The AAAI 2017 Spring Symposium on
Computational Construction Grammar and Natural Language Understanding
Technical Report SS-17-02

Broad Coverage, Domain-Generic Deep Semantic Parsing

James F. Allen[1,2] and Choh Man Teng[2]
[1]Department of Computer Science, University of Rochester, Rochester, NY14627
[2]Institute for Human and Machine Cognition, 40 South Alcaniz Street, Pensacola, FL 32502
jallen@ihmc.us, cmteng@ihmc.us

Abstract

The TRIPS parser is a broad-coverage domain-general deep semantic parser that produces logical forms grounded in a general ontology. While using many of the techniques of modern syntactic theory, the system is semantically driven and uses many of the ideas of construction grammar. Unlike most work in semantic parsing, which is limited to specific simple domains, the TRIPS parser performs well in many diverse domains, after incorporating domain-specific named entity recognition where needed. The TRIPS grammar uses syntactic, semantic and ontological constraints simultaneously to construct semantically accurate parses, and includes many rules that capture the common constructions of everyday spoken language.

Introduction and Motivation

We describe work on developing a broad-coverage, deep semantic parser (TRIPS). The system combines rich grammatical, lexical and ontological information to produce semantic representations expressed in a general ontology. The grammatical rules include ones motivated purely from syntax, constructions restricted to certain specific semantic types, and rules that encode constructions of conversational speech. By exploiting mappings from WordNet (Fellbaum, 1998) we can handle over 100,000 words, plus additional domain-specific named entities produced by named entity recognition techniques using domain-specific resources.

Recent work on learning semantic parsers (e.g., Matuszek et al., 2012; Tellex et al., 2013; Branavan et al., 2010; Chen and Mooney, 2011) is highly domain dependent and useful only for that domain. One cannot, for instance, reuse a system learned in one robotic domain in another robotic domain, let alone in a different domain such as database query. In addition, these techniques work only in limited tasks/domains. In contrast, we are developing a broad-coverage, domain-general semantic parser. It operates reasonably well in many different domains and genres including: human dialogue with an embodied system discussing and performing simple tasks in a blocks world, system-biologist dialogue involving construction and exploration of causal chains of protein reactions (e.g., cancer pathways), texting with teens to help them manage their asthma, understanding simple commonsense stories, and learning tasks from combined verbal instruction and demonstration (e.g., learning to navigate the web and perform tasks in a web browser). Each of these systems uses the exact same grammar, lexicon and ontology, but may have its own specialized named entity recognizers (e.g., protein names in biology, human names in the story domain). The only other difference between applications is the search parameters that optimize the parsing for each genre (e.g., texting, dialogue, scientific papers, dictionary definitions).

Using a scoring metric for matching logical forms described in Allen et al. (2008), which is roughly equivalent to the more recently developed SMatch score (Cai and Knight, 2013), the parser currently achieves 70-80% semantic accuracy on dialogue and moderately complex text (e.g., Wikipedia articles). On highly complex text, such as biology scientific papers, although the system may not always produce a complete parse, often it is able to identify semantically coherent fragments, and the system has performed well in event extraction tasks (Allen et al., 2015). The TRIPS parser tuned to a few different domains is available online.[1] These sites also provide an API for parsing sentences using the service.

[1] Try one of the parsers online, tuned for: discussions about biology (http://trips.ihmc.us/parser/cgi/bob), discussions involving planning, learning and acting in a physical blocks world (http://trips.ihmc.us/parser/cgi/cabot), and general parsing of simple commonsense stories (http://trips.ihmc.us/parser/cgi/step).

Overview of the Approach

There are three distinct sources of knowledge that are combined to guide the parsing in the TRIPS system: The ontology, the lexicon with linking rules, and the grammar.[2] While we discuss each of these sources separately, it is key to remember that during parsing, the system uses all the sources of knowledge simultaneously in order to arrive at the best pragmatic/semantic interpretation.

The Ontology

The TRIPS ontology aims to combine the correspondences to the semantics that underlies linguistic structure (cf. VerbNet (Kipper et al., 2008) for verbs) and commonsense taxonomies of the objects and events in the world (cf. FrameNet (Baker et al., 1998)). For instance, VerbNet indicates that *push, shove, yank* and *pull* should be in the same class as they behave similarly in language. FrameNet also classifies these words together in their MANIPULATION frame, but also includes verbs such as *grasp, grab, kiss* and *touch.* VerbNet clusters *touch, kiss* and *grasp* in a different class.

