The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

# Effective Broad-Coverage Deep Parsing

**James F. Allen, Omid Bahkshandeh, William de Beaumont, Lucian Galescu, Choh Man Teng**
IHMC, 40 S. Alcaniz St, Pensacola, FL 32501
{jallen, omidb, wbeaumont, lgalescu, cmteng}@ihmc.us,

## Abstract

Current semantic parsers either compute shallow representations over a wide range of input, or deeper representations in very limited domains. We describe a system that provides broad-coverage, deep semantic parsing designed to work in any domain using a core domain-general lexicon, ontology and grammar. This paper discusses how this core system can be customized for a particularly challenging domain, namely reading research papers in biology. We evaluate these customizations with some ablation experiments.

## Introduction: Broad, Deep Semantic Parsing

Representing the underlying meaning of language has been of interest to computational linguistics for a long time. Recently there has been a renewed interest in developing effective parsers that can generate deep semantic representations. This endeavor is complicated by the fact that the search space for semantic parsers is far greater than that for syntactic parsers. Specifically, context-free syntactic parsers can take advantage of the fact that one only needs to consider a single constituent of any syntactic type (e.g., NP, S) between any two positions in the sentence. This property enables effective parsing algorithms such as the Earley algorithm (Earley, 1970) and various Chart-based parsing strategies. Semantic parsers, on the other hand, cannot make this assumption. A single noun phrase might have many different semantic meanings due to word sense ambiguity as well as attachment ambiguity within the noun phrase. Thus a key problem for semantic parsers is how to manage this larger search space.

Researchers have dealt with this problem in different ways by limiting the scope and/or depth of the representations produced. Broad-coverage semantic dependency parsers such as those developed and tested in SemEval series (Oepen et al., 2014) generate a shallow and partial semantic representation. Some other semantic parsing aproaches such as Das (2014), essentially tag the words that evoke FrameNet (Johnson and Fillmore, 2000) frames, and identify the word sequences that act as arguments to the identified frame elements, which is by no means a detailed enough representation to support significant reasoning or inference. On the other hand, semantic parsers that are trained to produce AMR representations (Banarescu et al, 2013) produce a richer semantic representation that identifies the full predicate argument structures in the sentence. However, AMR representations typically only assign word senses to the words denoting events, tagging them with senses from PropBank (Palmer et al, 2005). The rest of the sentence, including most nouns, adjectives and adverbs remain as lexical items not tagged with senses.

Deep semantic parsers can be found for some specific domains, where the parser learns to map language input into an executable meaning representation. Many of these involve generating queries to databases (Branavan et al, 2010; Chen et al, 2011, 2013; Berant et al, 2013, Zhong et al, 2017; Pasupat, 2015) or generating a sequence of commands to a robot (Matuszek, et al, 2012; Tellex et al, 2013). The representations produced, however, are highly specific to the domain and are not transferable to other domains. Because of the high domain specificity, limited inventory of word senses, and relatively simple sentences, they do not hit significant search issues when parsing.

Unlike the previous work, the goal of the TRIPS parser (Allen & Teng, 2017) is to produce a semantic parser that is 1) broad-coverage, 2) domain-generic, and 3) deep[1]. We define broad-coverage to mean that the parser produces a semantic representation for any given English sentence. Domain-generic means the parser is usable for input text or ASR output in any domain, and deep means that it generates a representation of the meaning of the sentence that

[1] Currently, the TRIPS parser, using the same grammar, lexicon and ontology, is in active use in many domains, including the following: reading research papers in biology to extract information about causal models of biological pathways (Allen et al, 2015); understanding text-based conversation with teens about managing asthma (Rhee et al, 2014); understanding human/system dialog to collaboratively plan and build structures in a physical blocks world (Perera et al, 2017); understanding dialog about music for collaborative music composition (Quick & Morrison, 2017).

has a semantics clear and comprehensive enough to support automated reasoning (such as deduction and intention recognition, among others). The advantage of such a system is that it provides a generic semantic parser that can be used in a wide range of domains and applications, much like the various syntactic parsers, such as the Stanford CoreNLP parser (Manning et al, 2014) which is now used “off the shelf” for many tasks[2].

This paper describes how the TRIPS parser is customized to operate effectively in a specific domain, namely reading research papers in biology. We have shown that this system can identify and extract relevant biological events at a level comparable with human performance on the same task (Allen et al, 2015). Here we explore which of the customizations enable this performance.

