Contents

Deep Semantic Analysis of Text

James F. Allen[1,2]
Mary Swift[1]
Will de Beaumont[2]

University of Rochester[1]
Institute for Human and Machine Cognition, Pensacola[2]
email: `james@cs.rochester.edu`

Abstract

We describe a graphical logical form as a semantic representation for text understanding. This representation was designed to bridge the gap between highly expressive "deep" representations of logical forms and more shallow semantic encodings such as word senses and semantic relations. We also present an evaluation metric for the representation and report on the current performance on the TRIPS parser on the common task paragraphs.

1 Introduction

As building rich semantic representations of text become more feasible, it is important to develop standard representations of logical form that can be used to share data and compare approaches. In this paper, we describe some general characteristics that such a logical form language should have, then present a graphical representation derived from the LF used in the TRIPS system (Allen et al., 2007).

The Logical Form is a representation that serves as the interface between structural analysis of text (i.e., parsing) and the subsequent use of the information to produce knowledge, whether it be for learning by reading, question answering, or dialogue-based interactive systems.

It's important to distinguish two separable problems, namely the ontology used and the structure of the logical form language (LFL). The ontology determines the set of word senses and semantic relations that can be used. The LFL determines how these elements can be structured to capture the meaning of sentences. We are addressing the latter in the paper. Consider some principles for designing useful LFs.

Preserve Rich Semantic Content in Phrasing

The LFL should allow one to express the dependencies and subtleties that are expressed in the sentence. On the simple end, this means the LFL should allow us to represent the differences between the NP *The pigeon house*, which is a type of house, and *the house pigeon*, which is a type of pigeon. On the more complicated end, the LFL should be able to capture complex quantifier structures such as those in the NPs *Nearly all peaches*, or *Every dog but one*, and phenomena such as modal operators, predicate modifiers, and explicit sets.

One might argue that capturing such complex phenomena in the LFL is premature at this time, as existing techniques are unlikely to be able to produce them reliably. On the other hand, if we don't allow such subtleties in the gold-standard LFL, we will tend to stifle long-term work on the difficult problems since it is not reflected in the score in evaluations.

Encoding ambiguity compactly when possible

This issue has a long history in the literature, with the most classic case being quantifier scoping. Underspecified representations of quantifier scoping are a prime focus in the development of modern logical form languages such as MRS (Copestake et al., 2006), and work goes all the way back to early natural language systems (e.g. Woods, 1978). Other techniques for compactly encoding ambiguity include prepositional phrase attachment, and most critically, the use of vague predicates and relations. For example, for many cases of noun-noun modification, the exact semantic relation between the nouns cannot be determined, and actually need not be determined precisely to be understood.

Enable Viable Partial Interpretations

In many cases, because of limitations in current processing, or because of the fragmentary nature of the language input itself, a system will only be able to construct partial interpretations. The LFL should be constructed in a way such that partial representations are easily compared with full representations. In particular, the interpretation of a fragment should be a subset of the full logical form of the entire sentence. It is a fortunate circumstance that representations that tend to compactly encode ambiguity tend also to have this subset property.

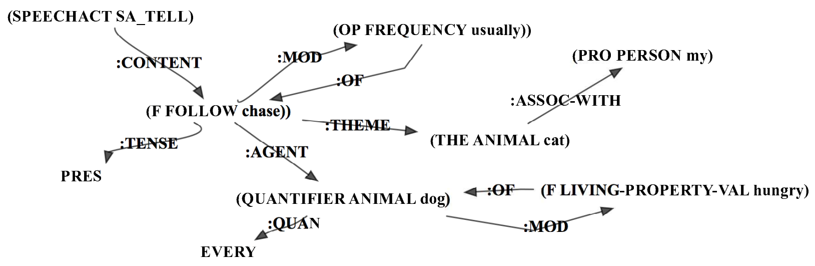

Figure 1: An LF Graph for "Every hungry dog usually chases my cat"

2 Overview of LF Graphs

An example LF-graph is shown in Figure 1. This graph introduces much of the formalism. Each node represents either a speechact, a proposition, a generalized quantifier, an operator or a kind. Nodes are labelled in three parts, the *specifier,* indicating the semantic function of node, the *type,* indicating conceptual class drawn from the ontology, and the *word* from the input. The latter allows us to relate the nodes in the LF graph back to the input. The edges are labelled with semantic roles that indicate argument structure and other critical properties such as modification relationships.

