified presentation format) for the short sentence *He asked the UN to authorize the war* from a recently processed text about Colin Powell is presented below. The numbers associated with the ontological concepts indicate instances of those concepts: e.g., REQUEST-ACTION-69 means the 69th time that the concept REQUEST-ACTION has been instantiated in the world model used for, and extended during, the processing of this text or corpus.

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 ; *ref. resolution done*
WAR-73
  THEME-OF   ACCEPT-70
  SOURCE-ROOT-WORD war

The above says that there is a REQUEST-ACTION event whose AGENT is HUMAN-72 (Colin Powell),

whose BENEFICIARY is ORGANIZATION-71 (United Nations) and whose THEME is ACCEPT. The ACCEPT event, in turn, has a THEME of WAR-73. Note that the concept ACCEPT is *not* the same as the English word *accept*: its human-oriented definition in the ontology is "To agree to carry out an action, fulfill a request, etc", which fits well here.

The **Fact Repository** contains a list of remembered instances of ontological concepts. As it does not play a significant role in the evaluation regimen reported in this paper, we will provide no further description here.

2 Generating Gold Standard TMRs

We have developed a human-aided version of the OntoSem analyzer in which the results of all three major stages of ontological semantic analysis – preprocessor output, syntax output and semantic output – can be inspected and corrected by a human. For purposes of evaluation, we have used it to produce gold standard (GS) outputs for each of the three stages. The production of gold standard outputs proceeds as follows:

1. Run the OntoSem analyzer on an input text.
2. Correct preprocessor output by hand in a text file. Preprocessor output is relatively simple to read in text format, and we have found it quickest to simply correct it by hand. It takes on average 1 minute to correct an average-length (> 25 words) sentence.
3. Input the corrected preprocessor results into the analyzer and produce a syntactic analysis.
4. If necessary, use a specially developed visual editing interface to add or delete edges on the chart that presents the results of syntactic analysis, to remove spurious parses, to correct phrase and clause boundaries, and to add any missing phrase or clause parses.
5. Feed the correct syntax back into the analyzer and obtain a semantic analysis.
6. If necessary, correct the semantic analysis.

We plan to integrate this capability with our knowledge acquisition interfaces in order to produce a full-function text processing system. The side effects of this process will include the creation of a bank of gold standard TMRs as well as, possibly, less importantly, gold standard results of preprocessing and syntactic analysis. Such resources are clearly valuable as training data for statistical NLP, and a number of projects are devoted to entirely or in a large measure to their creation. The process of producing gold standard TMRs, unlike most of the resource acquisition

approaches, is, to a significant degree, automated – which reduces the incidence of interannotator disagreement and generally makes the process faster and cheaper.

In the OntoSem research paradigm, knowledge acquisition (enhancement of the ontology, the lexicon and other basic static knowledge sources) is an ongoing process. The process of creating gold standard TMRs provides an empirical impetus for knowledge acquisition. This process will not at all interfere with our evaluation regimen because our approach does not rely on having a standard test corpus. We will simply run the entire evaluation procedure (starting with the production of the gold standard TMRs) on a new corpus, analyze the results and move on to yet another corpus, and so on.

This approach cannot be directly exported to those systems that do not involve knowledge acquisition of the kind OntoSem uses. However, the set of gold standard TMRs produced through our evaluation process will be made freely available and can serve as the test corpus for any other semantic analyzer (word sense disambiguator and/or semantic dependency extractor). This will be our direct contribution to the resource set in the field. Of course, using this resource will involve resolving the differences in the notation and semantics between the TMR structures and any other metalanguage.

This methodology for producing gold standard semantic outputs is the only one we consider practical. It is not possible for a human to produce gold standard semantic outputs by hand because of the complexity of the knowledge, as well as the high probability of annotator disagreement due to valid semantic paraphrasing (e.g., one annotator might describe the meaning of *weapons of mass destruction* as the union of BIOLOGICAL-WEAPON and CHEMICAL-WEAPON, whereas another might describe it as WEAPON that has the potential to kill more than 10,000 people).