We start by providing a quick overview of the TRIPS parser, the meaning representation it produces, the ontology, grammar, lexicon and the general operation of the parser. We then describe various customization options that can be used to attain good performance. We will discuss problems of attaining broad lexical coverage (i.e., attaining the extent of WordNet (Fellbaum et al, 1998)), of focusing search when dealing with complex sentences, and of adding domain specific information (both relevant named entities and relevant word senses). We then explore issues in extracting relevant knowledge into a desired domain-specific representation. The key enabling mechanism for almost all these extensions is the TRIPS ontology. As we show in the next section, it provides the link between the parser and the domain-specific customizations. We end with a series of ablation experiments, where we ablate various aspects of the system and show how each affects the overall performance.

## A Quick Overview of the TRIPS Parser

The TRIPS parser produces a semantic representation that formally is an underspecified scoped modal logic that subsumes prior representations such as MRS (Copestake et al, 2005) and Hole Semantics (Blackburn and Bos, 2005) (Manshadi et al, to appear). While it can be written in several equivalent formats, the most readable for humans is the graphical representation, where each node is a discourse entity labeled with its ontology type and quantifier information, and the links indicate argument relationships (semantic roles) and scoping/modification constraints. As a very simple example, Figure 1 shows the semantic graph for the sentence *He tried to buy the square pizza,* identifying the speech act, word senses, semantic roles, modification relationships, quantification and more. One might note the structural similarity to the AMR representation. Some of the key differences are that the TRIPS LF includes sense labeling for every word, the presence of quantifier information, a richer semantics of predicates from adjectives and prepositions, and analysis of the surface speech act. Particularly important is the fact that all the senses are drawn from an extensive ontology (namely, the TRIPS ontology), whereas AMR only tags the words denoting events using PropBank verb senses. This difference can be illustrated by considering the lexical entries for two verbs: *buy* and *purchase*. In the TRIPS ontology, as in WordNet, these two words share a sense in common (ONT::PURCHASE). In the AMR representation, these items have senses buy.01 and purchase.01 respectively which are not connected in any systematic way.

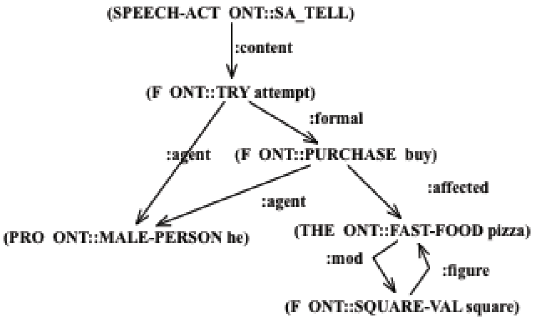
(SPEECH-ACT ONT::SA_TELL)
:content
(F ONT::TRY attempt)
:formal
:agent
(F ONT::PURCHASE buy)
:affected
:agent
(PRO ONT::MALE-PERSON he)
(THE ONT::FAST-FOOD pizza)
:mod
:figure
(F ONT::SQUARE-VAL square)

*Figure 1: The meaning representation of “He tried to buy the square pizza”*

Key to linking all the word senses is the TRIPS ontology, which provides an upper ontology for English words. The event ontology is organized by the temporal/causal properties of the events. The top distinction is between events involving change and events of state (corresponding to the active vs. stative distinction made in linguistics). The ontology also is strongly interlinked with the semantic roles associated with each class. It defines an inheritance hierarchy of semantic roles. For instance, all events that require an AFFECTED role (the object undergoing some force or change) are under ONT::EVENT-OF-CHANGE, while all events involving the EXPERIENCER role (for entities in cognitive/perceptual states) are under ONT::EVENT-OF-EXPERIENCE. As one moves down the hierarchy we find events that correspond roughly to many of the frames in FrameNet or the classes of VerbNet (Kipper et al, 2008), although the TRIPS ontology often divides the lexical items into finer grained classes due to its emphasis on organizing the ontology based on both linguistic realization and temporal/causal entailments. The ontology of non-events, such as physical and abstract objects, to a very rough approximation resembles an abstraction of the noun hierarchy in WordNet. The ontology of properties (adjectives and adverbs) is organized around scales/domains (e.g., the property *Low* is defined in terms

[2] A range of parsers customized to different domains can be found at http://trips.ihmc.us/parser.

of the *Height* scale).