Consider each of the core node types. The first term type captures the meanings of fragments that define eventualities (i.e., events and properties). For instance, the node (F FOLLOW chase) in Figure 1 refers to an eventuality of the type FOLLOW (which would be defined in the ontology). Additional information about the eventuality is captured by the outgoing edges, which identify two arguments, the :*Agent* and the :*Theme*, and one other that provides the tense information for later contextual interpretation (PRES is the present tense).

The second node type captures generalized quantifier constructions. The node (THE ANIMAL cat) indicates a definite description referring to a object of type ANIMAL in the ontology. Generalized quantifiers that have universal import are indicated as shown in the node (QUANTIFIER ANIMAL dog), where an edge labelled :QUAN gives the specific quantifier involved. Note also the presence of a modification to the type (the :MOD) arc, which points to another eventuality, namely (F LIVING-PROPERTY-VAL hungry), which in turn has an argument (:OF) pointing back to the modified node. The :MOD link is critical for capturing dependencies that allow us to reconstruct the full logical form from the graph. For instance, it allows us to retain the distinction between head noun and the modifiers (e.g., *the pigeon house* vs *the house pigeon*).

Table 1 shows the core set of generalized quantifiers used in TRIPS (and subsequently interpreted in discourse processing, especially reference resolution. A large set of quantifiers that indicate the size (e.g., *many, some, five, at most three, a few, ...*) are treated as an indefinite construction with a (often vague) size modifier.

Table 1: Core Generalized Quantifiers

Type	**Description**
THE	a definite form (we expect to be able to resolve it from context)
A	an indefinite form (we expect it to introduce new objects)
PRO	a pronoun form (we expect it to be resolved from local context)
IMPRO	an implicit anaphoric form
BARE	forms with no specifier and ambiguous between generic, kind, and indefinite
QUANTIFIER	“universally” quantified constructions (e.g., EVERY)
QUANTITY-TERM	a quantity expressed in units (e.g., three pounds)
WH-TERM	“wh” terms as in questions (e.g., which trucks)
KIND	the definition of a kind (aka lambda abstraction)

The next term type specifies modal operators, and seen in Figure 1 as the node (OP FREQUENCY usually). The operator nodes must be distinguished from the terms for predications (F) to support algorithms for quantifier and operator scoping.

The final class of node in Figure 1 is the speech act performed by an utterance: (SPEECHACT TELL). This has no third argument as it does not arise from any single word in the utterance. The semantic role :content indicates the propositional content of the speech act, and additional roles indicating the speaker and hearer are suppressed. Speech acts have modifiers in order to handle phenomena such as discourse adverbials.

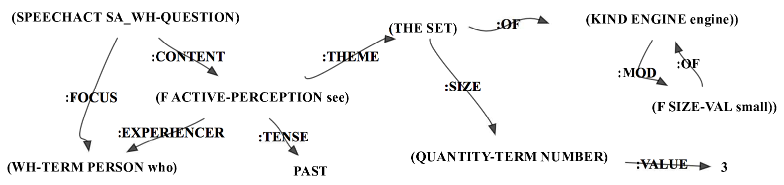

Figure 2: The LF graph for "Who saw the three small engines"

Figure 2 shows another LF graph which captures some additional key constructions. It shows another speech act, for Wh-questions, and shows the handling of plurals. LF graphs distinguish explicitly between singular and plurals by modeling sets, in which an :of argument that points to the type of objects in the set.

The KIND operator is used to define these types (aka lambda abstraction). Thus *the three small engines* is a SET of size three with elements of KIND ENGINE and which are small.