In sum, gold standard outputs are used for a number of purposes: to evaluate the results of fully automatic analysis (see below); as training data for machine learning, with the goal of improving the system's static knowledge sources; to trigger manual acquisition of knowledge for lacunae; or to derive high-confidence TMRs for use in mining information for a fact repository. Last but not least, the gold standard TMRs produced according to our methodology can also be directly used in a variety of applications – from human-assisted knowledge-based MT to knowledge acquisition for general-purpose reasoning systems.

3 Automated Evaluation of Ontological Semantic Analyses

Once the gold standard TMRs are produced, the evaluation of OntoSem proceeds fully automatically. For each input we evaluate several "runs" as follows:

- **as is**: we simply input the text and evaluate the outputs;
- **baseline 1**: same as above, except we force the analyzer to use the first lexical sense of each word; the first senses in our lexicon entries are typically the most central and frequent ones;
- **baseline 2**: same as baseline 1, except we use the first sense that has the correct part of speech (as specified in the gold standard preprocessor results);
- **correct preprocessor output**: we use the gold standard preprocessor output as input to the syntactic and semantic analyzer;
- **corrected syntax output**: we use the gold standard syntax (and gold standard preprocessor output) as the input to the semantic analyzer.

For each run, we produce four output files: preprocessor results; syntax results; semantics results and evaluation results The evaluation is performed by automatically comparing the actual preprocessor, syntax or semantic results to the corresponding gold standard outputs. The evaluation produces statistics and/or measurements as follows.

General text-level statistics are collected from the golden standard outputs and include a) the word/phrase count; b) the number of input words that are not in the OntoSem lexicon; c) the syntactic ambiguity count, which is the number of phrases and clauses in the syntactic output; d) the semantic ambiguity count, which is the product of the number of senses of each word, which provides an estimate of the overall theoretical complexity of semantic analysis; and e) the word sense ambiguity count, which is the number of semantic combinations the analyzer actually needed to examine to produce the result; this number provides an estimate for the actual complexity of semantic analysis: syntactic clues often help prune many spurious analyses and the efficient semantic analysis algorithm (Beale, et. al. 1995) reduces the total number of combinations that have to be examined while maintaining accuracy.

For this evaluation, the lexicon provided almost complete lexical coverage of the input texts (in fact only one word was missing). We will use the results of this first evaluation as a baseline for future evaluation of the degradation of the results due to incompleteness of the static knowledge.

Results from the operation of the preprocessor, syntactic analysis and semantic analysis are collected for each evaluation run.

The **preprocessor statistics** are recorded as follows (m is the number of matches between an actual run and the gold standard, n is the number of mismatches): a) abbreviations, time, date and number recognition (m/n); b) named entity recognition (m/n); c) part of speech tagging (m/n). The overall score of the preprocessor is calculated as the average of m/m+n for all three measures.

Syntactic analysis statistics measure the quality of the determination of phrase boundaries, heads of phrases, and phrase attachment.

a) For phrase boundaries, an overall score between 0.0 and 1.0 is returned for each phrase, with 1.0 reflecting a perfect match. Each phrase in the gold standard syntax output is compared to its closest match in the output under consideration. The output phrase that has the same label (NP, CL, etc.), the same head word, and the closest matching starting and ending points is used for the comparison. Each phrase is given the score:

$$1 - (|gstart - start| + |gend - end|)/(gend - gstart)$$

where *gstart* is the gold standard word number at the start of the phrase and *start* is the word number at the start of the phrase being evaluated. Thus, if the gold standard phrase began at word 10 and ended at word 16, and the closest matching phrase in the output being evaluated began at word 9 and ended at word 17, then the score for this phrase would be $1 - (|10 - 9| + |16 - 17|) / (16 - 10) = (1 - (2 / 6)) = 2/3$. If no matching phrase could be found (i.e. no overlapping phrase could be found with the same phrase label and head word), then a score of 0.0 is assigned. The score for the whole sentence under evaluation is the average of the scores for each of the phrases.

b) For phrase head determination, the standard (m/n) measure is used. For each phrase in the gold standard syntax, it is determined if there exists a phrase with the same part of speech and head word that overlaps with the gold standard phrase.

c) Attachment is also measured as (m/n). For each phrase in the gold standard syntax, the evaluation procedure looks for a phrase that overlaps with it that has the same part of speech, the same head word and the same constituents. For example, if the gold standard output has a PP attached to a NP, it will be shown to be a constituent of that NP. If the output being evaluated attaches the PP at a different constituent, then a mismatch will be identified.