The information in the ontology is key to the word sense disambiguation performed by the parser. Each ontology type can specify semantic preferences on the objects that can fill its argument roles. When parsing, the system attempts to identify combinations of word senses that violate the fewest preferences. More precisely, the parser is a chart-parser using a best-first search strategy. Each grammatical rule and lexical entry used incurs a small cost, and semantic preference violations incur additional cost. The parser searches until a specified number of spanning interpretations are found. These are guaranteed to be the least-cost interpretations.

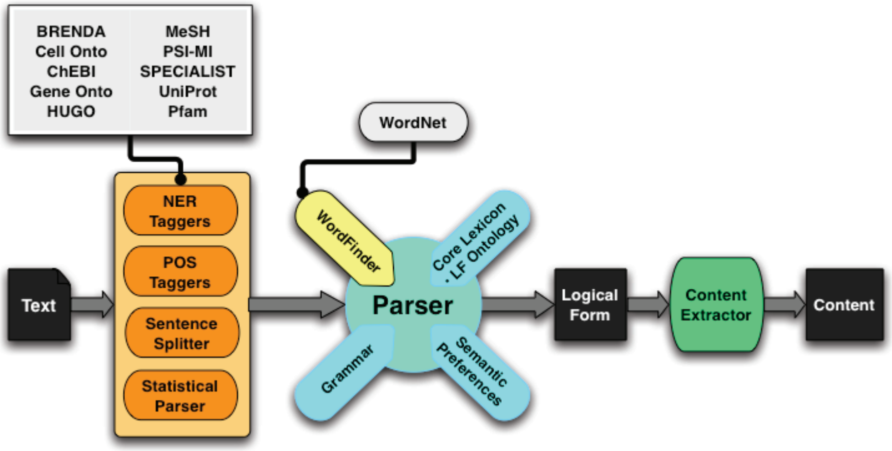

*Figure 2: Customizing the TRIPS parser for the biology domain*

### Expanding the Vocabulary via WordNet

The fact that the lexicon is organized via the ontology allows for automatic extension of the lexicon to attain nearly full coverage of WordNet. This is enabled by a hand-built mapping from WordNet senses to TRIPS ontology types[3]. Complete coverage of WordNet senses is feasible due to the hypernym hierarchy that allows one to define mappings at the right level of abstraction for the TRIPS Ontology. Using the mapping, the system can determine the ontology type for a given word. The challenge then is to determine the appropriate argument structures and semantic preferences for the arguments. This is done by gathering up the argument structures for existing lexical items associated with the ontology type and using these as candidate structures for the new target word. For instance, the word *attenuate* is not in the TRIPS lexicon. One of its WordNet synsets, *attenuate%2:30:00*, has a hypernym *decrease%2:30:00*, which maps to ONT::DECREASE. Words in the TRIPS lexicon of this ontology type include *decrease, constrict* and *compress*. All the argument structures for these words are then used to build lexical entries for *attenuate*. For more details, see Allen & Teng (2017).

### Constituent Advice

A key form of preprocessing provides advice on constituent boundaries to the parser. One or more statistical syntactic parsers are used to identify likely syntactic boundaries of the major constituents (e.g., S, VP, NP, ADJP and ADVP). This analysis produces a constituent bracketing of the sentence that is used to influence the semantic parse in two ways: first, constituent hypotheses proposed by the semantic parser that cross the brackets in the advice can be penalized; second, semantic constituents that exactly match the syntactic analysis can be boosted. By varying these penalties and boosts we can guide the search of the semantic parsing, yet still allow semantic issues to determine the final analysis. We will evaluate below the tradeoffs in these parameter settings.

## Domain Customization of the Parser

Figure 2 shows the overall architecture of the DRUM system, which is the TRIPS parser customized for reading research papers in biology.

### Front End Components

DRUM includes a number of domain-specific preprocessing components, including several off-the-shelf tools such as the Shlomo Yona sentencizer, the Stanford part-of-speech tagger (Toutanova and Manning, 2000), the Stanford named-entity recognizer (NER) (Finkel et al., 2005) and the Stanford Parser (Klein and Manning, 2003), the Enju parser (Hara et al., 2005), trained specifically with the GENIA corpus (Kim et al., 2003), and many ontologies and other lists of terms in the biomedical domain[4].

### Genre Specialization

The chart produced by the parser is 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, in dialogue systems speech acts such as GREET (e.g., hello) are expected. For papers

[3] The native lexicon and the ontology with the WordNet mappings are at www.cs.rochester.edu/research/trips/lexicon/browse-ont-lex.html

[4] Named entities used in other domains include geographic names (geonames.usgs.gov/domestic/index.html), and personal names (www.ssa.gov/oact/babynames/limits.html).

in the biomedical domain, such speech acts almost never occur and thus are discounted in favor of TELL statements. Similarly, in dialogue systems utterances are expected to be fairly short and colloquial, whereas in scientific text the sentence structures are expected to be much more formal and involved. The parameters for parsing and the cost table are set accordingly.