LF-graphs are interesting as they offer the possibility of comparing the semantic content of different approaches, from shallow approaches that identify word senses and semantic roles, to complex representations produced by state-of-the-art deep parsers. On the shallow side, a word sense disambiguation system would produce a set of nodes with the word senses labeled from an ontology, but not indicating a specifier, and not capturing any semantic roles. A system that identifies semantic roles can capture its results using the edges of the graph.

On the other hand, we can show that the LF-graph formalism is equivalent to the TRIPS logical form language (LFL), which is a "flat" scope-underspecified representation of a reference modal logic with generalized quantifiers and lambda abstraction.

We have developed an efficient quantifier scoping algorithm on this LFL that constructs possible fully-scoped forms in the reference logic, and we can prove that we derive the same sets of possible interpretations as the representations constructed by MRS (Manshadi et al., 2008). Figure 3 shows the TRIPS logical form that produced Figure 1, and Figure 4 shows one of the interpretations produced by the scoping algorithm.

```
(SPEECHACT a1 TELL :content f1)
(F f1 (:* FOLLOW Chase)  :agent x :theme y)
(EVERY x (:* ANIMAL Dog) :mod f2)
(F f2 (:* LIVING-PROPERTY-VAL Hungry) :of x)
(A  y   (:* ANIMAL Cat))
(OP p1 (:* FREQUENCY usually) :of f1)
```

Figure 3: TRIPS Logical Form of “Every hungry dog usually chases my cat”

```
Every(x, Dog(x) ^ Hungry(f2) ^ theme(f2,x),
  Frequent(
    A(y, Cat(y),
       Chase(f1) ^ agent(f1,x) ^ theme(f1,y))))
```

Figure 4: One possible scoped interpretation shown in reference representation

Coreference

The final information encoded in the LF graphs is coreference information. Referential expressions are connected to their antecedents using a :coref arc. Note this can only encode referential relations to antecedents that actually appear previously in the text. Simple forms of bridging reference can also be encoded using the insertion of IMPRO nodes that stand in for implicit arguments, and may then co-refer with terms in the graph.

3 The LF Ontology and Word Senses

The LF ontology is the source of the semantic types and semantic roles that are used in the LF graphs. In this paper, we use the LF ontology of the TRIPS system. The TRIPS ontology also defines a rich set of semantic features that are crucial for constraining ambiguity at multiple levels of language processing. For example, the grammar uses selectional restrictions to guide word sense disambiguation and prepositional phrase attachment during parsing, and reference resolution uses the semantic features to identify valid referents and discard invalid ones.

The TRIPS LF ontology is designed to be linguistically motivated and domain independent. The semantic types and selectional restrictions are driven by linguistic considerations rather than requirements from reasoning components in the system (Dzikovska et al., 2003). Word senses are defined based on subcategorization patterns and domain independent selectional restrictions. As much as possible the semantic types in the LF ontology are compatible with types found in FrameNet (Johnson and Fillmore, 2000). FrameNet generally provides a good level of abstraction for applications since the frames are derived from corpus examples and can be reliably distinguished by human annotators. However we use a smaller, more general set of semantic roles for linking the syntactic and semantic arguments rather than FrameNet's extensive set of specialized frame elements. The LF ontology defines approximately 650 semantic types and 30 semantic roles. See Dzikovska et al. (2004) for more discussion of the relationship between FrameNet and the LF ontology. We also expanded our verb coverage by integrating VerbNet entries (Swift, 2005; Benoit Crabbe and Swift, 2006).

The LF ontology also differs from FrameNet in its use of a rich semantic feature set. Our semantic features are an extended version of EuroWordNet (Vossen, 1997). There are five top-level distinctions: physical object, abstract object, situation, time and proposition. Subtypes are defined to capture distinctions in lexical aspect, spatial abstractions (point, region...), origins (natural, artifact...) and so on.