A score between 0.0 and 1.0 is assigned for b and c as follows: $Score = m/(m+n)$. The Syntactic Analysis Overall Score is then the average score of a, b and c.

Semantic analysis statistics measure the quality of word sense disambiguation (WSD) and semantic dependency (SD) determination. For WSD, three measures are computed.

A) First, the standard match/mismatch (m/n) is used. Each TMR element in the gold standard semantic representation is marked with the word number from the input text from which it arose. The TMR element in the semantic representation being evaluated that corresponds to that same word number is then compared with it.

B) Second, the evaluation system produces a weighted score for WSD complexity. An overall score between 0.0 and 1.0 is returned. A mismatch of a word with more senses is penalized less than a mismatch of a word with fewer senses. The score for each mismatch is $1 - (2 / number\text{-}of\text{-}senses)$, if the word has more than 2 senses, and 0.0 if it has less than or equal to 2 senses. An exact match is given a score of 1.0. The overall score for the sentence is the average score for each TMR element.

C) The system also computes a weighted score for WSD "distance." An overall score between 0.0 and 1.0 is returned. A mismatch that is ontologically "close" to the correct sense is penalized less than a mismatch that is ontologically "far" from the correct semantics. The ontological distance is computed using the Ontosearch algorithm (Onyshkevych 1997) that returns a score between 0.0 and 1.0 reflecting how close the two concepts are in the ontology, with a score of 1.0 indicating a

perfect match. The overall score for the sentence is the average score of each TMR element.

D) The quality of semantic dependency determination is computed using the standard (m/n) measure. Each TMR element in the gold standard is compared to the corresponding TMR element in the semantics being evaluated. Each property modifying the gold standard TMR element that is also in the evaluation TMR element increments the m count, each property in the gold standard TMR element that is not in the evaluation TMR element increments the n count. The fillers of matching properties are also compared. If the filler of the gold standard property is another TMR element (as opposed to being a literal), then the filler is also matched against the corresponding filler in the semantic representation being evaluated, incrementing the m and n counters as appropriate. The relations between TMR elements is one of the central aspects of Ontological Semantics which goes beyond simple word sense disambiguation. This score reflects how well the dependency determination was performed.
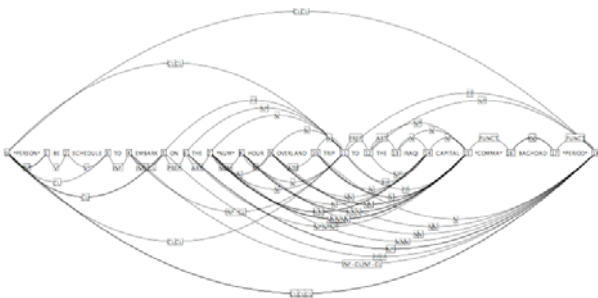

Figure 3: Syntactic Analysis of Sample Text

A normalized score between 0.0 and 1.0 is calculated for a and d as follows: Score = m/(m+n).

Example Semantic Evaluation

We will now exemplify the evaluation of the semantic analysis of the sample sentence in 1:

1. *Hall is scheduled to embark on the 12 hour overland trip to the Iraqi capital, Baghdad.*

The analyzer produces the syntactic analysis shown in Figure 3. This analysis contains many spurious parses (along with the correct ones). The gold standard parse of this sentence is shown in Figure 4. The illustrations are difficult to read but the number of edges can be visually compared.

In order to make an interesting evaluation example, we forced the semantic analyzer to misinterpret *capital*. The analyzer actually chose the correct sense, CAPITAL-CITY, but here we will force it to select the monetary sense, CAPITAL.

We will now demonstrate the calculation and significance of the semantic evaluation parameters.

A) Match/mismatch of TMR elements. In this example, there will be six matches and one mismatch – the CAPITAL concept that should be CAPITAL-CITY. A score of $6/7 = 0.86$ is also calculated for use in the overall semantic score.