## Named Entity Recognition

The named entity tagger takes its data from many external ontology and vocabulary resources. These are merged into one table with information on each term, which concepts it may represent in the external ontologies, and the term's status in relation to the concept (e.g. preferred name vs. synonym). For resources that have hierarchies for these concepts, the tagger also saves the hierarchy in a common format, and uses it to help map each term to one of the TRIPS ontology types. It maintains a relatively small set of mappings from high-level concepts in the external ontologies to a few of the TRIPS ontology types, and it follows is-a relationships up from a matched term, through its ontology, through one or more of these mappings, to the TRIPS ontology. Resources without hierarchies are generally mapped to a single ontology type. To deal with multiple conflicting matches, the matches are scored based on the term status and on differences in capitalization and punctuation. The TRIPS parser uses these scores to guide its search.

Within the same tagger, there is some specialized code for specific kinds of biology-related terms, including micro RNAs, amino acids, molecular sites, mutations, and modifications. The post-translational modifications are derived from part of the Gene Ontology, but the others are tagged according to relatively formulaic naming conventions. This code also produces some domain-specific information for these terms (for example, the site of a mutation, whether it is an insertion, deletion, or substitution, and which amino acids are involved), which is passed on to the parser in the same way as the concept IDs from the external resources.

## Event Extraction (Content Extractor)

In the biology domain, as well as in other applications, we often aren't interested in the full logical form, but desire a more focused representation of specific content. Specifically, in DRUM we are interested in biological entities, biological events and event relationships. Because much of the variation in sentence constructions is handled by the extended TRIPS system, we are able to use a relatively compact and easy-to-maintain specification for extracting such events and relationships from the logical form, while coping with fairly complex and nested formulations.

Instead of having to write one rule to match each keyword/phrase that could signify an event, many of these words/phrases have already been systematically mapped to a few types in the TRIPS ontology, using a combination of the TRIPS internal lexicon and the extension from WordNet. For instance, *accumulate, gain, amplify, multiply, boost, double* all map to the TRIPS ontology type ONT::INCREASE.

```
(EVENT V31830 ONT::REGULATE :AGENT V826 :AFFECTED V848)

(EVENT V826 ONT::ACTIVATE :AFFECTED V318)

(EVENT V848 ONT::PHOSPHORYLATION :AFFECTED V845
:DRUM ((:DRUM :ID GO::|0016310| :NAME "phosphorylation")))

(TERM V318 ONT::PROTEIN-FAMILY :NAME W::RAS :DRUM
((:DRUM :MEMBER-TYPE PROTEIN :MEMBERS (HRAS NRAS
KRAS)))

(TERM V845 ONT::PROTEIN :NAME W::ASPP-2 :DRUM ((:DRUM
:ID UP::Q13625)))
```

*Figure 3: Extracted events and terms from "RAS activation regulates ASPP2 phosphorylation"*

In addition, the parser handles various surface structures, and the logical form contains normalized semantic roles. For example, *RAS activates RAF, RAF is activated by RAS, The activation of RAF by RAS, Activated RAF, RAF activation* all are parsed into the same basic logical form with the semantic roles AFFECTED: *RAF* and, where applicable, AGENT: *RAS*. Thus, very few (often only one) extraction rule specifications are needed for each event type, covering a wide range of words and syntactic patterns.

As an example, consider the sentence "*RAS activation regulates ASPP2 phosphorylation.*" There are three events in this sentence: the central *regulation* event and two nested events, *activation* and *phosphorylation*. The extractions of the three events are shown in Figure 3, together with the two terms, *RAS* and *ASPP2*. Note that the word "activation" is mapped to the TRIPS ontology type ONT::START. It is this ontology type that triggers the following extraction rule for an ACTIVATE event:

```
rule-activate (40): ACTIVATE(AGENT, AFFECTED)
← ONT::start (AGENT, AFFECTED)
```

Similar rules extract the regulation and phosphorylation event as well.

## The Extraction Knowledge Base

The extracted content is assembled into a graph-based representation, called the Extraction Knowledge Base (EKB). Typically, there is a single EKB for the full textual input (e.g., from reading a full paper). In this evaluation exercise, however, we build one for each sentence. The EKB is serialized into an XML-based format, which can be used by various reasoners. For example, there are inference rules to infer that a phosphorylation event changes the state of the substrate from an unphosphorylated state to a phosphorylated state. Gyori et al. (2017) show how dynamic molecular models can be built from the EKB derived from text describing molecular mechanisms, using additional sources of information.

