Advances in Cognitive Systems 4 (2016) 35–55 Submitted 9/2014; published 6/2016

Language Understanding With Ontological Semantics

Marjorie McShane MARGEMC34@GMAIL.COM
Sergei Nirenburg ZAVEDOMO@GMAIL.COM
Stephen Beale STEPHENBEALE42@GMAIL.COM
Cognitive Science Department, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY USA

Abstract

Ontological Semantics is an approach to automatically extracting the meaning of natural language texts. The OntoSem text analysis system, developed according to this approach, generates ontologically grounded, disambiguated text meaning representations that can serve as input to intelligent agent reasoning. This article focuses on two core subtasks of overall semantic analysis: lexical disambiguation and the establishment of the semantic dependency structure. In addition to describing the knowledge bases and processors used to carry out these tasks, we introduce a novel evaluation suite suited specifically to knowledge-based systems. To situate this contribution in the field, we critically compare the goals, methods and tasks of Ontological Semantics with those of the currently dominant paradigm of natural language processing, which relies on machine learning.

1. Introduction

Human-level, automatic semantic analysis of language has been a core goal of the field of artificial intelligence since its inception. However, work in this paradigm – referred to as AI-NLP – has been largely supplanted over the past 20 years by a knowledge-lean paradigm that concentrates on distributional, statistical, machine learning-oriented methods applied to big data; this paradigm facilitates the configuration of applications that do not address the role that knowledge and reasoning capabilities play in language understanding by people. Indeed, contributions that can be viewed as descendants of AI-NLP count for just a small minority of recent work on automatic language processing, even though both the theoretical and practical prospects of AI-NLP are as tantalizingly promising as ever, albeit in the medium or long term.

In this paper, we describe a system, OntoSem, which is the most recent implementation of the AI-NLP theory called Ontological Semantics (Nirenburg & Raskin, 2004). Although Ontological Semantics has been under development for over 25 years, and although various application systems have implemented the evolving theory over that time, as in Beale et al.'s (1995) Mikrokosmos machine translation system, this is the first in-depth description of the current-generation system. Like any theory-driven implementation, OntoSem reflects a large inventory of methodological decisions and design choices – the devil in the details of translating theory into practice. Among the most important recent design choices was incorporating a statistically-trained syntactic parser into this primarily knowledge-based, psychologically-inspired environment. The practical consequences of this choice, as well as the tacit theoretical assumptions it brought to light, will be a recurring thread in the narrative.

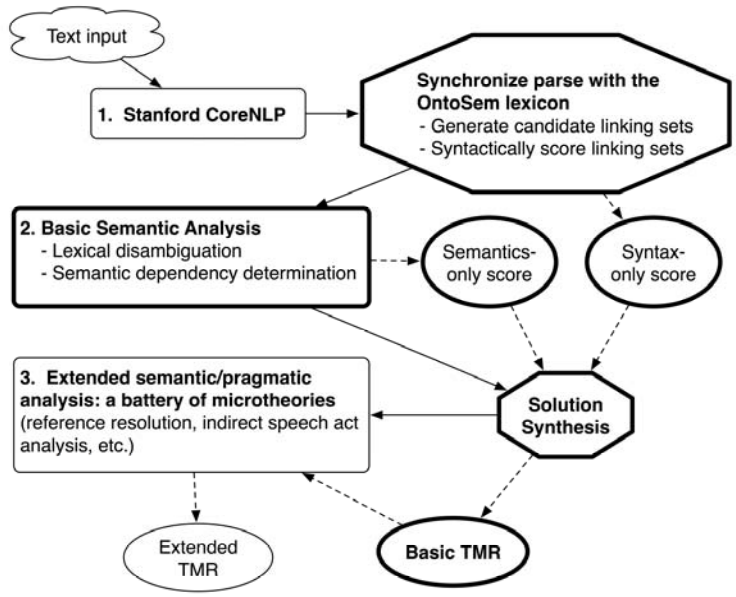

Figure **1**. The architecture of the OntoSem text analysis system. The numbered engines to the left represent the three conceptual stages of text analysis. The processors shown by octagons provide support services. The modules with a thick black border are the focus of this paper. Dotted arrows show data flow and solid arrows show control flow.

The three main stages of language processing in OntoSem are syntactic analysis, basic semantic analysis, and extended semantic/pragmatic analysis, as shown in Figure 1. In this paper we concentrate on *basic semantic analysis*, defined as lexical disambiguation and the establishment of the semantic dependency structure.

In the figure, basic semantic analysis, along with the closely associated processors and knowledge structures we will discuss, are indicated using thick black borders. The result of basic semantic analysis is an ontologically-grounded basic *text meaning representation* (TMR). The basic TMR serves as input to agent reasoning about discourse-level aspects of text meaning, such as reference resolution, the interpretation of indirect speech acts, and other subtasks we have discussed elsewhere.[1] The results of discourse-oriented analysis are recorded in the extended TMR**,** which is used to populate agent memory. Agent memory, in turn, serves as input to an agent's decision making and action.

In applications, OntoSem text analysis is used as the language perception module of the perception-reasoning-action pipeline of artificial intelligent agents, as illustrated by its role in the functioning of the intelligent agents playing the role of virtual patients in the Maryland Virtual Patient clinician training prototype system (Nirenburg et al., 2008; McShane et al., 2013).

1 These include our work on reference resolution (McShane, 2009; McShane & Nirenburg, 2013), verbal ellipsis resolution (McShane & Babkin, 2015), paraphrase interpretation (McShane et al., 2008), and nominal compounding (McShane et al., 2015).

1.1 Theoretical Claims and Methodological Choices

We identify five theoretical claims that are central to Ontological Semantics:

1. Natural language understanding involves translating input strings into ontologically-grounded *text meaning representations* and then storing these knowledge structures in agent memory.
2. Translation into the ontologically-grounded metalanguage (a) focuses on the content of the message rather than its lexico-syntactic form, (b) resolves analysis problems such as lexical and referential ambiguity, underspecification, ellipsis, and paraphrase, and (c) permits the same knowledge bases and reasoning engines to be used for different natural languages (McShane & Nirenburg, 2012; McShane et al., 2005).
3. Language understanding relies on three static knowledge bases – the ontology (for knowledge about concept types), the fact repository (for knowledge about concept instances), and an ontological semantic lexicon. It employs a battery of algorithms that manipulate both linguistic and extra-linguistic knowledge.
4. The global interpretation of text meaning is built up compositionally from the interpretations of progressively larger groups of words and phrases (Beale, 1996).
5. Each meaning interpretation is assigned a confidence level, which reflects the degree to which the global interpretation deviates from the expectations of the supporting knowledge bases and algorithms. In collaborative human-agent applications, confidence levels will help agents to decide whether to act upon their understanding of a language input or, by contrast, to seek clarification from a human collaborator.