In addition, neither VerbNet nor FrameNet attempts to place their concepts within a more general ontology that underlies reasoning systems (cf., SUMO (Niles and Pease, 2001), Dolce (Gangemi et al., 2003)). Putting all these constraints together, Figure 1 shows a fragment of the TRIPS ontology that covers the types discussed above. Note the abstraction hierarchy relates concepts that are similar in structure but have important differences in semantics (e.g., ONT::PULL and ONT::PUSH are distinct but both events of ONT::APPLY-FORCE).

One result of taking linguistic semantics seriously is that the ontology also provides a clean organization of semantic roles, given that the roles are defined using clear semantic principles. The TRIPS semantic roles are defined in a way that cleanly identifies classes in the ontology. In fact, the semantic roles that a word in an ontology type can take is a key factor in deciding where the type might fit in the ontology. Conversely, an ontology type that is associated with words with widely varying semantic roles is an indication that the type probably needs to be refactored.

The TRIPS ontology makes a distinction between *core roles*, those that are realized as direct arguments to the verb, and *relational roles*, which identify relations between an event and a proposition and are typically produced by grammatical constructions (cf., Goldberg, 1995) rather than being in lexical item definitions (see discussion later). Table 1 shows the five most important core argument roles with an informal gloss defining them.

[2] Browse the lexicon and ontology online at www.cs.rochester.edu/research/trips/lexicon/browse-ont-lex.html

- **EVENT-OF-CHANGE**
  - **EVENT-OF-ACTION [AGENT]**
    - **EVENT-OF-AGENT-INTERACTION [AGENT, AGENT1]**
      - **COMMUNICATION [AGENT, AGENT1, FORMAL]**
    - **EVENT-OF-CAUSATION [AGENT, AFFECTED]**
      - **TOUCH [AGENT, AFFECTED] "touch"**
        - **APPLY-FORCE [AGENT, AFFECTED]**
          - **PUSH [AGENT, AFFECTED] "push" "shove"**
          - **PULL [AGENT, AFFECTED] "pull" "yank"**
      - **BODY-MANIPULATION [AGENT, AFFECTED] "grasp" "grab"**
      - **KISSING [AGENT, AFFECTED] "kiss"**
  - **EVENT-OF-UNDERGOING-ACTION [AFFECTED]**
    - **BECOME [AFFECTED, FORMAL] "become"**
- **EVENT-OF-STATE [NEUTRAL]**
  - **EVENT-OF-EXPERIENCE [NEUTRAL, EXPERIENCER]**
    - **PERCEPTION [NEUTRAL, EXPERIENCER] "feel"**
  - **HAVE-PROPERTY [NEUTRAL, FORMAL] "be"**

Figure 1. A Fragment of the TRIPS Ontology

Because most lexical resources that include roles (e.g., VerbNet and Propbank (Palmer et al., 2005)) are organized in classes that are not encoded in an ontology, there is little constraint on what roles a class of verbs might take. The TRIPS roles, on the other hand, must be consistent with inheritance down the TRIPS ontology hierarchy, similar to properties in a semantic network representation.

Figure 1 shows the top of the event hierarchy and the roles that are defined and inherited down the hierarchy. Note that at the most abstract level, event types are distinguishable by the semantic roles the events can take.

ROLE	Key Properties
AGENT	entity causes/instigates the event
AFFECTED	entity changed by the event
NEUTRAL	entity has temporal existence, but neither causes nor is changed by the event
FORMAL	entity is a formal construct (e.g., proposition, action type)
EXPERIENCER	cognitive entity engaged in cognition or perception

Table 1. Some Key Core Semantic Roles

Note also the ontology is critical for enabling semantically-constrained constructions in language. For instance, consider the noun phrase *The melting temperature of ice*. This construction involves a discontinuous constituent as *of ice* is an argument to the event *melting*. This construction cannot be used for all nouns, however. Consider that *The melting dog of ice* cannot be interpreted in the same way. This construction is limited to nouns that are subclasses of the TRIPS class ONT::DOMAIN, which includes nouns that identify scales, including *temperature, rate* and *weight*.