While the EKB is meant primarily as a knowledge representation, it includes information about textual provenance, so its serialization can also serve as an annotation format. Thus, it is possible to convert from this format to any number of existing event annotation formats used in the literature, such as the PubAnnotation JSON format (Kim and Wang, 2012), the standard BioNLP-ST standoff format. Generally, though, these conversions are lossy, due to the fact that TRIPS uses a richer ontology for events, entity types, modality, and causal relations compared to the ones typically used in the community.

# Evaluation

We use as experimental data 60 sentences extracted from various systems biology papers obtained from PubMed Central. In general, these sentences were deemed (by us or by a third party) as containing useful information about biomolecular mechanisms. In some cases we retain only one meaningful clause from a longer sentence. The average sentence length is 12.5 words. Here are a few examples:

> *Protein kinase A inhibits ERK1/2 by interfering with the activation of Raf-1 by Ras.*
> *Ack1-mediated AKT Tyr176-phosphorylation resulted in translocation of Ack1/AKT complex to the nucleus.*
> *Sorafenib induces apoptosis in AML cells through Bim.*

For evaluating parsing performance, we constructed a set of gold annotations for the test set. Four researchers post-edited the semantic representations produced by TRIPS. The annotations were carried out via an interactive graph-based annotation tool (Bakhshandeh et al., 2016). This tool set up the data collection as a two-step annotation process: (1) For each given sentence, one annotator annotated the sentence, and (2) another annotator reviewed the annotation and either returned the annotation with feedback to the first annotator or marked it as gold. We iterated over the sentences until getting 100% inter-annotator agreement.

Reference EKBs were curated by one of the authors familiar with both the system and the domain, and then reviewed by a second researcher for accuracy.

Given the gold annotated logical form graphs, we computed the accuracy of the semantic representations produced by the system using the Smatch metric (Cai and Knight, 2013), developed for comparing AMR representations. We use Smatch to compute the accuracy of the sentences parsed with various ablations with respect to the gold standard. We report the precision, recall and F1 scores.

Knowledge extraction performance is measured by computing precision and recall at the level of EKB assertions (entities, events, causal relations). That is, a hypothesized assertion matches the reference assertion if and only if its type, all attributes and all entities or events that have some role in that assertion (e.g., participants in an event, or attributes of an entity that reference another entity) match with an identical role. Note that this is a fairly strict criterion. Any one mismatch of the items in the EKB assertion would result in its being marked as incorrect.

## Results

Figure 4 shows the results of the ablation experiment, in which one aspect of the system is deleted in each row. The first row gives the performance of the full system and shows the precision, recall and F1 scores of the EKB score (i.e., how many extractions we got exactly correct) and the Smatch precision, recall and F1 scores on the parser output. Thus, the full system has an EKB F1 score of 83.28% associated with a parser F1 score of 87.63%. The next three rows show the effect of ablating advice from the statistical parsers. If we use just the CoreNLP parser for advice (i.e., ablate the Enju parser), we see a slight drop in the EKB score to 82.21%, with a parser Smatch score of 86.27%. If on the other hand, we only use Enju for advice (and ablate the CoreNLP parser), performance drops to 79.75% EKB score with a slight decrease in Smatch score. Finally, with no statistical parsing advice at all (but with named entity recognition and WordNet) we get an EKB score of 69.75% – a relative performance degradation of over 16%! – from semantic representations with a Smatch score of 78.92%. In summary, the advice from the syntac-

| *Ablated Feature* | ***EKB Score*** | | | ***Smatch Score*** | | |
|---|---|---|---|---|---|---|
| | ***Precision*** | ***Recall*** | ***F1*** | ***Precision*** | ***Recall*** | ***F1*** |
| ***Full (no ablation)*** | *84.60%* | *82.00%* | *83.28%* | *86.62%* | *90.07%* | *87.63%* |
| ***- Enju*** | *82.46%* | *80.78%* | *82.21%* | *85.65%* | *88.27%* | *86.27%* |
| ***- CoreNLP*** | *80.58%* | *78.94%* | *79.75%* | *86.00%* | *86.67%* | *86.02%* |
| ***- Stat. Parsers*** | *71.71%* | *67.89%* | *69.75%* | *78.57%* | *81.45%* | *78.92%* |
| ***- WordNet*** | *83.69%* | *80.78%* | *82.21%* | *86.38%* | *87.52%* | *86.53%* |
| ***- NER*** | *20.48%* | *8.79%* | *12.30%* | *67.47%* | *69.30%* | *67.82%* |
| ***Bare*** | *26.15%* | *6.95%* | *10.99%* | *58.02%* | *54.95%* | *55.52%* |

*Figure 4: Results of the ablation experiment*

tic parsers clearly make a significant improvement in performance, and combining the advice from the two parsers (eliminating any disagreements) provides better advice than using either alone.