In addition to the theoretical claims of Ontological Semantics, the current OntoSem implementation reflects many methodological choices. Some high-level choices include:

1. Sentences are treated as a whole: meaning representations are not built up from left to right.
2. Syntactic and semantic analysis are carried out in a pipeline. This is a consequence of the decision to import a parser that works at the sentence level. We are motivated both theoretically and practically to move to a processing architecture that would both permit incremental, left-to-right processing and let more heuristic evidence be leveraged across analysis modules. For our preliminary work on integrating syntactic, semantic and discourse analysis, see McShane et al. (2010) and McShane and Nirenburg (2013).
3. Syntactic analysis is provided by the statistically trained Stanford CoreNLP toolset (version 3.4.1; Manning et al., 2014), and we make no cognitive modeling-related claims about how it is carried out.
4. We model the selection of a single analysis from multiple candidate analyses using a function that incorporates *bonuses* for precisely aligning with knowledge-based expectations and *penalties* for violating them. This function combines constraints deriving from different aspects of the language system, such as syntax and semantics.
5. Like all such models (Bailer-Jones, 2009), the current OntoSem account is incomplete, not covering every element of knowledge about language and the world. However, it is sufficient to demonstrate the potential of the approach for use in agent systems in the near-term, mid-term, and long-term.

As the above theoretical and methodological statements should make clear, the OntoSem approach to extracting meaning from natural language text is as human-inspired as possible but as methodologically inclusive as necessary. OntoSem is regularly tasked with arriving at a single interpretation of sentences that might contain 20 to 30 words that have an average of five senses each. This is a vast decision space that is best managed by well-known algorithmic techniques, which we incorporate to support human-inspired cognitive modeling.

1.2 Comparisons with Related Work

Before proceeding to the system description, let us briefly mention a few programs of research that most closely resonate with Ontological Semantics.[2] Both Schank's (1972) Conceptual Dependency Theory and Wilks' Preference Semantics (Wilks, 1985; Wilks & Fass, 1992) also pursue expectation-oriented language analysis using semantic lexicons, and generate meaning representations using a form of interlingua. Moreover, we share many of Schank's and Wilks' theoretical views. Unlike Ontological Semantics, however, these theories were not applied to broad-scale text analysis, though they did inspire approaches that were. Schank-like script-based reasoning inspired the Knowledge Machine project (Clark & Porter, 2004), and Preference Semantics was used by Fass (1997) in his treatment of metaphor and metonymy.

A rather different affinity exists between Ontological Semantics and Hobbs' (1989) Axiomatic Semantics, which shares our goals of encoding and leveraging world knowledge in support of language-oriented reasoning, as well as tackling a broad spectrum of difficult semantic analysis problems. However, the focus of Hobbs' development effort is quite different: rather than building broad-coverage lexicons or language processors, he builds axiomatic knowledge bases to support reasoning about linguistic input. His approach might be seen as dovetailing with that of Ontological Semantics by contributing to overall text understanding once certain core semantic analysis processes have been carried out. The treatment of these core semantic analysis processes, realized through the OntoSem text analysis system, is the focus of the present article.

1.3 Content and Organization of the Paper

The remainder of the paper is organized as follows. First we describe the knowledge bases of OntoSem (Section 2) and the target output knowledge structures of OntoSem processing (Section 3). Then we describe how the two core facets of basic semantic analysis – lexical disambiguation and the establishment of the semantic dependency structure – are naturally integrated under our theory (Section 4). Next we describe, in turn, the four processors from Figure 1 that contribute to basic semantic analysis: the syntactic parser (Section 5), the engine that synchronizes the syntactic parse with the OntoSem lexicon (Section 6), the basic semantic analyzer (Section 7), and Hunter-Gatherer (Section 8), a program that combines heuristic evidence to facilitate the construction of a text meaning representation. Section 9 presents a system evaluation and Section 10 offers thoughts about the contributions of this work to intelligent agent development.

---

[2] For a more extensive review, see Nirenburg and Raskin (2004, Chapters 3 and 4). The generational attribution of these programs of research should come as no surprise: AI-NLP has been largely sidestepped by the mainstream NLP for the past 20 years, a trend that may be changing due to evidence that knowledge-lean methods will not solve the fundamental problems of language understanding.

2. The Core Knowledge Bases

OntoSem language analysis relies on two core static knowledge bases: the ontology and the lexicon. The ontology is a formal model of the world that is encoded in a metalanguage for describing meaning derived from any source, be it language, agent perception, agent reasoning, or simulation (McShane & Nirenburg, 2012). The metalanguage of description is unambiguous, permitting automatic reasoning about language and the world to be carried out without the interference of lexical and morphosyntactic ambiguities.[3] The ontology is organized as a multiple-inheritance hierarchical collection of frames headed by concepts that are named using language-independent labels.[4] We describe salient properties of the ontology using an excerpt from the ontological frame for DRUG-DEALING:

DRUG-DEALING		
IS-A	*value*	CRIMINAL-ACTIVITY
AGENT	*default*	CRIMINAL, DRUG-CARTEL
	sem	HUMAN
	relaxable-to	SOCIAL-OBJECT
THEME	*default*	ILLEGAL-DRUG
INSTRUMENT	*default*	MONEY
HAS-EVENT-AS-PART	*sem*	BUY, SELL

Concepts divide up into EVENTs, OBJECTs, and PROPERTYs. PROPERTYs are primitives, which means that their meaning is understood to be grounded in the real world without the need for further ontological decomposition. The facets *value, default*, *sem,* and *relaxable-to* allow for recording progressively less typical fillers for properties. The OntoSem ontology currently contains 9,075 concepts which are described by an average of 15 properties each. Property values can be locally defined or inherited from parent concepts.

Since the OntoSem ontology is language independent, its link to any natural language must be mediated by a lexicon. Consider, for example, the first two verbal senses for *address*, shown in Table 1. Syntactically, both expect a subject and a direct object in the active diathesis, filled by $var1 and $var2, respectively.[5] However, in address-v1, the meaning of the direct object (^$var2) is constrained to a HUMAN (or, less commonly, ANIMAL), whereas in address-v2 the meaning of the direct object is constrained to an ABSTRACT-OBJECT. (The constraints appear in italics because they are virtually available, being accessed from the ontology by the analyzer at runtime.) This difference in constraint values permits the analyzer to disambiguate: if the direct object is abstract, as in *He addressed the problem*, then *address* will be analyzed as CONSIDER; by contrast, if the direct object is human, as in *He addressed the audience*, then *address* will be analyzed as SPEECH-ACT.

3 A description of, and rationale for the form and content of, the ontology is available in Nirenburg and Raskin (2004, Section 7.1). An axiomatic definition, originally due to Kavi Mahesh, is presented in Section 7.1.6 of that work.

4 As described in Nirenburg and Raskin (2004), these labels are similar to English words only for the benefit of knowledge engineers. For purposes of machine processing, they could as easily be random sequences of characters, since a concept is defined by its inventory of property values. This ontology is equally useful for languages other than English because word senses in those languages will map to universal ontological concepts.

5 Variables are written, by convention, as $var followed by a distinguishing number. Variables permit us to map textual content from the input to elements of the syn-struc, then link each syn-struc element with its semantic realization in the sem-struc.