The ontology also assigns to each class a set of semantic features that are used as selectional preferences during parsing. The feature vectors are an efficiency device that allows us to work with a single inheritance hierarchy, with a constant-time subsumption check operation during parsing. For instance, the semantic feature value COMESTIBLE identifies edible objects. Types in the ontology with this feature value include ONT::FOOD, ONT::ALCOHOL, ONT::MEDICATION, ONT::FISH and ONT::WATER.

These features and semantic restrictions are used in defining the ontology types. For example, ONT::CONSUME, a type containing verbs such as *eat, drink, ingest* and *nibble*, is defined as follows.

```
ONT::CONSUME
  :PARENT ONT::EVENT-OF-CAUSATION
  :ROLES
    ( :AGENT  [type ONT::ORGANISM] [origin LIVING] )
    ( :AFFECTED
        [type ONT::PHYS-OBJECT]
        [mobility MOVABLE]
        [form SUBSTANCE]
        [object-function COMESTIBLE] )
```

This says that events of type ONT::CONSUME typically have living organisms as the AGENT and a comestible substance as the AFFECTED (the entity being consumed). Note that these semantic restrictions are only preferences. As described in the section on parsing, these restrictions are used to guide parsing to the best interpretation, not to eliminate interpretations that violate the restrictions. Thus the classic metaphor *my car drinks gas* can still parse.[3]

The Logical Form (LF)

Before discussing the system further, it may be helpful to review the target output of parsing, namely the *logical form* (LF). The logical form consists of a set of related terms drawn from the ontology that describe events, relations and object types, where the concepts are linked by

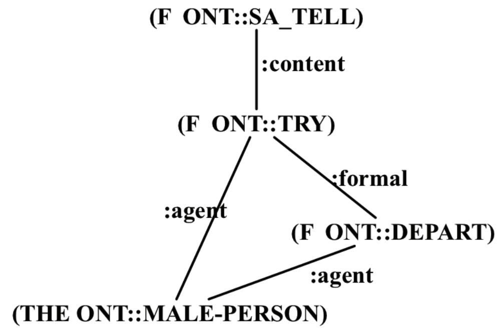

Figure 2: LF graph for "he tried to leave"

semantic roles. The root of the LF is always a speech act. There are several equivalent notations for expressing the LF. In the *term form*, the output is a series of terms, each one denoting an event/proposition or unscoped quantified expression. For instance, the logical form of the sentence *The man tried to leave* is captured by the following LF terms (simplified), starting with a term that encodes the surface speech act:

```
(SPEECHACT s1 ONT::SA_TELL :content w1)
(F w1 ONT::TRY :agent m1 :formal d1)
(F d1 ONT::DEPART :agent m1)
(THE m1 ONT::MALE-PERSON)
```

Each term consists of a specifier that indicates the type of the term, where F denotes a proposition/relation/event, and THE denotes a definite description. Other common specifiers include indefinite expressions (A), plurals (THE-SET, INDEF-SET) and various generalized quantifiers. The second element of a term (e.g., d1) is a new constant (aka a Skolem constant) that is used to refer to this term in other LF terms. It also serves as a discourse entity available in subsequent discourse processing. The third element of a term is the ontology type (e.g., ONT::DEPART). This is followed by a set of semantic role/value pairs (e.g., :agent m1). This LF representation falls into the category of constraint-based underspecified-scope representations whose properties are described in detail in Manshadi et al. (2008).

An alternative representation of the logical form is the *LF graph* (Allen et al., 2008). The LF graph of the same sentence *The man tried to leave* is shown in Figure 2. Each term is represented as a node labelled with the specifier and ontology type, and the roles are represented as arcs. One might note a strong similarity between this graphical representation and the recently developed representation AMR (Banarescu et al., 2013). At a structural level these two representations are very similar, but the TRIPS LF is significantly more expressive semantically because it is derived from the ontology. For instance, in AMR the types are PropBank word senses, which serve to distinguish

[3] Note this produces a literal interpretation of the sentence. Metaphorical reasoning about what this actually means is taken as a post-parsing issue.