The fifth row gives the performance if we ablate the capability to generate new lexical entries based on WordNet. Here we see only a minor effect of about 1%, presumably reflecting the nature of the corpus where there is a significant amount of technical vocabulary that is handled by the named entity recognizer. In other domains, we have found WordNet lookup to be very useful. In the biology domain, it provides only incremental benefit.

The sixth row shows what happens if we ablate named entity recognition. We see a dramatic decline in performance with an F1 EKB score of only 12.30%, reflecting the fact that sentences that describe molecular mechanisms use many names (genes, proteins, etc.), as can be seen from the examples provided above. Clearly named entity recognition is essential in the biomedical domain. The parser Smatch F1 score, however, remains quite respectable at 67.82%, which we note is about the level of performance of the best current AMR parsers. This is the result of a NER backoff strategy in the core parser in which unknown words are mapped to semantically underspecified nouns. Thus, the parser still can build much of the right semantic information around these unrecognized names.

Finally, as a point of reference, in the last line we report the performance of the core parsing system with no preprocessing or extensions (i.e., everything ablated). As expected, we see a dramatic decrease in performance, with only a 10.99% F1 score on the EKB extractions, even as the parses get a moderate F1 score of 55.52%.

While there is a high correlation (0.95) between the parser and EKB scores, there is significant difference between the actual scores attained by each for the individual ablation conditions. For example, as we discussed above, without NER we still get respectable parses but almost no usable EKB information.

## Discussion

It is not surprising that named entity recognition is such a critical component of a system reading biology papers. The text is full of technical jargon and a vast number of named entities (not only protein and cell names, but biological processes, binding sites, genes, and more). By integrating named entity taggers so that these domain specific entities are classified into the TRIPS ontology, these terms can participate fully in the semantic parsing. It is also not surprising that the WordNet lookup has only a small effect. WordNet is a substantial resource of general everyday English but has scant coverage of the specialized biomedical vocabulary.

Perhaps the most complex interaction is that between the advice generated by the statistical parsers and the TRIPS semantic parser. First note that one cannot avoid the need for deep semantic parsing. The statistical parsers generate only syntactic information and so do not produce the information needed to build the EKB. On the other hand, just using the semantic parser with no guidance about the syntactic structure leads to a significant decline in performance. As described above, we use two different methods to control the influence of the syntactic parsers on the search. First, the *bracket crossing penalty* penalizes constituents produced by the semantic parser that are inconsistent with the syntactic structure. The effect of the bracket crossing penalty is determined by a parameter that varies from 0 to 1. When a violation occurs, the score of the constituent is modified multiplicatively. So, if the parameter is set to 0, then any inconsistent constituent is immediately eliminated from future consideration. If the parameter's value is 1, then inconsistent constituents are not penalized, or equivalently, the advice is ignored. For values between 0 and 1, the semantic coherence of an interpretation can override the advice from the syntactic parsers, but with varying penalties.

Second, the *constituent boost* reinforces constituents produced by the semantic parser whose boundaries exactly agree with the syntactic constituents. This parameter varies between 0 and 1, denoting the percentage of the difference of the current score and 1.0 that should be added to the constituent score. For example, with a boosting factor of 0.2, a constituent with a score of 0.9 would be boosted to a score of 0.92.