We are not attempting to capture all possible word senses in our ontology. Rather, we are looking for the level of abstraction that affects linguistic processing, and leave finer distinctions for subsequent discourse processing and inference. In order not to lose information in the LF, our word senses are a tuple of form (:* <LF-type> <word-type>), where the LF-type comes from the Ontology, and the <word-type> is a canonicalized version of the word. For example, the property of a switch/device being *on* or *off* is associated with an LF type ARTIFACT-PROPERTY-VAL. Another sense of *on* is its spatial reading, of type SPATIAL-LOC, which also includes words such as *behind* and *in front of.* These two senses of *on* are:

(:* ARTIFACT-PROPERTY-VAL ON)
(:* SPATIAL-LOC ON).

Though we don't have the space to describe it here, TRIPS provides an ontology mapping capability that allows developers to easily map the TRIPS LF forms to a domain-specific ontology (Dzikovska et al., 2008).

4 System Overview

Much recent text processing work has focused on developing "shallow", statistically driven, techniques. We have taken a different approach. We use statistical methods as a preprocessing step to provide guidance to a deep parsing system that uses a detailed, hand-built, grammar of English with a rich set of semantic restrictions. This way, we hope to obtain deeper, more accurate interpretations. Because the parser was developed to identify likely fragments when an entire interpretation cannot be constructed, we believe it can match statistical methods in its precision and recall measures.

The TRIPS grammar is lexicalized context-free grammar, augmented with feature structures and feature unification. The grammar is motivated from X-bar theory, and draws on principles from GPSG (e.g., head and foot features) and HPSG. The search in the parser is controlled by a set of hand-build rule preferences encoded as weights on the rules, together with a heavy use of selectional restrictions (encoded in the lexicon and ontology) to eliminate semantically anomalous sense combinations.

The TRIPS parser uses a packed-forest chart representation and builds constituents bottom-up using a best-first search strategy similar to A*, based on rule and lexical weights and the influences of the techniques addressed below. The search terminates when a pre-specified number of spanning constituents have been found or a pre-specified maximum chart size is reached. The chart is then searched using a dynamic programming algorithm to find the least cost sequence of constituents according to a cost table that can be varied by genre. For instance, when processing text as in the experiments reported here, we mostly expect UTT constituents encoding the speech acts TELL, then less likely the speech acts WH-QUESTION and YN-QUESTION and we don't expect dialog-based speech acts such as CONFIRM or GREET. In addition, we also assign costs to non-sentential constituents (e.g., NPs, ADVPs, etc). The resulting least cost sequence produces a set of logical forms that are the results reported here.

Here we describe the different ways that shallow methods contribute to deep processing.

Using Preprocessors

First, statistical processing is used as a preprocessor. The TRIPS parser accepts a word lattice as input, which we have used when working with speech recognition where we want to consider multiple word hypotheses simultaneously. We have used this capability to allow for preprocessors as well. For instance, we use multiple named entity recognizers (NER) to identify names of people, companies, geographical locations, and so on. The output of the NERs are treated as additional constituent hypotheses in the input to the parser. As an example, consider the sentence *The New York Times is a newspaper.* Assuming an NER identifies *The New York Times* as a name with semantic type PUBLICATION, the input to the parser will be:

```
(word "the" 1 2)
(word "new" 2 3)
(word "york" 3 4)
(word "times" 4 5)
```

```
(constit "the new york times" 1 5
         :syn (NAME :class PUBLICATION))
(word "is" 5 6)
(word "a" 6 7)
(word "newspaper" 7 8)
```

As the parser runs, it chooses between interpreting the words individually or using the name, depending on what produces the best overall interpretation. In addition, we use a specialized recognizer that identifies possible street addresses (e.g., 15 N 25th St NE). Note we don't need specialized NERs for dates and times as they are handled in the main grammar.

Part of Speech Tagging

We also use a part-of-speech tagger to preprocess the input and provide a likely POS tag (or set of tags) for consideration by the parser. Rather than eliminating the interpretations that do not match, the parser simply assigns more weight to interpretations consistent with the tags. This allows the parser to override bad POS assignments in some cases.