senses within an individual verb but do not link identical or similar senses across verbs. By using the ontology as our word senses, words with the same or similar senses are treated uniformly. Furthermore, AMR for the most part only assigns senses to verbs, whereas we assign senses to all content words. Another key difference is that AMR does not include quantifier information, so it cannot distinguish between the noun phrases *the boy, a boy, the boys, boys* and *some boys,* which of course is critical information for subsequent discourse processing.[4]

The Lexicon

The lexicon provides the linking between the syntax-based models produced by the grammar and the ontology. The grammar is a context free grammar augmented with features that are combined using unification. The parser uses the grammar to compute the grammatical relations in the sentence, i.e., logical subject, object, indirect object and complement arguments to all verbs. The lexical entries specify how to map these grammatical roles to the semantic roles associated with a verb sense in the ontology. As an example, here is a lexical entry for the word *eat*:

```
W::EAT
   :PARENT   ONT::CONSUME
   :TEMPL   AGENT-AFFECTED-XP-TEMPLATE
```

The template indicates how the grammatical roles map to the semantic roles and is defined as follows:

```
AGENT-AFFECTED-XP-TEMPLATE
   LSUBJ  (% NP)   AGENT
   LOBJ (% NP)   AFFECTED
```

This says that the logical subject should be a noun phrase and maps to the AGENT role, and the logical object is also an NP and maps to the AFFECTED role. In conjunction with the definition of ONT::CONSUME above, the system can determine that a logical subject that is an NP with semantic type ONT::ORGANISM can map to the AGENT role of ONT::CONSUME, and a logical object NP that is a comestible physical object can map to the AFFECTED role.

Templates can also include syntactic feature information, and capture more complex relationships between the roles. For instance, the verb *try* has the following template (simplified), among others:

```
AGENT-FORMAL-SUBJCONTROL-TEMPL
   LSUBJ  (% NP (var ?v))   AGENT
   LCOMP (% CP (LSUBJ (% NP (var ?v))))  FORMAL
```

This template says that the logical subject, which has the identifier ?v, is the AGENT, and the complement is a CP construction with the subject set to the same identifier ?v. Thus this template makes explicit the implicit argument of the complement in a sentence such as *The man tried to leave,* linked through the lexical entries for the words *try* and *leave*:

```
W::TRY
   :PARENT   ONT::TRY
   :TEMPL   AGENT-FORMAL-SUBJCONTROL-TEMPL
W::LEAVE
   :PARENT   ONT::DEPART
   :TEMPL   AGENT-TEMPL
```

As shown in Figure 2, the AGENT of the ONT::DEPART event is mapped to the same entity as the AGENT of the ONT::TRY event.

The Grammar

The TRIPS grammar is a hand-built extensive grammar of English in the style of X-bar theory (Jackendoff, 1977), consisting of approximately 550 rules covering most of the common constructions of English. It is lexically-driven in that the subcategorization information for all words is encoded in the lexicon and the grammar uses "meta" rules along the lines of "a verb phrase is a verb followed by its subconstituents". It uses head and foot features, and GAP feature propagation from GPSG (Gazdar et al., 1985) to handle discontinuous phenomena needed to account for constructions such as questions and relative clauses.

As a highly simplified example, one of the rules for constructing noun phases says an NP can be a specifier (which might be a simple article such as *the*, a possessive construction such as *John's*, or a complex quantified expression such as *every other),* followed by an Nbar constituent (as in Xbar theory), as long as they agree in number and person. TRIPS encodes person/number as a single feature AGR, so the rule is as follows (where N1 is Nbar):

```
NP[AGR ?agr] <— SPEC[AGR ?agr]  N1[AGR ?agr]
```

For this rule to succeed, the AGR features of the SPEC and N1 constituents must unify to instantiate the AGR feature of the new constructed NP. The AGR of the NP could later be used to enforce subject/verb agreement, as shown below.

So far this describes a fairly standard treatment of syntax. However, the grammar also integrates semantic information from the lexicon by importing the semantic preferences on arguments, as well as constructing the logical forms of constituents on the fly. The selectional preferences are a set of feature values that are stored in a feature called SEM. For example, here is a simple declarative sentence rule that enforces number/person agreement (AGR) as well as unifying the semantic

[4] The complete specification of the logical form can be found at http://trips.ihmc.us/parser/LF%20Documentation.pdf

preferences of the subject of the VP constituent with those of the NP constituent proposed to fill this role in the sentence:

```
S <— NP[AGR ?agr SEM ?SEM]
      VP[AGR ?agr [SUBJ [SEM ?sem]]]
```

This says a sentence can be an NP followed by a VP that selects for a subject that is semantically compatible with the NP. As discussed below, semantic restrictions are not absolute. If they are violated the S constituent can still be built but is dispreferred compared to constituents without any semantic violations.

