University of Rochester

Thesis Proposal

# Corpus Annotation and Inference with Episodic Logic Type Structure

*Gene Louis Kim*

supervised by
Professor Lenhart K. Schubert

April 19, 2018

**Abstract**

A growing interest in tasks involving language understanding by the NLP community has led to the need for effective semantic parsing and inference. Modern NLP systems use semantic representations that do not quite fulfill the nuanced needs for language understanding: adequately modeling language semantics, enabling general inferences, and being accurately recoverable. This document proposes a plan to create a semantic parser of an initial form for a logic that balances these needs.

We present a plan for creating a high-precision semantic parser for an underspecified logical form (ULF) from annotated texts and which fits into further processing for ambiguity resolution. The semantic representation is grounded in Episodic Logic (EL) and ULF fully resolves the type structure with respect to EL while leaving further issues such as scope, word sense, and anaphora in a restricted, but unresolved, state. We hypothesize that a divide-and-conquer approach to semantic parsing will lead to higher quality semantic analyses by simplifying the problem, both from the perspective of researchers careful to handle linguistic phenomena appropriately and from the perspective of building an accurate semantic parser. In addition to creating this parser, this project aims to quantify the use of parsed ULFs for making inferences directly. ULFs enable structural inferences, including Natural Logic-like inferences, without further resolution, which are useful for downstream tasks such as dialogue and question answering.

# Contents

# Chapter 1

# Introduction

After many years of preoccupation with more modest goals, the computational linguistics and computational semantics communities are setting their sights on deeper language understanding, intelligent dialogue, and reasoning by machines. This is attended by a growing recognition of the need for *semantic parsing* as a fundamental task in automating these profoundly human capabilities. In a sense, semantic parsing is an elaboration of traditional syntactic parsing, providing at least a preliminary representation of the meanings underlying the surface syntax, and as such a starting point for deeper understanding and thus for generating relevant inferences or actions.

The goal of this project is to develop methods that enable accurate formalization of the semantic content of arbitrary English sentences. We will focus on the goal of generating type-coherent initial logical forms (so-called *unscoped logical forms* – ULFs), and this project seeks to demonstrate that ULFs both provide a stepping stone to full capture of sentential meanings and enable interesting classes of inferences that subsume Natural Logic (NLog) inferences.

A particularly auspicious development in *general* representations of semantic content has been the design of the *abstract meaning representation* (AMR) (Banarescu et al., 2013) followed by numerous research studies focused on generating AMR from English and on using it in tasks dependent on representation of the semantic content of text. The origins of AMR go back to the Penman text generation project of the 1980s (Bateman, 1990), but new inspiration was drawn from more recent opportunities for machine learning of semantic parsers, and for potential applications in machine translation. AMR is intended as a kind of intuitive normal form for the relational content of English sentences, for example factoring both of the sentences *"The man described the mission as a disaster"* and *"As he described it, the mission was a disaster"* into four typed entities that are presumed to exist: a describing event

$d$, a man $m$, a mission $m2$, and a disaster $d$, where the latter three are linked to the describing event $d$ by binary relations *arg0, arg1,* and *arg2* respectively.

However, given the limited goals of AMR, some phenomena were deliberately neglected, such as articles, tense, and the distinction between real and hypothetical entities. For example, *"The boy thinks he saw a ghost"* is represented in terms of a boy, a ghost, and a seeing event (all with the same existential status), where there is an *arg1* relation between the thinking event and the seeing event (which might be current or in the past or future). Apart from neglecting various phenomena, AMR treats many others in a very rough way, such as rendering modifications like *big rat* as two independent predicates, *big* and *rat*. This is probably good enough for language generation (NLG) and machine translation (MT): We can recover *big rat* from *is big and is a rat*, and *small elephant* from *is small and is an elephant*, without worrying that the NLG or MT system will declare that the rat is bigger than the elephant. In other words, AMR was not intended for inference, and is not well-suited to it. But to achieve deeper understanding we do need to worry about this, and inference is a central activity and requirement for understanding. We do not want a dialogue system or other NLU system to jump to unwarranted conclusions about the reality of ghosts or about rats that are bigger than elephants.