Using on-line resources

We have built a system called WordFinder that draws on WordNet (Miller, 1995) and COMLEX (Grishman et al., 1994) to construct (underspecified) lexical representations using mapping rules from high-level WordNet classes into our LF ontology. We deliberately stay at a fairly abstract level as we would rather have a few semantically abstract lexical entries rather than the many highly-specific senses found in WordNet, which we have not found useful for parsing.

Using Preferences during Parsing

Preferences (either syntactic or semantic) can be given to the parser based on statistical or other analyses. We have used the Collins parser as a preprocessor to extract hypotheses for the three constituents (NP, VP, and ADVP) which in pretests had a precision greater than 60% (Swift et al., 2004). For instance, the sentence *The New York Times is a newspaper,* the Collins preprocessor would produce the following preferences:

(NP 1 5) (NP 6 8) (VP 5 8) (S 1 8)

With simple sentences, this information has little effect. But on longer complex sentences, we found that the preferences allow us to produce more accurate interpretations in faster time. Note again that the parser is not required to follow this advice – all this information does is add a preference for such interpretations.

Another mechanism we use is logical form preference patterns. Local form patterns of predicate types and arguments can be specified with preference scores. Consider the sentence "He put the box in the corner near the house". Although the location adverbial "near the house" could possibly describe the putting event, it is much more likely that it modifies the corner. Thus the pattern (PUT :agent :theme :destination) is preferred over the pattern (PUT :agent :theme :destination :location). We have only tested this capability so far with hand-specified patterns, though we plan to start experiments using learned patterns derived from propositions extracted from corpora (e.g. van Durme et al., 2008). The overall system, using all these techniques, is shown graphically in Figure 5.

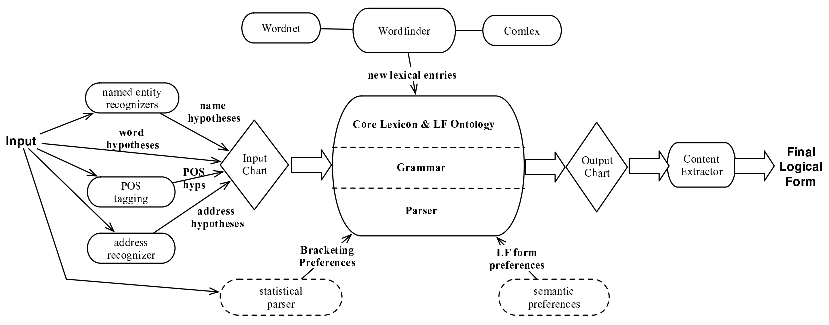

Figure 5: Using Shallow Methods to Inform Deep Parsing (the subsystems in dotted ovals were not used in the reported evaluations

5 An Evaluation Metric for LF Graphs

In this section we define an evaluation metric for LF-graphs that allows us to quantify our system performance against gold standard representations.

The evaluation metric between a gold LF graph *G* and a test LF graph *T* is defined as the maximum score produced by any node/edge alignment from the gold to the test LF. More formally, an alignment *A* is a 1-1 mapping from the nodes of the gold graph to nodes of the test graph (or to a pseudo empty node if there is no corresponding node in the test graph). Once we have defined a scoring metric between aligned nodes and edges, we define the match between a gold and test graph as the maximum score produced by an alignment. While complex scoring functions can be used, our results reported here use a simple measure:

$\text{Nscore}_A(\text{n})$ = 2 if both the indicator and word in the label of n matches the label of A(n), 1 if one of them matches, and 0 otherwise.

Escore$_A$(e) = 1 if e connect nodes n1 and n2, and there is an edge between A(n1) and A(n2) with same label, 0 otherwise.

Gscore(G,T) = max$_A$(Sum$_{n,ein}$(Nscore$_A$(n)+Escore$_A$(e))

Once we know Gscore(G,T), we can compute semantic precision and recall measures by comparing this to the G and T graphs aligned with themselves, which gives us the maximum possible gold and test scores.