The TRIPS Parsing System

The TRIPS system is a packed-forest chart parser which builds constituents bottom-up using a best-first search strategy (Allen et al., 2008). In highly simplified terms, every application of a rule and use of a lexical item has a cost and the parser searches for the minimum cost solution. In general, grammatical rules all have the same cost, although in some cases there are minimal differences to encode a slight preference ordering. Likewise, most lexical entries (each identifying a single sense of a word) also have the same cost, with a few exceptions when we encode a priori preferences. We also have customization preferences for lexical entries based on their derivations. These preferences can be adjusted to account for domain-specific language style and usage.

The primary drivers of word sense disambiguation are subcategorization constraints and semantic preferences on arguments from the ontology. The subconstituents often identify good senses. For instance, the verb *have* has two senses among others: ONT::CONSUME and ONT::MAKE-IT-SO. The former is used in *We had a pie* and the latter *Have him open the door.* The latter sentence is not possible with the ONT::CONSUME sense of *have,* as this sense does not subcategorize for this construction. Likewise, the first sentence is not possible with the ONT::MAKE-IT-SO sense of *have*. The system constrains the word senses to those that have templates that match the syntax. When unifying SEM structures, rather than returning a binary yes/no decision, we compute their semantic distance based on the number of semantic features that match or disagree. The more semantic disagreement, the higher the cost of the constituent to be built.

We may also have preferences for certain senses based on the parts of the ontology that are relevant to the current application domain. In many cases, the logical form produced by parsing is transformed into a new representation based on the domain ontology using an ontology mapping system. In systems where such mapping rules are defined, we can automatically compute which TRIPS ontology types correspond to domain-specific types and prefer these senses.

The system can also use a variety of statistical preprocessors to improve accuracy. These include the Stanford tools for part-of-speech tagging, named entity recognition and syntactic parsing. The preprocessors attach into the input stream additional information that acts as "advice" to the parser. As an example, in domains involving complex text, we find the Stanford CoreNLP parser (Manning et al., 2014) can help guide TRIPS to better interpretations. The statistical parser is not used for parsing per se, but we extract from its output the major phrase boundaries (S, NP, ADJP and ADVP). Any constituent constructed by the TRIPS parser incurs an additional cost if it is inconsistent with the Stanford phrase boundaries. In dialogue domains, however, the Stanford parser is generally a liability and not used.

Attaining Broad Lexical Coverage

The TRIPS hand-built lexicon contains about 9000 lexical lemmas (with morphology and multiple senses this translates to many more lexical entries). While this is a fair number of words, it in no way has adequate coverage for parsing arbitrary English. In this section we discuss techniques that allow us to build deep semantic representation for sentences even though they contain words not in the lexicon.

One of the most useful techniques capitalizes on the richness of WordNet. WordNet contains over 100,000 words and multi-words, excluding proper names. The noun and verb word senses (i.e., their synsets) are organized hierarchically using hypernym (subclass in ontology-speak) relations. But the WordNet hypernym hierarchy is sparse in places and there is no upper ontology for verbs. Furthermore, the hypernym relations do not always correspond to intuitive subclass relationships. As part of the ontology building process in TRIPS, we have identified synsets in WordNet that correspond to TRIPS types. These mappings enable us to derive the TRIPS types and their subcategorization patterns for words not found in the TRIPS lexicon.