B) Weighted score for WSD complexity. The word *capital* has three senses in our English lexicon, corresponding to the CAPITAL-CITY, CAPITAL (i.e. monetary) and CAPITAL-EQUIPMENT meanings. It will receive a score of $1 - 2/\text{number-of-senses} = 1 - 2/3 = 0.33$. If there were two or less senses, it would have received a score of 0.0. If there were many senses of capital, its score would have been higher, reflecting the fact that there was a more complex disambiguation problem. The other six TMR elements receive a score of 1.0. The total score for the sentence is therefore $6.33/7 = 0.90$.
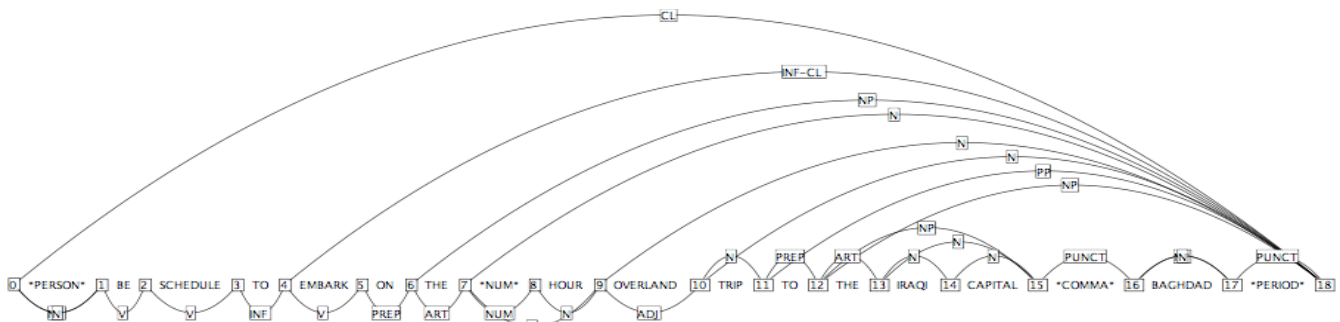

Figure 4. Gold Standard Syntactic Analysis for a sample sentence.

C) Weighted score for WSD distance. We determine the distance between the chosen meaning, CAPITAL, and the correct meaning, CAPITAL-CITY, by submitting the concept pair to Ontosearch:

(ontosearch capital capital-city) $\rightarrow$ 0.525
PATH:
CAPITAL **IS-A** FINANCIAL-OBJECT
FINANCIAL-OBJECT **SUBCLASSES** DEED
DEED **IS-A** DOCUMENT
DOCUMENT **PRODUCED-BY** NATION
NATION **LOCATION-OF** CITY
CITY **SUBCLASSES** CAPITAL-CITY

Ontosearch returns a score between 0.0 and 1.0 reflecting the closeness of the two concepts. An exact match would return a score of 1.0. Ontosearch also returns the path traversed to link the two concepts. In this case, the score returned is relatively low, and the "strange" path needed to connect the two concepts reflects this. So the score for this TMR element is 0.52. The other TMR elements in the sentence all receive a score of 1.0, so the score for the sentences is 6.52/7 = 0.93.

D. Semantic dependency determination. In the example input, there are six links between TMR elements. Thus, the instance of SCHEDULE-EVENT has as its THEME the instance of TRAVEL-EVENT, which has an instance of CAPITAL as its DESTINATION, an instance of HUMAN as its AGENT and an instance of HOUR as its DURATION. CAPITAL is linked to NATION and CITY. Each link is checked against the gold standard. In this case, all six links match. This increments the dependecy match counter by six. The fillers of the link, i.e. the TMR element that it points to, are also checked. For this example, the DESTINATION of the TRAVEL- EVENT should be CAPITAL-CITY, but it is CAPITAL. This increments the mismatch counter by one. The other five fillers match with the gold standard, thus the match counter is incremented by 5. For the whole sentence, the dependency matches will be 11 and the mismatches will be 1. In this case, the mismatched dependency was caused by the misanalysis of *capital*. In other cases, mismatched dependencies can arise by incorrect linking between syntactic and semantic structures. A score of 11/12 = 0.92 is calculated for use in the overall score.