Precision(G,T) = Gscore(G,T)/Gscore(T,T)
Recall(G,T) = Gscore(G,T)/Gscore(G,G)

A more general function of node matching would be more informative. For instance, with words not in our core lexicon, we usually derive an abstract sense that is not the most specific sense in our ontology, however is an abstraction of the correct sense. A scoring function that would give such cases partial credit would have raised our scores (cf. Resnik and Yarowsky, 1997).

Evaluation Procedure and Results

To evaluate our system on the shared texts, we built gold representations for each. We did this by first generating an LF-graph by running the system, and then correcting this by hand using a graphical editor. Appendix A gives the gold standard for a sample paragraph.

Table 2 reports the results on our baseline system, which was the first run we made on the shared texts once they became available. In addition, we report results on the latest version of the system after making some lexicon and grammar additions based on problems found in parsing the paragraphs.

Specifically, we added 16 new lexical items (1 verb, 12 nouns, 2 adjectives and 1 adverb); 17 new or modified senses for existing lexical items; 3 new ontology concepts and one grammar rule, to handle the formulation of meters per second as “m/s”.

6 Conclusion

We have described a graphical logical form language for expressing a significant amount of the semantic content in natural text. The representation allows for the specification of partial information extracted from sentences, yet is expressive enough to capture many of the subtleties and complications present in linguistically motivated approaches, including supporting complex processes such as quantifier scoping, reference resolution, and reasoning.

Table 2: Evaluation Results

	Base System		**Final System**	
Text	**Prec**	**Recall**	**Prec**	**Recall**
1 "physics"	70.1%	70.1%	73.4%	80.0%
2 "cancer"	62.1%	71.9%	71.9%	79.3%
3 "dining"	86.7%	90.4%	90.8%	94.6%
4 "dogs"	63.0%	68.6%	63.8%	69.1%
5 "guns"	55.0%	64.0%	63.8%	73.4%
6 "gardens"	47.4%	53.6%	59.7%	62.0%
7 "wind"	n/a	n/a	65.8%	76.3%
Average	64.1%	69.7%	69.9%	76.4%

We also briefly described a hybrid system architecture centered around a domain-general, broad coverage parsing framework capable of producing deep analyses of texts. Statistical and corpora-based approaches serve to inform the parsing in order to achieve a balance between depth of analysis and broad coverage.

We find the results very encouraging, given this is our first evaluation of the system on text rather than dialog. While it is hard to quantify exactly without further detailed analysis, the remaining errors probably break down roughly evenly between gaps in grammatical coverage, word sense disambiguation errors and inadequacies in our search. Looking at grammatical coverage, the single biggest problem appears to be conjoined sentences with subject ellipsis. Regarding our search problems, because we are building semantic structures rather than syntactic, the search space is much bigger than a traditional CFG. We believe that integrating a statistical parser preprocessor and the LF-preference mechanism will start to address this problem.

Acknowledgments This work was supported by grants from the National Science Foundation (#0748942), the Defense Advanced Research Projects Agency (FA8750-07-D-0185), and the Of[fb01]ce of Naval Research (N000140510314).

References

Allen, J., M. Dzikovska, M. Manshadi, and M. Swift (2007, June). Deep linguistic processing for spoken dialogue systems. In *ACL 2007 Workshop on Deep Linguistic Processing*, pp. 49–56. Association for Computational Linguistics.

Benoit Crabbe, Myroslava Dzikovska, W. d. B. and M. Swift (2006, June).

Extending the coverage of a domain independent dialog lexicon with VerbNet. In *Proceedings of the Third International Workshop on Scalable Natural Language Understanding (ScaNLU06) at HLT-NAACL 2006*, pp. 25–32.

Copestake, A., D. Flickinger, C. Pollard, and I. Sag (2006). Minimal recursion semantics: An introduction. *Research on Language and Computation 3*(4), 281–332.

Dzikovska, M., J. Allen, and M. Swift (2008). Linking semantic and knowledge representations in a multi-domain dialogue system. *J. Log. and Comput. 18*(3), 405–430.