For example, Figure 3 shows a very small fragment of the TRIPS ontology hierarchy mappings from the WordNet synset hierarchy. Here we see a slice of the TRIPS ontology going up from ONT::DECREASE, a class for events that involve the lowering of position of an object on a physical or abstract scale (e.g., temperature, weight). ONT::DECREASE is a subclass of ONT::CHANGE-MAGNITUDE (events of objects changing position on a scale), which is a subclass of ONT::CHANGE (events of objects changing in some way). On the WordNet side we

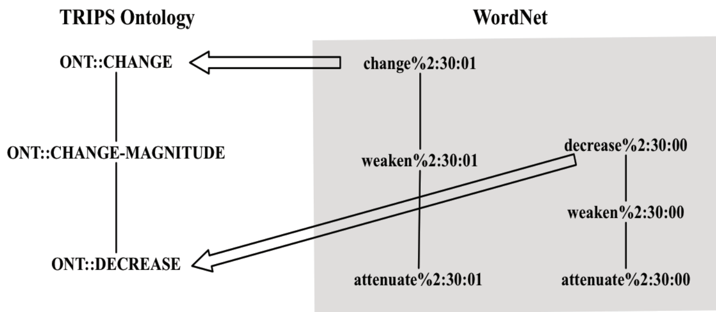

Figure 3: Example of Mapping from WordNet to the TRIPS Ontology

see two hypernym hierarchies starting from two synsets for the word *attenuate*. WordNet has finer grained sense distinctions than most people would like, but we can use the resource as is. Two WordNet synsets in this fragment have mappings to TRIPS types, indicating we believe these synsets capture a subset of the events in the corresponding TRIPS types.

The key idea in generating abstract lexical entries for unknown verbs builds from the same intuition that motivated VerbNet - that the set of constructions a verb supports reflects its semantic meaning. While in VerbNet the constructions are used to cluster verbs into semantic classes, we work in the opposite direction and use the semantic classes to predict the likely syntactic constructions.

To generate the lexical entries for an unknown word we use the TRIPS/WordNet mappings as follows.

Given an unknown word *w:*

i. Look up word *w* in WordNet and obtain its possible synsets
ii. For each synset, find a mapping to the TRIPS ontology
    - If there is a direct mapping, return the mapping
    - Otherwise traverse up the WordNet hypernym hierarchy and recursively check for a mapping
iii. For the set of mapped TRIPS ontology types, find the most specific types and eliminate the more general ones
iv. For each remaining TRIPS type
    - Gather all the templates from lexical entries in this type
    - For each template, generate a lexical entry for *w* with this ontology type and template

The result of this process is a possibly over-generated set of underspecified lexical entries.

As *attenuate* is not in the TRIPS lexicon, the above algorithm would identify two possible TRIPS senses for the word: ONT::DECREASE and ONT::CHANGE. The latter would be eliminated as it is more general than ONT::DECREASE, which contains the following lexical entries and associated templates:

decrease	AGENT-AFFECTED-TEMPL
compress	AGENT-AFFECTED-TEMPL
constrict	AFFECTED-TEMPL

We then generate the following possible lexical entries (ontology type and template pairs) for *attenuate*:

ONT::DECREASE
  AGENT-AFFECTED-TEMPL
  e.g., *They attenuated the pressure.*
ONT::DECREASE
  AFFECTED-TEMPL
  e.g., *The pressure attenuated.*

As we will discuss in the next section, these templates can also support resultative constructions even though the RESULT role is not explicitly named in the templates. For example, in *They attenuated the pressure to 4*, the parser will be able to identify "*to 4*" as a result of the attenuate event.

We use similar techniques to generate lexical entries from words tagged by the named entity recognizers, although these are typically common nouns of a few select types (e.g., people names, companies, proteins) that only need the basic noun templates.

Finally, for words that are not known to TRIPS or WordNet or the named entity recognizers, we build abstract lexical entries based on their syntactic types. Unknown verbs would take the default EVENT-OF-CHANGE ontology type with AGENT-AFFECTED (transitive) and AFFECTED (intransitive) templates, while unknown nouns assumes the general type ONT::REFERENTIAL-SEM and are treated as standard count nouns. If we have part-of-speech tagging advice that indicates other features (e.g., singular/plural, infinitive/past participle) these features are also included in the generated entries.

While we end up with underspecified semantics for such words, these vague entries allow the parser to continue to find a plausible parse. Similarly, although the TRIPS-WordNet mappings possibly overgenerate entries for an unknown word, it is more important to cover the possible constructions than to exclude impossible constructions.