Table 1. Two verbal senses for the word *address*. The symbol ^ indicates “the meaning of”.

address-v1	address-v2
address-v1 anno definition “to talk to” example “He addressed the crowd.” syn-struc subject $var1 v $var0 directobject $var2 sem-struc SPEECH-ACT AGENT ^$var1 *(sem HUMAN)* BENEFICIARY ^$var2 *(sem HUMAN) (relax.-to ANIMAL)*	**address-v2** anno definition “to consider, think about” example “He addressed the problem.” syn-struc subject $var1 v $var0 directobject $var2 sem-struc CONSIDER AGENT ^$var1 *(sem HUMAN)* THEME ^$var2 *(sem ABSTRACT-OBJECT)*

These examples highlight several aspects of the OntoSem lexicon. First, it supports the combined syntactic and semantic analysis of texts. Second, the metalanguage for describing its sem-strucs is the same one used in the ontology. And third, the sem-strucs—and, often, the associated syn-strucs—from the lexicon for one language can be ported into the lexicon of another language with minimal modification (McShane et al., 2005), which greatly enhances the multilingual applicability of the OntoSem suite of resources. The English lexicon currently contains 21,041 superentries covering 28,925 word (and phrase) senses, of which 4,991 are verbal, 18,365 are nominal, and 5,569 represent other parts of speech.

3. The Output of Semantic Analysis: Text Meaning Representations

Text meaning representations (TMRs) are ontologically-grounded, disambiguated, semantic interpretations of language input (Nirenburg & Raskin, 2004, Chapter 6) that are generated by the OntoSem analyzer. Each TMR frame is headed by a numbered instance of an ontological concept and is related to other frames (i.e., other instantiated concepts) using ontologically defined relations. Consider an example sentence and the associated TMR in Table 2, which contains four frames. This TMR reflects the processing of this sentence in isolation, making the instance number for all concepts “1”.

(1) Manometry can be used to confirm the diagnosis.

The core proposition in this sentence is “manometry (is) used to confirm the diagnosis”. The semantic head is an instance of the concept CONFIRM (CONFIRM-1), whose THEME is an instance of the concept DIAGNOSIS and whose INSTRUMENT is an instance of the concept MANOMETRY. This proposition is scoped over by an instance of the concept MODALITY, whose type is “potential” and whose value is “1” – the highest value on the abstract{0,1} scale. Italics indicate metadata, which is used by developers for evaluation and debugging.

This text meaning representation is a *basic* TMR because it reflects only the results of basic semantic analysis, consisting of lexical disambiguation and semantic dependency determination. At this stage, the system has not yet attempted to resolve the coreference of *the diagnosis* or determine if this is an indirect speech act (it is not). In this paper, we concentrate on the automatic generation of such *basic* TMRs.

The main benefit of grounding the representation of text meaning in an ontology is that the system can use all ontologically-recorded information about the component concepts in its reas-

Table 2. The text meaning representation for example sentence (1).

CONFIRM-1 THEME DIAGNOSIS-1 INSTRUMENT MANOMETRY-1 SCOPE-OF MODALITY-1 MODIFIER-FROM-SENSE (INSTRUMENT USE-V4) *word-num* *5* *textpointer* *confirm* *from-sense* *confirm-v1*	**MODALITY-1** TYPE POTENTIAL VALUE 1 SCOPE CONFIRM-1 *word-num* *1* *textpointer* *can* *from-sense* *can-aux1*
MANOMETRY-1 INSTRUMENT-OF CONFIRM-1 *word-num* *0* *textpointer* *manometry* *from-sense* *manometry-n1*	**DIAGNOSIS-1** THEME-OF CONFIRM-1 *word-num* *7* *textpointer* *diagnosis* *from-sense* *diagnosis-n1*

oning. For example, since the ontology asserts that MANOMETRY is a descendant of MEDICAL-PROCEDURE, and that MANOMETRY is used to test for a particular inventory of ANIMAL-DISEASEs, the system knows – as would a well-informed human – that this text is most likely talking about the event of DIAGNOSE whose THEME is one of those ANIMAL-DISEASEs. Now that the goal of, and rationale for, deep-semantic analysis should be clear, let us move on to the stages of processing required to automatically generate basic TMRs.

4. Word Sense Disambiguation and Semantic Dependency Determination

As just mentioned, this paper focuses on the automatic generation and evaluation of what we call *basic* text meaning representations (TMRs), which contain two types of semantic information: disambiguated word senses and the semantic dependencies among them.

Word sense disambiguation must be considered a core task in natural language understanding if one believes, as we do, that intelligent agents must ultimately be able to understand and reason about meaning to a degree that approximates the competence of people. In the best case, OntoSem processing yields a single, confident sense selection for each word. However, there can also be residual ambiguity, in which the analyzer cannot select definitively among multiple candidates, or the zero-candidate outcome.

Residual ambiguity typically derives from one of two sources. On the one hand, the relevant constraints in the lexicon or ontology might be insufficiently narrow. Consider, for example, the word *operation*, which has, among others, a military sense, MILITARY-OPERATION, and a medical sense, SURGERY. Let us assume that, by some oversight of the person acquiring the ontology, the default AGENT for both of these senses was listed as HUMAN, which implies that all humans are equally appropriate fillers for that case-role. Given an input like

(2) The surgeon started the operation at 10:00.

the system would have no reason to prefer one meaning of *operation* over the other. However, if the ontological constraints were tighter, as, indeed, they are in the current version of the ontology (shown in Table 3), then the system would readily select the SURGERY meaning because a central tenet in disambiguation is to prefer slot fillers that correspond to tighter semantic constraints.

Table 3. Constraints on two senses of the word *operation* in the ontology.

SURGERY			MILITARY-OPERATION		
AGENT	*default*	SURGEON	AGENT	*default*	MILITARY-ROLE
	sem	PHYSICIAN		*sem*	HUMAN
	relaxable-to	HUMAN		*relaxable-to*	GEOPOLITICAL-ENTITY

The second source of residual ambiguity is significantly more challenging: ambiguity in which the local context of the word, as defined by syntactic dependencies, does not provide sufficient disambiguating evidence. Consider two examples from a project in which we processed a corpus of sentences about piracy.

(3) The French navy handed over nine suspected pirates to the authorities in Puntland.

(4) After a four-month saga, $3.2 million in cash was dropped by parachute – and the pirates left the ship in February 2009.

To a human – a powerful reasoning machine – these examples are clearly about pirates at sea, not infringers of copyright; but the knowledge required to reach this conclusion is not ontologically recorded as constraints on the case roles of the given events: TRANSFER-POSSESSION (instantiated by *handed over*) can have any PHYSICAL-OBJECT as its THEME, and MOTION-EVENT (instantiated by *left*) can have any ANIMAL as its AGENT. In both cases, the semantic clues supporting disambiguation come from outside of the local semantic dependency structure. This paper does not address extra-clausal reasoning of this type; we mention this source of residual ambiguity simply to emphasize that clause-level processing described below must be viewed within the larger context of text-level semantic and pragmatic analysis.

Apart from word sense disambiguation, the second requirement for generating a basic TMR is semantic dependency determination, which is supported by the linking information in the OntoSem lexicon. In contrast to the widely pursued NLP task known as “shallow parsing” or “case role analysis” – which labels dependencies between uninterpreted text strings (Gildea & Jurafsky, 2002) – semantic dependency determination in OntoSem involves determining the semantic dependencies among semantically analyzed entities (ontological concepts). As such, word sense disambiguation and semantic dependency determination in OntoSem are tightly integrated processes. The main challenges of semantic dependency determination involve interpreting the semantic dependencies in: a) input that contains less common, non-basic syntactic diatheses (e.g., *Comedy I like!*); b) input that is unexpected with regard to the current state of the lexicon (i.e., a legal syntactic configuration might not have been recorded); and c) freely occurring but polysemous adjuncts, whose lexical disambiguation and semantic dependency structure must be simultaneously computed (e.g., *with nuts* is a THEME in *I ate ice cream with nuts,* but an INSTRUMENT in *I fed the squirrel with nuts*).