Dzikovska, M., M. Swift, and J. Allen (2003, August). Integrating linguistic and domain knowledge for spoken dialogue systems in multiple domains. In *Proceedings of Workshop on Knowledge and Reasoning in Practical Dialogue Systems at the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*, Acapulco, Mexico, pp. 383–389.

Dzikovska, M., M. Swift, and J. Allen (2004, May). Building a computational lexicon and ontology with FrameNet. In *Proceedings of Workshop on Building Lexical Resources with Semantically Annotated Corpora at The 4th International Conference on Language Resources and Evaluation (LREC'04)*, pp. 53–60.

Grishman, R., C. Macleod, and A. Meyers (1994). Comlex syntax: Building a computational lexicon. In *COLING*, pp. 268–272.

Johnson, C. and C. J. Fillmore (2000). The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, San Francisco, CA, USA, pp. 56–62. Morgan Kaufmann Publishers Inc.

Manshadi, M., J. Allen, and M. Swift (2008, August). Toward a universal underspecifed semantic representation. In *13th Conference on Formal Grammar (FG 2008).*

Miller, G. A. (1995). Wordnet: a lexical database for english. *Commun. ACM 38*(11), 39–41.

Resnik, P. and D. Yarowsky (1997, June). A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of SIGLEX '97*, pp. 79–86.

Swift, M. (2005, February). Towards automatic verb acquisition from VerbNet for spoken dialog processing. In K. Erk, A. Melinger, and S. S. im Walde (Eds.), *Proceedings of Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*, pp. 115–120.

Swift, M., J. Allen, and D. Gildea (2004, Aug 23–Aug 27). Skeletons in the parser: Using a shallow parser to improve deep parsing. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA, pp. 383–389. Association for Computational Linguistics.

van Durme, B., T. Qian, and L. K. Schubert (2008, Aug 18–Aug 22). Class-driven attribute extraction. In *COLING '08: Proceedings of the 24th international conference on Computational Linguistics*. Association for Computational Linguistics.

Vossen, P. (1997, March 5-7). EuroWordNet: a multilingual database for information retrieval. In *Proceedings of the DELOS workshop on Cross-language Information Retrieval*, Zurich.

Woods, W. A. (1978). Semantics and quantification in natural language question answering. *Advances in Computers 17*, 1–87.

Appendix A: Sample Analyses (Hand Built Gold Standards)

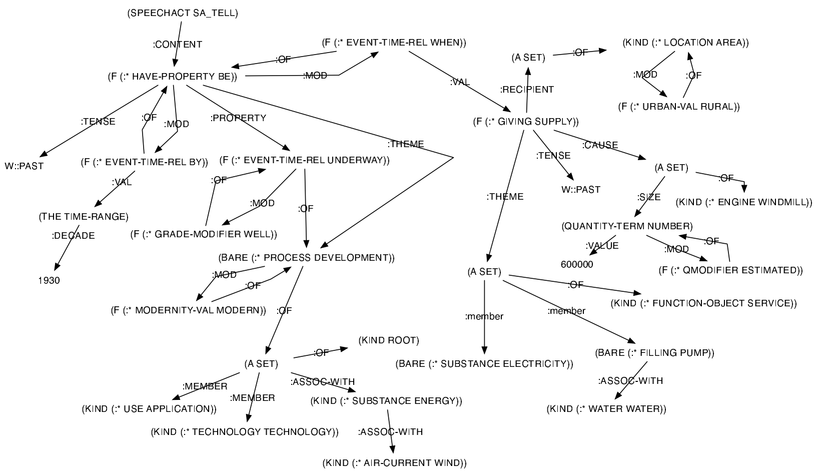
(SPEECHACT SA_TELL)
:CONTENT
(F (:* HAVE-PROPERTY BE))
(F (:* EVENT-TIME-REL WHEN))
:OF
:MOD
:VAL
(A SET)
:OF
(KIND (:* LOCATION AREA))
:MOD
:OF
(F (:* URBAN-VAL RURAL))
:RECIPIENT
(F (:* GIVING SUPPLY))
:TENSE
:OF
:MOD
:PROPERTY
:THEME
W::PAST
(F (:* EVENT-TIME-REL BY))
(F (:* EVENT-TIME-REL UNDERWAY))
:VAL
(THE TIME-RANGE)
:DECADE
1930
(F (:* GRADE-MODIFIER WELL))
(BARE (:* PROCESS DEVELOPMENT))
(F (:* MODERNITY-VAL MODERN))
(A SET)
(KIND ROOT)
:MEMBER
:ASSOC-WITH
(KIND (:* USE APPLICATION))
(KIND (:* TECHNOLOGY TECHNOLOGY))
(KIND (:* SUBSTANCE ENERGY))
(KIND (:* AIR-CURRENT WIND))
:CAUSE
(A SET)
:SIZE
(KIND (:* ENGINE WINDMILL))
(QUANTITY-TERM NUMBER)
:VALUE
600000
(F (:* QMODIFIER ESTIMATED))
:member
(KIND (:* FUNCTION-OBJECT SERVICE))
(BARE (:* SUBSTANCE ELECTRICITY))
(BARE (:* FILLING PUMP))
(KIND (:* WATER WATER))