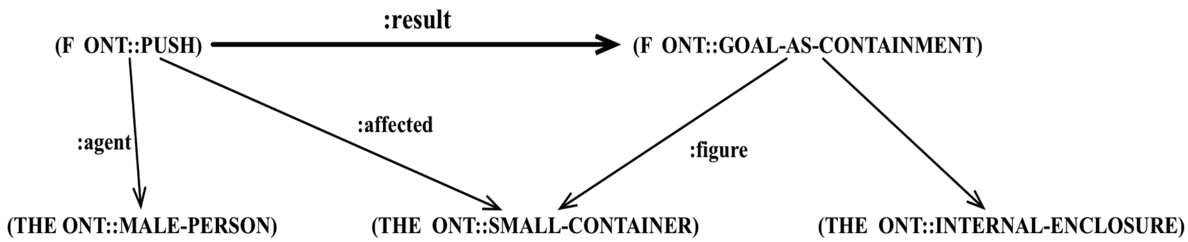

Figure 4: The logical form for the resultative construction of "He pushed the box into the room"

TRIPS and Constructions

While using many of the techniques of traditional computational syntactic grammars, TRIPS also uses semantically and pragmatically motivated rules during parsing. Here we discuss two such examples. One is the encoding of conventional constructions relating to intention, i.e., the speech acts. The second example looks at the TRIPS treatment of the resultative construction in which the grammatical rule contributes key parts of the semantics of the sentence.

Conventional Constructions of Spoken Language

While grammars are typically viewed as the domain of syntax, once semantics is fully integrated, they become a quite flexible framework for capturing constructions typically ignored in syntactic work. Many of these constructions in spoken language are handled in the TRIPS grammar by mapping to surface speech acts. For instance, common utterances such as *uh-huh* and *OK* are mapped to the speech act ONT::SA_ACK (an acknowledgement act). Similarly, *yes* and *no* are mapped to ONT::SA_RESPONSE (assuming there is a previous yes/no question).

A more general rule maps utterances consisting of an evaluative adjective phrase (e.g., *good, not bad, very sweet*) to ONT::SA_EVALUATE, used when the speaker responding is expressing an opinion about the current state of affairs under discussion. The useful construction *How about X?,* where *X* could be a person (*How about John?)* or event (*How about going out for coffee?),* maps to ONT::SA_REQUEST-COMMENT acts. There are also the conventional greetings, apologies, thanks, etc that are such a key part of social life. By viewing the output of parsing as a speech act situated in a dialogue rather than a syntactic sentence, we have the flexibility to encode the conventional meanings of such constructions in a straightforward way.

The Resultative Constructions

Many computational lexicons attempt to define all the possible uses of a verb and its argument structure. For instance, the entries for *push* in VerbNet (funnel-9-3) and Propbank both explicitly indicate that *push* may take a *Destination* argument such as *He pushed the box into the room.* Our approach, as in construction grammar, is that the meaning of such sentences is computed by a syntactic-semantic rule VP-RESULT-ADVBL that applies to a wide range of event types. The rule captures one instance of the resultative construction (Goldberg and Jackendoff, 2004; Boas, 2005), and can be described informally as follows:

> *A transitive event of type EVENT-OF-CAUSATION followed by an adverbial construction of type PATH or TRAJECTORY asserts that the event results in the adverbial proposition being true of the direct object.*

As a result of such constructions, the lexical entry for *push* (in a subclass of ONT::APPLY-FORCE) is simply an AGENT-AFFECTED transitive verb. Using the VP-RESULT-ADVBL construction, we get the logical form shown in Figure 4 for *He pushed the box into the room.* The semantics of the RESULT link is that the result proposition is caused by the event and remains true after the event. The TRIPS roles also include SOURCE (in which a proposition true at the start of the event ceases to be true as a result of the event) and TRANSIENT-RESULT (where a proposition caused by the event also ceases to be true by the end of the event). By using these roles and a few variants of the VP-RESULT-ADBVL construction, we can handle a wide range of resultative constructions ranging from forms commonly associated with the event, to ones that are rarer but still understandable, such as *He pushed it flat*, and *He combed his dog to sleep*.

Other constructions coerce non-transitive verbs into transitive ones with a result, which handle the classic example *He sneezed the dust off the table*, as well as *He sang me out of the room, The dog barked the cat up the* *tree*, *He painted himself into a corner* and *He talked me deaf.*

While rarer cases are a lot of fun, the largest impact of this approach is a dramatic reduction in the size of the lexical entries. For example, the verb *push* needs only to be encoded in its ONT::APPLY-FORCE sense, and all the other variants that entail movement (e.g., *He push it out of the truck*, *He pushed it into the room, He pushed it along the path)* are handled by construction rules.