In a sense, the omissions and oversimplifications in AMR were the price paid to enable creation of AMR-annotated corpora of adequate size, and "fell-swoop" training of English-to-AMR parsers on these corpora. This is certainly a sound strategy for the purposes the designers had in mind, and work to date on parsing and using AMR is bearing this out. However, deep semantic processing leading to genuine understanding and inference is widely thought to require a divide-and-conquer approach that distinguishes multiple, interrelated goals; this includes assignment of semantic types to the logical-form constituents (predicates, names, determiners, predicate modifiers, relativizers, etc.), which determine how they can coherently combine; scoping of tense, quantifiers, and coordinators; resolution of anaphora; inference of event structure; disambiguation of word senses; recovery of elided or presupposed information; and conversion to some canonical form.

Our working hypothesis is that a practical starting point for a divide-and-conquer approach is the computation of a preliminary, surface-like logical form (ULF) that retains the full content of the original sentence, but resolves the semantic types and operator-operand structure of the constituents. We see the following advantages in this approach:

- In forming a training corpus for supervised machine learning, annotating sentences with logical forms that are close to the surface form should be relatively fast, reliable and loss-free, compared to annotations that require radical restructuring.

Our preliminary work on sentence annotation is providing confirming evidence for this hypothesis.

- Machine learning of the transduction from sentences to logical forms should be possible with a modest training corpus, again because the transductions are fairly close to an isomorphism from phrasal structure to ULF. As a quick preview, here is the ULF for *The boy wants to go"* (an example used in the AMR literature):
  `((the.d boy.n) ((pres want.v) (to go.v)))`.
- ULF allows a principled analysis of the structural relationships and processes that determine further transformations required for resolving scope ambiguities (for quantifiers, tense operators, coordinators) and coreference relations, inferring event structure, disambiguating word senses, and deriving a final canonical logical form. Much of the past work on episodic logic has addressed these issues. As a simple example of how ULFs can be successively disambiguated, the ULF mentioned above would become
  `(pres (the.d x (x boy.n) (x (want.v (to go.v)))))`
  after tense and determiner scoping, and this would be deindexed, Skolemized, and canonicalized (apart from word sense disambiguation) to
  `(|E|.sk at-about.p |Now17|),`
  `((the.d x (x boy.n) (x (want.v (to go.v)))) ** |E|.sk).`
  Here `|E|.sk` is a Skolemized episode variable, characterized by the sentence it is linked to via the '`**`' operator. (For the semantics of '`**`' see (Schubert, 2000).) The '`to`' operator forms a *kind of action* – see further discussion below. If we have a name for the boy via coreference, say Manolin, this would further become
  `(|E|.sk at-about.p |Now17|),`
  `((|Manolin| (want.v (to go.v))) ** |E|.sk).`
  Apart from the reification in `(to go.v)`, this is essentially a pair of first-order sentences, with the English-like wrinkle that the subject precedes the predicate.
- ULF itself, before any further transformations, already allows significant inferencing. Many of the inferences are similar to those enabled by *Natural Logic* (NLog), as implemented for example by MacCartney and Manning (MacCartney and Manning, 2008) and further developed subsequently. However, the kinds of inferences enabled by ULFs are broader in scope, and they can be generated spontaneously by forward inference. (In principle, this is possible for NLog inferences also, but uncertainties in the type structure of source sentences lead to the need for well-structured "target sentences" to be confirmed or disconfirmed.) Forward inference from language is an important capability, one that can show a degree of understanding independently of any particular application. Our techniques

can also be applied to existing tasks, such as the FraCaS inference challenge corpus (Cooper et al., 1996a) (premises and possible conclusions stated in English), including certain classes that were omitted in NLog trials.

- If ULFs can be computed with sufficient accuracy (as I expect), they could serve as a starting point for other semantically oriented NLP projects, such as extraction of general and specific "factoids" from text, or text summarization. Essentially ULFs would supply type-disambiguated, structured versions of the original sentences.

The remainder of this chapter outlines the structure of ULF, how ULF computation fits into a more comprehensive picture of semantic sentence analysis, and how ULFs, and ultimately deeper representations derived from them, can be used to make commonsense inferences. The following chapter will discuss related work before the last chapter which outlines the 3-stage research plan:

- the creation of an annotated corpus;
- the use of machine learning to develop a reliable English-to-ULF parser, and
- evaluation of the resulting parser; the internal evaluation will be much like that of AMR, and the external evaluation will include forward inference generation from sentences (via their computed ULFs).