Figure 6: Modern development of wind-energy technology and applications was well underway by the 1930s, when an estimated 600,000 windmills supplied rural areas with electricity and water-pumping services.

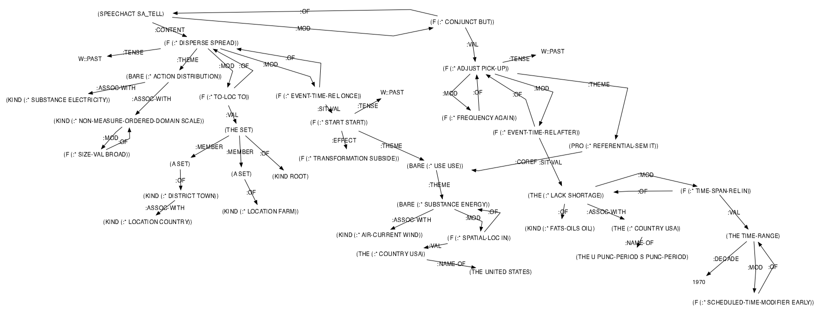
(SPEECHACT SA_TELL)
:CONTENT
(F (:* DISPERSE SPREAD))
(F (:* CONJUNCT BUT))
W::PAST
(BARE (:* ACTION DISTRIBUTION))
(KIND (:* SUBSTANCE ELECTRICITY))
(KIND (:* NON-MEASURE-ORDERED-DOMAIN SCALE))
(F (:* SIZE-VAL BROAD))
(F (:* TO-LOC TO))
(THE SET)
(A SET)
(KIND (:* DISTRICT TOWN))
(KIND (:* LOCATION COUNTRY))
(KIND (:* LOCATION FARM))
(KIND ROOT)
(F (:* EVENT-TIME-REL ONCE))
(F (:* START START))
(F (:* TRANSFORMATION SUBSIDE))
(BARE (:* USE USE))
(BARE (:* SUBSTANCE ENERGY))
(KIND (:* AIR-CURRENT WIND))
(F (:* SPATIAL-LOC IN))
(THE (:* COUNTRY USA))
(THE UNITED STATES)
(F (:* ADJUST PICK-UP))
(F (:* FREQUENCY AGAIN))
(F (:* EVENT-TIME-REL AFTER))
(PRO (:* REFERENTIAL-SEM IT))
(THE (:* LACK SHORTAGE))
(F (:* TIME-SPAN-REL IN))
(KIND (:* FATS-OILS OIL))
(THE U PUNC-PERIOD S PUNC-PERIOD)
(THE TIME-RANGE)
1970
(F (:* SCHEDULED-TIME-MODIFIER EARLY))

Figure 7: Once broad-scale electricity distribution spread to farms and country towns, use of wind energy in the United States started to subside, but it picked up again after the U.S. oil shortage in the early 1970s.