Conclusion

We described the mechanisms that enable the TRIPS system to achieve broad coverage, domain independent deep semantic parsing. Chief among these are the development of an expressive representation and a rich ontology, lexicon and grammar that interleave syntactic and semantic constraints that capture many of the constructions discussed in the literature. TRIPS has been deployed in many diverse application domains, requiring minimal customization efforts other than providing domain-specific named entity recognition where needed. A suite of systems, each customized to different domains and language genres, are available as interactive tools online and as parsing web services for those who would like to access the parser programmatically (see footnote 1).

Acknowledgements

This work has been partially supported by the DARPA Big Mechanism program under ARO grant W911NF-14-1-0391 and the DARPA CwC program under ARO grant W911NF-15-1-0542.

References

Allen, J., M. Swift and W. de Beaumont (2008). Deep Semantic Analysis for Text Processing. Symposium on Semantics in Systems for Text Processing (STEP 2008).

Allen, J. F., W. de Beaumont, L. Galescu and C. M. Teng (2015). Complex event extraction using DRUM. Workshop on Biomedical Natural Language Processing (BioNLP).

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. (1998). The Berkeley FrameNet Project. In Proceedings of COLING-ACL'98, pages 86–90, Montréal, Canada.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider (2013). Abstract meaning representation for sembanking. Proc. Linguistic Annotation Workshop.

Boas, Hans (2005). Determining the Productivity of Resultative Constructions. *Language* 81(2), pages 448-464.

Branavan, S., L. Zettlemoyer, and R. Barzilay (2010). Reading between the lines: Learning to map high-level instructions to commands. In ACL, pages 1268–1277.

Cai, Shu and Kevin Knight (2013). SMatch: an evaluation metric for semantic feature structures. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 748–752, Sofia, Bulgaria, August. Association for Computational Linguistics.

Chen D. L. and R. Mooney (2011). Learning to interpret natural language navigation instructions from observations. Proc. AAAI (AAAI-2011), pages 859–865.

Fellbaum, Christiane (1998, ed.). WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Gangemi, A., Navigli, R., Velardi, P. (2003). Axiomatizing WordNet Glosses in the OntoWordNet Project. Workshop on Human Language Technology for the Semantic Web and Web Services, 2nd International Semantic Web Conference (ISWC2003). Sanibel Island, Florida, October 20-23, 2003.

Gazdar, Gerald; Ewan H. Klein; Geoffrey K. Pullum; Ivan A. Sag (1985). Generalized Phrase Structure Grammar. Oxford: Blackwell, and Cambridge, MA: Harvard University Press.

Goldberg, Adele (1995). Constructions: A Construction Grammar Approach to Argument Structure. University of Chicago Press.

Goldberg, Adele and Ray Jackendoff (2004). The English resultative as a family of constructions. *Language* 80(3), pages 532-568.

Jackendoff, Ray (1977). X-bar-Syntax: A Study of Phrase Structure. Linguistic Inquiry Monograph 2. Cambridge, MA: MIT Press.

Kipper, K., A. Korhonen, N. Ryant, and M. Palmer (2008). A large-scale classification of english verbs. Language Resources and Evaluation Journal, vol. 42, pp. 21–40, 2008.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

Manshadi, Mehdi H., James Allen and Mary Swift (2008). Towards a Universal Underspecified Semantic Representation. 13th Conf. on Formal Grammar. Hamburg, Germany.

Matuszek, C., E Herbst, L Zettlemoyer, D. Fox (2012). Learning to Parse Natural Language Commands to a Robot Control System, Proc. of the 13th International Symposium on Experimental Robotics (ISER).

Niles, I. and Pease, A. (2001). Towards a Standard Upper Ontology. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Chris Welty and Barry Smith, eds.

Palmer, M., Gildea, D. and Kingsbury, P. (2005). The Proposition Bank: A corpus annotated with semantic roles, Computational Linguistics Journal, 31:1.

Tellex, S., P Thaker, J Joseph, N Roy. (2013). Learning Perceptually Grounded Word Meanings From Unaligned Parallel Data. Machine Learning Journal.