4   Results of the First Evaluation Run

Our first evaluation run returned the results summarized in Tables 1 and 2. The motivation for the different statistics and runs was given in Section 3.

5   Discussion and Future Work

The kind of evaluation that we have undertaken so far reflects our desire to understand the causes of less-than-maximum results, that is, to assign blame to the various components of the analyzer. The results clearly show that the preprocessor we have so far been using in the OntoSem system does not perform sufficiently well, and we will change the preprocessor for the future evaluations, for instance, by using the corresponding components of the Stanford Lexicalized Parser (accessible from http://nlp.stanford.edu/).

word count	204
sense count	604
syntactic ambiguity	192
semantic ambiguity	1.9×10^{17}
word sense ambiguity	4.8×10^{8}

Table 1. The general statistics for the first evaluation run of OntoSem

Our WSD evaluation environment differs from many WSD approaches in that it allows the "none of the above" outcome for the cases when the lexicon entries do not fit the expectations in the text even after a measure of constraint relaxation. The count of incorrectly determined word senses includes the above eventuality but also the case when the current system has to select an answer from a set of candidates none of which can be preferred on the basis of available heuristics. For future evaluations, we plan to use the version of the analyzer with additional available means of ambiguity resolution incorporated (see Figure 2 for a brief listing). In fact, we will use different combinations of the procedures for residual ambiguity resolution and recovery from "unexpected" input to determine their relative utility and contributions to the quality of semantic analysis (not only WSD but also semantic dependency determination).

The evaluation of semantic dependency determination is different from that suggested by Gildea and Jurafsky (2002) who designed a system to automatically learn the semantic roles of unknown predicates. First, that system does not actually do WSD; second, it makes assumptions that our work does not: it does not use any language-independent metalanguage to record meaning and concentrates on selectional

restrictions, a far more limited inventory than the set of all possible relations between concepts provided in our ontology.

The evaluation environment we have developed reduces the amount of time necessary to produce a gold standard output for each of the three stages of our analysis process quite dramatically. It is in this sense that it is a very important enabling element for larger-scale evaluation work that from this point on will become standard procedure in our work on building semantic analyzers

	Baseline A	Baseline B	As Is	Correct Preprocessor	Correct Syntax
Abbreviations, numbers, etc.	3/2	3/2	3/2	5/0	5/0
Named entities	14/10	14/10	14/10	24/0	24/0
Parts of Speech	121/83	121/83	121/83	204/0	204/0
Preprocessor Total	0.59	0.59	0.59	1.0	1.0
Phrase boundary score	0.81	0.8	0.91	0.97	1.0
Phrase heads	129/48	127/50	159/25	180/12	182/0
Attachments	86/38	87/37	100/53	166/15	`81/0
Syntax Total	0.74	0.77	0.81	0.94	1.0
WSD	57/54	59/52	63/48	86/25	98/15
WSD complexity	0.61	0.62	0.64	0.85	0.96
WSD distance	0.79	0.80	0.83	0.92	0.96
Semantic dependencies	104/182	113/173	136/150	198/88	229/43

Table 2. Results of the initial evaluation of the OntoSem semantic analyzer.

References

Stephen Beale, Sergei Nirenburg and Marjorie McShane. 2003. Just-in-time grammar. *Proceedings of the 2003 International Multiconference in Computer Science and Computer Engineering*, Las Vegas, Nevada.

Gildea, Dan and Dan Jurafsky. 2002. Automated labeling of semantic roles. *Computational Linguistics* 28(3): 245-288

Sergei Nirenburg, Marjorie McShane and Stephen Beale. 2003. Operative strategies in Ontological Semantics. *Proceedings of HLT-NAACL-03 Workshop on Text Meaning*, Edmonton, Alberta, Canada, June 2003.

Sergei Nirenburg and Victor Raskin. 2004 (forthcoming). *Ontological Semantics,* the MIT Press, Cambridge, Mass.

Onyshkevych, Boyan 1997. Ontosearch: Using an ontology as a search space for knowledge-based text processing. Unpublished PhD Dissertation. Carnegie Mellon University.