In procedural terms, word sense disambiguation and semantic dependency determination are interdependent because they rely on the same knowledge. Consider the first verbal sense of *read* (read-v1), shown in Table 4. If this sense is selected as the right answer when analyzing a clause headed by *read* (e.g., *John read the book*), this has several consequences: an instance of the concept READ will be instantiated to reflect the meaning of *read* (a word sense disambiguation

Table 4. The first sense of the word *read* in the ontology.

```
read-v1
    syn-struc
        subject        $var1
        v              $var0
        directobject   $var2
      sem-struc
        READ
          AGENT        ^$var1    (sem HUMAN)
          THEME        ^$var2    (sem BOOK-DOCUMENT)
```

decision); the meaning of the subject will fill the AGENT case-role (a semantic dependency determination decision); and the meaning of the direct object will fill the THEME case-role (another semantic dependency determination decision).

The remainder of the article describes the engines that contribute to word sense disambiguation and semantic dependency determination in OntoSem. Since the description of basic analyzer operation is, in and of itself, rather complex, we exclude from this exposition special cases – such as the treatment of unexpected input – that would considerably lengthen the narrative.

5. Syntactic Analysis

One of the principles of Ontological Semantics is that syntactic analysis, while not an end in itself, can contribute to semantic analysis – an approach operationalized by the linked syn-strucs and sem-strucs of the lexicon.

As mentioned earlier, we make no theoretical claims about human syntactic processing and, for practical reasons, use the Stanford CoreNLP toolkit (Manning et al., 2014) to provide heuristic evidence from the realm of syntax. This parser carries out phrase structure analysis as well as syntactic dependency analysis. Syntactic dependency analysis returns a set of syntactic dependency labels along with the text elements, numbered by sentence position, that serve as their arguments. For example, the parser returns 14 dependencies for the sentence

(5) Studies have shown that repeated concussions can result in permanent injury to the brains of football players.

For illustration, consider the three dependencies that involve the word *shown*:

- ***nsubj****(shown-3, Studies-1)* indicates that Studies-1 is the nominal subject of shown-3
- ***aux****(shown-3, have-2)* indicates that *have-2* is an auxiliary whose main verb is *shown-3*
- ***ccomp****(shown-3, result-8)* indicates that the clause headed by *result-8* is the clausal complement of *shown*-3.

Integrating an external parser as a replacement for the homegrown parsers we used for many years has yielded better coverage of syntactic phenomena, but it has required certain additional development efforts. For example, we had to record knowledge about how to align parser output with the syntactic expectations of argument-taking words recorded in OntoSem lexicon entries, a process described in McShane et al. (2015). Runtime computation of such alignments is particularly difficult when sentences include multiple non-basic syntactic structures, such as the passive voice and reduced relative clauses, as illustrated in

(6) [[[The article [written by the president of the student body]$_{\text{REDUCED-RELATIVE-CLAUSE}}$]$_{\text{NP}}$ was criticized by the administration.]$_{\text{PASSIVE-VOICE-CLAUSE}}$

Another necessary enhancement was preparing the system to reambiguate parsing decisions that cannot, in principle, be made confidently without the contribution of semantics. For example, in *Sue stroked the dog with the plush toy*, either the dog can be in possession of the toy, or Sue can be using the toy as an instrument of stroking – an example of ambiguous prepositional phrase attachment. Similarly, in the nominal compound *living room window safety lock* the pairwise internal structure of the nominal compound requires semantic analysis: [[[living room] window]] [safety lock]]. So, although all aspects of the syntactic parse are considered defeasible evidence by the semantic analyzer, these frequent, expected phenomena deserved explicit treatment.

6. Synchronizing the Parse with the OntoSem Lexicon

The Stanford CoreNLP parser outputs an inventory of syntactic dependencies among the words in the input sentence. The task now is to align the syntactic dependencies returned in the parse with the syntactic *expectations* for each candidate sense of each word, as recorded in the OntoSem lexicon. This alignment yields a *syntactically-motivated* preference for particular lexical senses. Since a lexical sense in the OntoSem environment contains both syntactic and semantic zones linked by variables, the syntactically motivated preference for a lexical sense directly translates into a preference for a particular semantic analysis of that word and particular semantic dependencies in which it participates. Returning to our example of read-v1, if syntactic analysis prefers the read-v1 lexical sense as the solution for the verb *read* in *John read the book*, this syntactically motivated preference is simultaneously weighing in on semantics, since read-v1 includes a sem-struc that interprets *read* as READ and expects its AGENT and, optionally, THEME slots to be filled.

The process just described encapsulates OntoSem's use of syntactic analysis to support semantic analysis. Operationalizing this process has required extensive effort, the details and scope of which deserve some comment, since they reveal what is involved in leveraging a syntactic parse in service of semantic analysis. We describe the component steps of the process in turn, albeit very briefly.

Optional. Reambiguate the parse, if applicable. Several aspects of syntactic structure, including prepositional phrase attachments and nominal compound bracketing, cannot be determined without invoking semantics. Since the parser returns just one solution for such cases, OntoSem detects and reambiguates them, offering all available analyses for consideration by the semantic analyzer.

Step 1: Retrieve all syntactic dependencies involving the word. Syntactic dependencies involving each word are output by the parser. As mentioned earlier, for the word *shown* from (5) above, three dependencies are retrieved: **nsubj**(shown-3, Studies-1), **aux**(shown-3, have-2), **ccomp**(shown-3, result-8). We refer to these as the *actual syntactic dependencies* since they are what actually occurs in the parse of the input sentence.

Table 5. Lexical senses for two verbal senses of *show.*

show-v1	show-v4
definition "to display physically" example "He showed me his new car." syn-struc subject $var1 v $var0 directobject $var2 indirectobject $var3 (opt+) sem-struc DISPLAY AGENT ^$var1 THEME ^$var2 BENEFICIARY ^$var3	definition "to demonstrate, with a non-agentive subj." example "Experience shows (us) this won't work." syn-struc subject $var1 v $var0 comp $var2 indirectobject $var3 (opt+) sem-struc PROVE INSTRUMENT ^$var1 THEME ^$var2 BENEFICIARY ^$var3

Step 2: Retrieve all lexical senses for the root word. For the word *show*, the system retrieves ten senses from the OntoSem lexicon[6] but, for purposes of illustration, we will describe only the first and the fourth verbal senses. Syntactically, both expect a subject and an optional indirect object, but whereas show-v1 expects a direct object, show-v4 expects a comp (i.e., a verbal complement clause). We present the sem-strucs of these senses in Table 5 only for the sake of completeness. Remember that, at this point of the process, the analyzer is not yet considering semantics at all – it is seeking syntactically-based votes for lexical senses when analyzing the sentence *Studies have shown that repeated concussions can result in permanent injury to the brains of football players.*

Step 3. Determine the candidate linking sets for each lexical sense. We define a *candidate linking set* as a particular lexical sense for a word along with a particular assignment (linking) of its variables to words in the input text. A linking set basically says, "Let us assume that a given lexical sense is the correct one for a given argument-taking word, and let us link its variables to words in the text in one particular way"; that will then act as a candidate solution for analyzing the word, which will then be evaluated as part of the overall problem of analyzing the sentence. For our *Studies have shown* example, whose relevant syntactic dependencies were presented earlier, the system generates the candidate linking sets in Table 6.