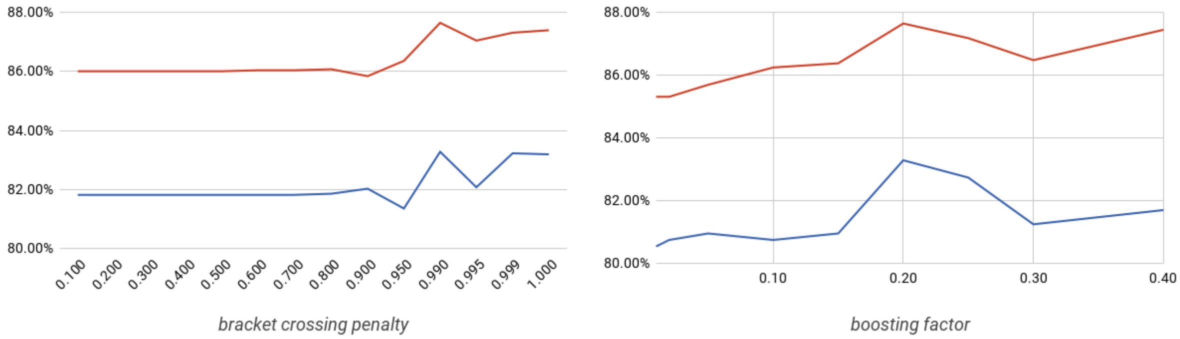

*Figure 5: The effect of parsing parameters on the EKB F1 score (blue lines) and the Smatch F1 score (red lines). The chart on the left shows the scores for different values of the bracket crossing penalty when the boosting factor is 0.2 (note: the X axis is not to scale). The chart on the right shows the effect of varying the boosting factor when the bracket crossing penalty is 0.99.*

All the ablation tests in Figure 4 were performed with a bracket crossing penalty of 0.99 and a boosting parameter of 0.2. We explored the effect of changing these parameters. In Figures 5 we plotted both the EKB and Smatch scores. The first graph shows how these scores vary for a range of bracket crossing penalties with the boosting factor locked at 0.2. Here we see the complex effect in balancing the advice. At the lower values (higher penalties), the parser is essentially forced to follow the syntactic advice even if it is semantically less preferable. As the parameter approaches 1, and thus the bracket crossing penalty becomes less onerous, it becomes possible that semantic considerations can overcome the penalty, producing improved performance. If the parameter is set to 1, this results in a drop in the performance as the parser is only using its semantically driven preferences and ignores advice from the syntactic parsers. Thus, only the smallest nudge from the bracket crossing penalty is needed for best performance.

Likewise, if we lock the bracket crossing penalty at 0.99 and vary the boosting factor we see a similar phenomenon, but most likely for different reasons (rightmost graph in Figure 5). The boosting factor increases the score of constituents that exactly match the constituent advice provided by the statistical parsers. Assuming such constituents are more likely to contribute to the final interpretation if they match the syntactic predictions, this can help focus the search in more promising areas. The down side of higher boosting factors, however, is that boosting adds a slight chaotic element to the best-first search. For example, suppose we have two possible noun phrases that span the same section of a sentence and have the same score based on semantic plausibility, but differ on semantic grounds (e.g., different senses or semantic roles). The order in which these constituents are processed is arbitrary, but the first one selected receives a significant boost in its score due to matching the syntactic advice. The difference in score between the two might become so great that the second possibility essentially does not receive any further attention in the search. By keeping the boosting factor low, the chance of this happening is reduced. We believe this is the phenomenon that is driving the curve in Figure 5.

## Conclusion

We have shown how a generic semantic parser can deliver both broad coverage and deep representations in complex domains such as reading biology papers. To attain such high performance, the basic system is augmented with an extensive named entity recognition component and provided with advice about the constituent structure generated by statistical syntactic parsers. While there is not the space to demonstrate this here, the same parsing system, with identical grammar, lexicon and ontology, performs equally well in many different domains, including dialogs involving collaborative planning in a blocks world, texting with teens about their asthma, and reading simple short stories about everyday events.

We studied the effect of the various components and parameters in the ablation experiments. Note that because of the way the test set was constructed, both parsing and knowledge extraction performance (Smatch and EKB scores) reported here are not indicative of the performance of the system in the wild (e.g., for reading full papers). Rather the goal of this work was to understand how the different extensions and customizations of the parser affected performance. Good parses, and even more so good EKBs, which provides the base for further reasoning (e.g., Gyori et al., 2017), are enabled by good customizations.

## Acknowledgements

This research was supported by the DARPA Big Mechanism program, under ARO contract W911NF-14-1-0391

# References

Allen, J.; de Beaumont, W.; Galescu, L.; and Teng, C.M. 2015. Complex Event Extraction using DRUM. In *Proceedings of BioNLP 2015*, 1-11. ACL-IJCNLP.

Allen, J.; Swift, M.; and de Beaumont, W. 2008. Deep semantic analysis of text. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, 343-354. Association for Computational Linguistics.

Allen, J., and Teng, C.M. 2017. Broad Coverage, Domain-Generic Deep Semantic Parsing. *AAAI Workshop on Construction Grammar*, March, Stanford University.