Step 4: Score each candidate linking set using heuristics. A linking set receives a high score if the expected OntoSem and parser dependencies are close according to a distance metric (OntoSem's "subject" and the parser's "nsubj" are very close), all required syn-struc variables are bound, optional syn-struc variables are bound (this receives only a slight scoring bonus, since it is not always the best choice), bindings involve an appropriate part of speech, any words specified in the syn-struc (like the direct object *bucket* in the idiom *kick the bucket*) have the appropriate head and features, and, if the word in question is a verb, the candidate set includes a subject linking. Scoring penalties should be adjusted according to text genre: e.g., elided subjects are to be expected in informal writing styles like text messaging.

[6] Most of these reflect different syntactic uses of the verb. Some are phrasals like *show up*, whereas some are nominal senses. Many more phrasals would need to be included to offer full coverage of uses of this word.

Table 6. Candidate linking sets. Each column represents a different candidate linking set. S(tudies) and R(esult) can be bound to any variable or can be left unbound.[7]

$var1	S	S	R	R			S			R			
$var2	R		S		S	R		S			R		
$var3		R		S	R	S			S			R	

Step 5. Prune the candidate set. Optionally, candidate linking sets whose scores are below a threshold (calculated as a percentage of the score of the highest-scored set) are pruned. Pruning at this stage implies that we have high confidence in the outcome of syntactic analysis and its ability to inform semantic decisions. When pruning is disabled – as would be appropriate for non-normative text genres like text messaging – all candidate sets are available to be evaluated semantically.

This concludes the description of how the results of syntactic analysis are used to create and score candidate linking sets that are then passed on for semantic analysis. In some cases, the candidate sets for different words will be incompatible. At the next stage of analysis, the algorithm will ensure that only compatible sets are combined in the final solution.

7. Basic Semantic Analysis

At this point, the analyzer has a list of linking sets for each word sense in the sentence. Each one is a candidate partial solution to overall sentence analysis. The analyzer also has a purely syntax-based score for each linking set, derived by comparing the parser's analysis of the actual use of each word in the context with the syntactic expectations for its use recorded in the syn-struc of each lexical sense of the word.

Now the analyzer undertakes to calculate a semantic score for each candidate variable binding in each lexical sense. It is important to note that syntactic scores and semantic scores are independent. In this step we are saying, *If we were to select a given lexical sense with a given variable binding – no matter how likely or unlikely that binding was judged to be on syntactic grounds – how good would it be semantically*? This semantic evaluation actually involves two processes. If the filler matches the expectations of one of the ontological facets (*default*, *sem*, or *relaxable-to*), then it is scored according to the tightest constraint it matches (e.g., SURGEON is scored very highly as the AGENT of SURGERY since it fills the *default* facet). If the filler does *not* match the expectations of any of the facets, then a special function (Onyshkevych, 1997) calculates the ontological distance between the actual filler and what is expected to be the filler. For example, the input *The red tie performed the surgery* violates the ontologically recorded expectation that the AGENT of SURGERY will at least be HUMAN (which fills the *relaxable-to* facet of the AGENT of SURGERY); however, the violation is explained by a chain of properties that metonymically links a person to the clothes he is wearing.

Table 7 summarizes two candidate solutions for the problem of selecting a word sense and argument bindings for the word *show* in our sample sentence. First consider the show-v1 column. If show-v1 is selected, then the meaning of *show* will be understood as DISPLAY; its AGENT slot will be filled by some meaning of 'study' and its THEME slot will be filled by some meaning of 'result'. The sem-struc, in combination with information about DISPLAY recorded in the ontology,

[7] In the sentences *He eats all day* and *She hit the ball with enthusiasm*, the adverbials *all day* and *with enthusiasm* should not be bound. Leaving all variables unbound is among the available options, as shown in the last column.

Table 7. Two candidate solutions for word sense disambiguation and semantic dependency determination for the word *show* in our sample sentence.

	show-v1	show-v4
Binding combination being evaluated	$var1 = studies $var2 = result	$var1 = studies $var2 = result
Syn-struc of verb sense being evaluated	$var0 - subject $var1 - directobject $var2 - pp $var3	$var0 - subject $var1 - comp $var2
Sem-struc of verb sense being evaluated	DISPLAY - AGENT ^$var1 - THEME ^$var2 - BENEF. ^$var3	PROVE - INSTR. ^$var1 - THEME ^$var2 - BENEF. ^$var3
Ontologically recorded constraints on head concept	DISPLAY - AGENT (HUMAN) - THEME (PHYSICAL-OBJECT) - BENEFICIARY (ANIMATE)	PROVE - INSTRUMENT (EVENT, OBJECT) - THEME (EVENT, THEORY, INFO., …) - BENEFICIARY (HUMAN)
Do any senses of the candidate binding fit the semantic constraints?	1. Is any meaning of "studies" HUMAN? (no) 2. Is any meaning of "result" a PHYSICAL-OBJECT (yes)	1. Is any meaning of "studies" an EVENT or OBJECT? (yes) 2. Is any meaning of "result" an EVENT, THEORY, INFO….? (yes)

tells us that the AGENT of DISPLAY should be HUMAN, and the THEME of DISPLAY should be PHYSICAL-OBJECT. So the question is, do any of the lexically recorded meanings of 'study' fulfill the HUMAN constraint, and do any of the lexically recorded meanings of 'result' fulfill the PHYSICAL-OBJECT constraint?

The analysis process described in the next section will conclude that the answer is that none of the available meanings of 'study' is HUMAN, which makes this sense unlikely to be correct. Now consider show-v4. If it is selected, then the meaning of 'show' will be understood as PROVE; its INSTRUMENT slot will be filled by a sense of 'study' and its THEME slot will be filled by some meaning of 'result'. The ontology tells us that the INSTRUMENT of PROVE can be an OBJECT or EVENT, and that the THEME of PROVE can be an EVENT, THEORY, INFORMATION, or any of a number of other semantic types. How well do the lexically recorded meanings of 'study' fulfill this EVENT/OBJECT constraint, and how well do the lexically recorded meanings of 'result' fulfill the EVENT/THEORY/INFORMATION constraint? The answer to both is 'perfectly', making show-v4 much more semantically suitable for this input than show-v1. Semantic analysis of this kind provides the semantic score which is then combined with the syntactic score computed previously to select the best overall analysis of an input sentence.

8. Combining the Syntactic and Semantic Scores

The previous two sections have shown that a considerable amount of syntactic and semantic processing is required to make just one lexical disambiguation decision. Considering that sentences in news articles can be 20 to 30 words long, the number of possible analyses of an average-sized newspaper sentence can be prohibitively large (each word can have multiple word senses and each word sense can have multiple linkings). Clearly, techniques must be developed to avoid this complexity. The key insight of our work is that natural language problems can almost always be viewed as bundles of tightly constrained subproblems that combine at higher levels to produce a complete solution. Beale (1996) shows that syntactic and semantic constraints effectively partition discourse into clusters of locally interacting networks. Here, we summarize

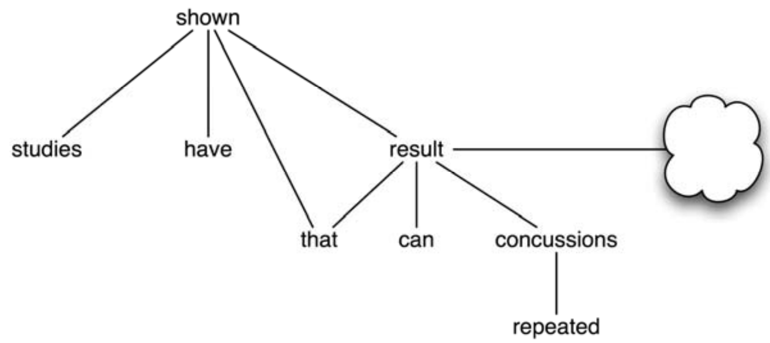

Figure **2**. A constraint diagram for the beginning of the sentence *Studies have shown that repeated concussions can result in permanent injury to the brains of football players*. The cloud represents the rest of the sentence, which has its own constraints.

that discussion, describing how solution synthesis and branch-and-bound techniques can be applied to the problem of natural language analysis.

The OntoSem module devoted to this problem is called Hunter-Gatherer, referring to two tools for finding solutions and achieving higher efficiency in problem solving – hunting and gathering. *Hunting* refers to reducing the search space by looking for suboptimal or impossible solutions and removing/killing them. *Gathering* suggests efficiently extracting answers to subproblems; collecting and combining satisfactory answers. Our approach involves: (1) using semantic dependency information to partition problems into manageable subproblems; (2) combining (gathering) results from these subproblems using a solution synthesis technique; and (3) pruning (hunting) these results using branch-and-bound techniques, which rejects alternatives that are known to be worse than the current best answer (Winston, 1984).

Let us reiterate the task set for the analysis engine: It must select one of the candidate lexical senses for each word of input. Since lexical senses for all words except simple nouns include a head-and-argument structure, selecting a lexical sense involves both word sense disambiguation and semantic dependency determination. A naïve approach to selecting lexical senses would be to try all combinations of candidates, but this would be impractical for any normal-length sentence since an n-word sentence with an average of m linking candidates per word has m^n possible combinations. Hunter-Gatherer achieves the same result as the naïve approach, but in an efficient manner that still promises to find the best solution as defined by the input parameters, including the weights of different features.

Hunter-Gatherer uses dynamic programming (e.g., Cormen 2001, pp. 327–8), a well-known method of solving a larger problem by breaking it down into simpler subproblems, then combining the solutions to the simplest subproblems to efficiently solve the entire problem. Consider, again, the excerpt from example (5), *Studies have shown that repeated concussions can result...*, which Figure 2 shows as a constraint diagram. The diagram should be understood as the merged superset of all candidate semantic constraints posited during earlier stages of processing. As such, each line represents some semantic constraint (we do not know which one yet), and each

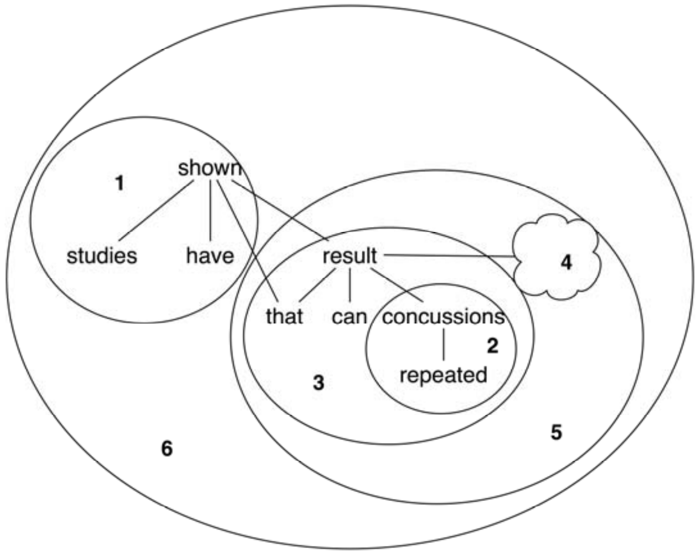

Figure 3. A constraint diagram for the beginning of the sentence *Studies have shown that repeated concussions can result in permanent injury to the brains of football players*, divided into subproblems.

word represents a text-level element (not yet its semantic interpretation). Hunter-Gatherer determines the best combination of word senses and semantic constraints that hold between those word senses. After it evaluates all candidate solutions for a subproblem, it labels the nodes of the graph with concepts and the lines with semantic dependency types.

Once Hunter-Gatherer has created a superset semantic constraint graph (which encodes the knowledge that some constraint might hold between the linked text elements, whatever their lexical senses), it identifies subproblems that are relatively isolated, as Beale (1996) describes in detail. Figure 3 shows a portion of the example sentence's constraint diagram divided into subproblems, numbered 1 to 6, roughly in the order of smallest to largest. This is also the order in which Hunter-Gatherer will process the subproblems, starting with smaller ones and using their results to process larger ones. Subproblem 4 represents the rest of the sentence, which will have its own subproblems that are not shown.

The key heuristic for creating the subproblems (the circles in Fig. 3) is to minimize the number of words that have one or more constraint lines leading outside of the subproblem. Such lines indicate that resolution of the given element relies on outside input; the more numerous the pending constraints, the less progress can be made on the solving the given subproblem in isolation. For example, for Subproblem 1, which contains the words *studies, have,* and *shown,* only the word *shown* has constraint lines that leave the circle, which is a good state of affairs for the Hunter-Gatherer approach.

The main principle behind Hunter-Gatherer can be illustrated by applying it to Subproblem 1, which includes three words. Let us make three simplifying assumptions for the sake of exposition: (1) each word has three senses, yielding 27 combinations of word senses; (2) there is no ambiguity of semantic dependencies (e.g., case-role-A always holds between *shown* and *studies*, and case-role-B holds between *shown* and *have*), such that our total for word sense and linking combinations remains 27; (3) each combination can be scored based on available heuristics.

For each interpretation of *shown* (Shown:1, Shown:2, Shown:3, which illustrate three candidate sets for *shown*), there is a combination of interpretations of *studies* and *have* that maximizes the overall score, indicated in boldface: [Shown:1 Studies:1 Have:2 **0.9**] [Shown:2 Studies:1 Have:3 **0.6**] [Shown:3 Studies:1 Have:1 **1.0**]. The result of Hunter-Gatherer's evaluation of Subproblem 1 is that, of the original 27 candidate solutions, it retains only these three. Although the last combination has a higher score than the other two, the system cannot, at this stage, eliminate the latter because there are constraints outside this subproblem that have not yet been taken into account. For example, the third candidate of *shown* might have low semantic constraint scores when combined, in Subproblem 6, with any candidate of *result*. And in general, further global constraints might overwhelm any local preference. Also note that ties in scoring (e.g., if there was another combination that used Cand3 of *shown* with a score of 1.0) are recorded so that the final, single global answer can be expanded if necessary. Thus, although the input complexity of this sub-problem is 27, corresponding to the 27 exhaustive combinations, the output complexity is only 3. When smaller subproblems are used as input to a larger one, it is only the output complexity that matters; the original input complexity is effectively obliterated.