Bakhshandeh, O.; Wellwood, A.; and Allen, J. 2016. Learning to Jointly Predict Ellipsis and Comparison Structures. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 62–74, Berlin, Germany. ACL.

Banarescu et al. 2013. Abstract meaning representation for Sembanking. *In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 178-186. ACL.

Blackburn, P., and Bos, J. (2005), *Representation and Inference for Natural Language: A First Course in Computational Semantics*, CSLI Publications.

Berant, J.; Chou, A.; Frostig, R.; and Liang P. 2013. Semantic parsing on Freebase from question-answer pairs. *Empirical Methods in Natural Language Processing* (EMNLP).

Branavan, S.; Zettlemoyer, L.; and Barzilay, R. 2010. Reading between the lines: Learning to map high-level instructions to commands. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1268–1277. ACL.

Cai, S., and Knight, K. 2013. Smatch: An Evaluation Metric for Semantic Feature Structures, ACL.

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

Copestake, A.; Flickenger, D.; Pollard, C.; and Sag., I. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation* 3.

Das, D. (2014). Statistical models for frame-semantic parsing. In *Proceedings of the ACL*.

Earley, J. (1970) An efficient context-free parsing algorithm, *Comm. Of the ACM* 13, 2:94-102.

Fellbaum, C. ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Finkel, J.R.; Grenager, T.; and Manning, C. 2005. Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. *Proc. of the 43nd Annual Meeting of the Association for Computational Linguistics* (ACL 2005), pp. 363-370.

Gyori, B.M.; Bachman, J.A.; Subramanian, K.; Muhlich, J.L.; Galescu, L.; P.K Sorger, P.K. 2017. From word models to executable models of signaling networks using automated assembly. *Molecular Systems Biology* 13:954. doi: 10.2015252/msb.17765.

Hara, T.; Miyao, Y.; and Tsujii, J.. 2005. Adapting a probabilistic disambiguation model of an HPSG parser to a new domain. In *Proceedings of IJCNLP*.

Johnson, C., and Fillmore, C.J. 2000. The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics Conference*, pp. 56–62. Morgan Kaufmann Publishers Inc.

Kim, J.D.; Ohta, T.; Teteisi, Y.; and Tsujii, J. 2003. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics* 19:i180–i182.

Kim, J.-D., and Wang, Y. 2012. PubAnnotation: A persistent and sharable corpus and annotation repository. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing* (BioNLP '12), pp. 202–205, Stroudsburg, PA, USA.

Kipper, K.; Korhonen, A.; Ryant, N.; and Palmer, M., 2008. A large-scale classification of English verbs. *Language Resources and Evaluation*, *42*(1), pp. 21-40.

Klein, D.; and Manning, C.D. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems* 15 (NIPS 2002), Cambridge, MA: MIT Press.

Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; and McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations,* pp. 55-60.

Manshadi, M.; Gildea, D.; and Allen, J. A Notion of Semantic Coherence for underspecified Semantic Representations. *Computational Linguistics*, to appear.

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

Oepen, S.; Kuhlmann, M.; Miyao, Y.; Zeman, D.; Flickinger, D.; Hajicˇ, J.; Ivanova, A.; and Zhang, Y. 2014. SemEval 2014 Task 8: Broad-coverage semantic dependency parsing. In *Proc. of SemEval*. Dublin, Ireland.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1).

Pasupat, P., and Liang, P. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. ACL.

Perera, I.E., Allen, J.F., Galescu, L., Teng, C.M., Burstein, M.H., Friedman, S.E., McDonald, D.D. and Rye, J.M. (2017). Natural Language Dialogue for Building and Learning Models and Structures. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17),* pp. 5103-5104.

Rhee H., Allen J., Mammen J., and Swift M. (2014). Mobile phone-based asthma self-management aid for adolescents (mASMAA): a feasibility study. *Patient preference and adherence*, 2014:8(63-72).

Tellex, S.; Thaker, P.; Joseph, J.; and Roy, N. 2013. Learning Perceptually Grounded Word Meanings from Unaligned Parallel Data. *Machine Learning Journal*.

Toutanova, K., and Manning, C.D. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*.

Quick, D. and Morrison, C.T. (2017). Composition by Conversation. In *Proceedings of the 43$^{rd}$ International Computer Music Conference*, pp. 52-57.

Zhong, V.; Xiong, C.;and R Socher, R. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. arXiv:1709.00103.