To conclude, the core analysis engine of OntoSem – Hunter-Gatherer – carries out the process described above to find what it considers to be the best global semantic interpretation of an input, which it outputs in the form of a text meaning representation. The objective accuracy of this result depends upon many things: the quality of evidence provided by the parser; the accuracy and coverage of OntoSem lexicon and ontology; the effectiveness of various scoring mechanisms, which were configured by a mixed process of knowledge-based development and evidence-based training; and the extent to which the input is linguistically regular – i.e., correlates with the expectations of component analysis engines.

9. Evaluation of OntoSem

There are no established evaluation protocols for systems that attempt deep semantic analysis of text. We attempted to configure an evaluation procedure that would give the analyzer a fair opportunity to demonstrate its capabilities while neither overwhelming it with complexity (e.g., requiring it to fully interpret 40-word sentences) nor reducing the endeavor to a toy exercise. In what follows we describe a realistic framework for demonstrating the potential of our approach to offer near-term, mid-term, and long-term solutions for the problem of natural language understanding by intelligent agents.

The hypothesis we test here is that OntoSem can (a) disambiguate head verbs and (b) establish the correct dependency structure for its main arguments in the same way as a human would, given that both the system and the human represent meaning using the same lexical and ontological knowledge bases. We tested the hypothesis by having the system generate text meaning representations, then having a person check their accuracy – which was much faster than, and served the same purpose as, having the person create TMRs from scratch using the same knowledge bases. Our evaluation confirmed the hypothesis but underscored the need to continue to increase the size and coverage of OntoSem's static knowledge sources. We present, in order, the experimental setup, its results – including error classification – and conclusions.

9.1 Experimental Design

We characterize our experimental design in terms of eight decisions, along with their rationales:

1. We focused on verbal disambiguation and the establishment of the correct semantic dependency structure, as in CONCEPT (AGENT ___) (BENEFICIARY___). We did not evaluate disambiguation of the arguments because this often requires reference resolution and/or larger analysis of the discourse context, which was not our focus here (cf. the earlier discussion of the pirate examples).
2. We included in our test set an automatically selected subset of verbs from the OntoSem lexicon that met two criteria: they had more than one sense, and their syntactic and/or semantic descriptions made it possible to disambiguate them at the clause level. An example of verbs that do not meet the latter criterion are those that have both a physical and a metaphorical sense with identical arguments. For example, if person A attacks person B, he might be physically assaulting him or criticizing him, something that can only be determined using additional knowledge about the context.
3. Our evaluation corpus included four Sherlock Holmes stories: “A Scandal in Bohemia”, “The Red-Headed League”, “A Case of Identity”, and “The Boscombe Valley Mystery”. We selected these because they are freely available as part of Project Gutenberg [EBook #1661] and, to our knowledge, nobody has recorded linguistic annotations of these works, so there can be no question that our system operates on raw, unrestricted language input.
4. We automatically selected 200 sentences containing verbs from the selected set, processed them fully automatically, then manually evaluated the resulting TMRs.
5. One developer carried out the evaluation, with selective collaboration by another.
6. The evaluation involved not only detecting errors, but attempting to trace them back to a source that could be fixed to improve system functioning in the future.
7. We did not amend the lexicon or ontology – which had been acquired during work on various projects over many years – in any way to prepare for this evaluation.
8. The experimental setup included challenges of a type that are often filtered out of evaluation suites. For instance, some examples did not contain sufficient information to be properly disambiguated (e.g., semantically underspecified pronouns filled case roles) and other examples reflected what might be considered currently non-normative grammar. However, considering the importance of automatically processing non-standard language genres (texting, email, blogs), we felt it appropriate to make the system responsible for all encountered phenomena.

Together, these decisions produced a design that differs substantially from those commonly used in studies of natural language processing.

9.2 Experimental Results

Of the 200 examples, 92 verbal disambiguation decisions were correct (46%) and 120 of 151 dependency decisions were correct (79%). The vast majority of disambiguation errors (71.3%) resulted from the absence of the needed lexical sense in the OntoSem lexicon.[8] Often, the missing

[8] The rate of recording of lexical senses in OntoSem depends on whether or not the ontology already contains the concepts needed to describe the sense. If so, recording a sense can take as little as a minute. If not, or if a sense is particularly difficult to describe (such as *privacy, loneliness,* and *alien*), then acquisition can take far longer. In

sense was part of an idiomatic construction that had not yet been acquired, such as *draw* (close) *the blinds* in

(7) "The **drawn** blinds and the smokeless chimneys, however, gave it a stricken look, ..."

In other cases, the needed semantic representation (sem-struc) was available but the needed syntactic realization (syn-struc) was not. For example, to process

(8) "She became restive, insisted upon her rights, and finally **announced** her positive intention of going to a certain ball."

the lexicon must permit *announce* to take a direct object, whereas the current lexicon entry expected a clausal complement. Another noteworthy type of error involved the use of underspecified referring expressions as arguments of the verb. A verbal argument like *it* in

(9) "I walked round it and **examined** *it* closely from every point of view, but without noting anything else of interest."

gives the analyzer no evidence for disambiguation, prior to reference resolution, if more than one lexical sense permits the given argument to refer to a non-human object or an event. Here, the analyzer selected the abstract event ANALYZE, which expects an ABSTRACT-OBJECT as the THEME, rather than the physical event VOLUNTARY-VISUAL-EVENT, which expects a PHYSICAL-OBJECT as the THEME.

In four cases, the disambiguation error was due to a missing syntactic transformation. Transformations are used to turn nonbasic syntactic structures, such as the passive voice, into the basic structures that are recorded in the OntoSem lexicon, such as the active voice. Among the transformations not yet covered by OntoSem is Noun+Verb$_{\text{PROGRESSIVE}}$ → EVENT (THEME OBJECT), an omission that led to the misanalysis of *hang* in

(10) "When, in addition, I see a Chinese coin **hanging** from your watch-chain, the matter becomes even more simple."

The final type of error that deserves comment is unexpected input in the form of grammatical constructions that are not sufficiently canonical (at least in modern-day English) to be recorded in the lexicon. For example, the verb *pronounce* in

(11) "I found the ash of a cigar, which my special knowledge of tobacco ashes enables me to **pronounce** as an Indian cigar."

is used in the non-standard construction *X pronounces Y <u>as</u> Z.* The remainder of fine-grained error classification is primarily of interest to developers, with issues ranging from parsing errors to what might be called genuinely very difficult input, as in the elliptical sentence

(12) "We both thought the best resource was flight, when **pursued** by so formidable an antagonist; so you will find the nest empty when you call tomorrow."

in which the THEME of PURSUE is not within *pursue*'s minimal clause and, therefore, cannot, under the current system configuration, inform disambiguation.

---

additional, tactical decisions regarding the prioritization of lexical acquisition must be made. If one missing sense of a lexeme is identified, should all other senses – including all phrasal senses – be acquired at once? Needs of specific projects typically dictate this type of decision making.

Table 8. Error classification for verbal sense selection on an evaluation suite of 200 examples.

Error Type	Errors	Percentage of Errors
The error can be fixed by adding or adjusting a lexical sense.	77	71.3%
Disambiguation is impeded by an underspecified argument (e.g., ‘it’).	3	2.7%
A needed syntactic transformation was missing.	4	3.7%
Unexpected input (non-canonical English)	6	5.6%
Miscellaneous	18	16.7%
TOTAL	108	100%

Table 8 summarizes the results of this evaluation exercise. The evaluation served its goal of validating that OntoSem works as designed: it generates the same disambiguation and dependency decisions for inputs as a person would *when the lexical and ontological resources are sufficient to cover the given inputs*. As expected, most errors are attributable to lexical insufficiencies – something that could, under a different experimental setup, be readily resolved.

That is, prior to the evaluation we could have optimized the inventory of lexical senses for each selected verb, particularly by boosting the inventory of recorded idioms and phrasals (McShane et al., 2015). This would have substantially decreased the error rate and possibly better highlighted the analyzer's ability to manipulate competing constraints, but at the cost of less realistically conveying the current state of our knowledge resources. Another option would have been to leave the lexicon as is, task the analyzer with analyzing all sentences featuring the selected verbs, but have it return disambiguation results only if it was confident in its decision (if exactly one disambiguation decision received a high score). These choices are among many that should be considered in a community-wide discussion of what counts as useful evaluation for knowledge-based natural-language systems, not to mention cognitive systems overall.

10. Final Thoughts

The main theoretical claim of Ontological Semantics is that the process of language understanding can be modeled using a combination of static knowledge resources and processors that implement psychologically-inspired analysis methods. This process yields automatically generated, ontologically-grounded text meaning representations that align with the meaning representations that humans would create using the same ontological metalanguage and the same state of the lexical and ontological knowledge bases. Since the semantic analysis engines rely on inspectable mechanisms, overtly specified knowledge, and an explanatory theoretical approach, we do not expect our results to be limited by any theoretical or practical ceiling; in fact, we have yet to find issues that we believe are intractable given appropriate knowledge modeling.

Zooming out to the overall field of natural language processing, a common motivation for the recent preference for knowledge-lean system building over knowledge-rich approaches has been the desire to bypass the so-called *knowledge acquisition bottleneck* – i.e., the expense of manually acquiring high-quality, machine-tractable knowledge. In our view, the manual acquisition of knowledge is an excellent and necessary investment, particularly when the same knowledge is used to support a broad variety of functionalities in agent systems. We obtained strong supporting evidence for the reusability of acquired knowledge in our work on the Maryland Virtual Patient clinician training system (McShane et al., 2013), in which the same knowledge that permits agents to understand the utterances of human interlocutors also lets them model the plans and goals of those interlocutors, make decisions, and carry out simulated action based on those decisions. Spending time and effort on acquiring such knowledge is, in our opinion, a more prudent use of human resources than annotating texts for subsequent use by machine learning algorithms. In short, developing agent-oriented language processing capabilities involves a very different cost-benefit analysis than treating natural language in isolation. As shown by our system evaluation, manually acquired knowledge can be successfully brought to bear in service of endowing agents with the capability of deep semantic language analysis.

Acknowledgments

This research was supported in part by Grant N00014-09-1-1029 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research. Thanks to Pat Langley and the reviewers for their helpful comments on drafts of this paper.

References

Bailer-Jones, D. (2009). *Scientific models in philosophy of science*. Pittsburgh, PA: University of Pittsburgh Press.

Beale, S. (1996). *Hunter-Gatherer: Applying constraint satisfaction, branch-and-bound and solution synthesis to natural language semantics* (Technical Report MCCS-96-289). Computing Research Lab, New Mexico State University, Las Cruces, NM.

Beale, S., Nirenburg, S., & Mahesh, K. (1995). Semantic analysis in the Mikrokosmos machine translation project. *Proceedings of the Second Symposium on Natural Language Processing*. Bangkok, Thailand.

Clark, P., & Porter, B. (2004). *KM: The Knowledge Machine 2.0: Users manual*. Available at `http://www.cs.utexas.edu/users/mfkb/km.html`.

Cormen, T. H., Leiserson, C. E., Rivest, R.L., & Stein, C. (2001). *Introduction to algorithms* (2nd Edition). Cambridge, MA: MIT Press.

Fass, D. (1997). *Processing metonymy and metaphor*. London: Ablex Publishing Corporation.

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

Hobbs, J. R. (1989). World knowledge and word meaning. In Wilks, Y. (Ed.) *Theoretical issues in natural language processing* (pp. 16-21). Hillsdale, NJ: Lawrence Erlbaum Associates.

Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit (pp. 55-60). *Proceedings of the Fifty-Second Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Stroudsburg, PA: Association for Computational Linguistics.

McShane, M. (2009). Reference resolution challenges for an intelligent agent: The need for knowledge. *IEEE Intelligent Systems*, *24*, 47-58.

McShane, M., Beale, S., & Babkin, P. (2014). Nominal compound interpretation by intelligent agents. *Linguistic Issues in Language Technology*, *10*, 1-36.

McShane, M., Beale, S. & Nirenburg, S. (2010). Reference resolution supporting lexical disambiguation. *Proceedings of the Fourth IEEE International Conference on Semantic Computing*. Pittsburgh, PA.

McShane, M. & Nirenburg, S. (2012). A knowledge representation language for natural language processing, simulation and reasoning. *International Journal of Semantic Computing*, *6*, 3-23.

McShane, M. & Nirenburg, S. (2013). Use of ontology, lexicon and fact repository for reference resolution in Ontological Semantics (pp. 157-185). In A. Oltramari, P. Vossen, L. Qin, & E. Hovy (Eds.), *New trends of research in ontologies and lexical resources*. Berlin, Heidelberg: Springer-Verlag.

McShane, M., Nirenburg, S., & Beale, S. (2005). An NLP lexicon as a largely language independent resource. *Machine Translation*, *19*, 139-173.

McShane, M., Nirenburg, S., & Beale, S. (2008). Resolving paraphrases to support modeling language perception in an intelligent agent. In J. Bos & R. Delmonte (Eds.), *Semantics in Text Processing: STEP 2008 Conference Proceedings*. London: College Publications.

McShane, M., Nirenburg, S., & Beale, S. (2015). The Ontological Semantic treatment of multiword expressions. *Lingvisticae Investigationes*, *38*, 73-110.

McShane, M., Nirenburg, S., Beale, S., Jarrell, B., Fantry, G., & Mallott, D. (2013). Mind-, body- and emotion-reading. *Proceedings of the Annual Meeting of the International Association for Computing and Philosophy*. University of Maryland, College Park, MD: IACAP.

Nirenburg, S., McShane, M., & Beale, S. (2008). A simulated physiological/cognitive "double agent". In J. Beal, P. Bello, N. Cassimatis, M. Coen, & P. Winston (Eds.), *Papers from the AAAI Fall Symposium: Naturally Inspired Cognitive Architectures*. Technical report FS-08-06, Menlo Park, CA: AAAI Press.

Nirenburg, S. & Raskin, V. (2004). *Ontological Semantics*. Cambridge, MA: MIT Press.

Onyshkevych, B. (1997). *An Ontological-Semantic framework for text analysis*. Doctoral dissertation, Center for Machine Translation, Carnegie Mellon University, Pittsburgh, PA.

Schank, R. (1972). Conceptual Dependency: A theory of natural language understanding. *Cognitive Psychology*, *3*, 532-631.

Wilks, Y. 1985. Text structures and knowledge structures. *Quaderni di Semantica, 6*, 335-344.

Wilks, Y. & Fass, D. (1992). The preference semantics family. *Computing and Mathematics with Applications*, *23*, 205-221.

Winston, P. H. (1984). *Artificial intelligence*. Reading, MA: Addison-Wesley.