## 1.1 Unscoped Logical Form, Deeper Semantic Analysis, and Inference

The idea of the ULF in our project was not developed in isolation, but rather as part of a more comprehensive approach to deriving deep, semantically coherent and inference-enabling representations of linguistic content, namely, Episodic Logic (Hwang, 1992; Hwang and Schubert, 1993; Schubert and Hwang, 2000). In this section I give a high-level description the form and meaning of ULFs followed by the role of ULFs in deeper understanding and the kinds of inferences enabled by ULFs and further resolved forms.

### 1.1.1 ULF type structure

The following six examples provide an idea of the language-like syntax of ULFs. The first two are from the Tatoeba database, the next three are from *The Little Prince* (which was used for the first AMR-annotated corpus), and the last is from the Web:

1. *Could you dial for me?*

   ```
   (((pres could.aux-v) you.pro ((dial.v {ref1}.pro) (adv-a (for.p me.pro)))) ?)
   ```

2. *If I were you I would be able to succeed.*

```
((if.c (I.pro (were-cf.v (= you.pro)))
  (I.pro ((pres would.aux-s) (be.v (able.a (to succeed.v)))))) \.)
```

3. *He neglected three little bushes*

```
(he.pro ((past neglect.v) (three.d (little.a (plur bush.n)))))
```

4. *Flowers are weak creatures*

```
((k (plur flower.n)) ((pres be.v) (weak.a (plur creature.n))))
```

5. *My drawing is not a picture of a hat*

```
((my.d drawing.n) ((pres be.v) not.adv-s (a.d (picture-of.n (a.d hat.n)))))
```

6. *Very few people still debate the fact that the earth is heating up*

```
(((fquan (very.adv-a few.a)) (plur person.n))
    (still.adv-s (debate.v
       (the.d (n+preds fact.n
                    (= (that ((pres prog)
                              ((the.d |Earth|.n) heat_up.v))))))))))
```

As can be seen, ULF structure quite closely reflects phrase structure; and the type tags of atomic constituents, such as `.pro, .v, .p, .a, .d, .n,` etc., are intended to echo the part-of-speech origins of these constituents, such as *pronoun, verb, preposition, adjective, determiner, noun,* etc., respectively. Originally, ULFs contained some $\lambda$-abstracts, for example to form a conjunctive predicate from postmodified nouns, but we have introduced syntactic sugar elements that relieve annotators from coding such abstracts. An example is seen in (6): The `n+preds` macro takes a noun and one or more predicates as complements, and these are expanded into a $\lambda$-abstracted conjunctive predicate in postprocessing. As a result, ULFs are relatively amenable to human creation and intuitive interpretation. Moreover, as mentioned in the Introduction, the proximity to surface structure enables NLog-like inference and more.

But then isn't parsing into ULF just another variant of syntactic parsing? The essential difference is that the type tags correspond to broad semantic categories (certain types of model-theoretic functions), and as such ensure that the type structure of ULFs – their operator-operand combinations – are semantically coherent. Richard Montague's profoundly influential work can be viewed as demonstrating the crucial importance of paying attention to the semantic types of words and phrases, and that doing so leads to a view of language as very close to logic; as a result it lends itself to inference, at least to the extent that we can resolve – or are prepared to tolerate – various forms of ambiguity, context-dependence and indexicality.

These semantic types are not as high-order as Montague's, nor as "rigid" as Montague's, but they suffice for maintaining type coherence. In particular, quantification is first-order, i.e., it iterates over individual entities, not over predicates, etc.

– though through reification of predicate meanings and sentence meanings, we can "talk about" kinds of things, kinds of actions, propositions, etc., not just ordinary objects.

As soon as we take semantic types seriously in ULFs like the above, we see that certain type-shifting operators are needed to maintain type coherence. For example, in sentence (1) the phrase *for me* is coded as `(adv-a (for.p me.pro))`, rather than simply `(for.p me.pro)`. That is because it is functioning here as a *predicate modifier*, semantically operating on the verbal predicate `(dial.v {ref1}.pro)` (*dial a certain thing*). Without the `adv-a` operator the prepositional phrase is just a 1-place predicate. Its use as a predicate is apparent in contexts like *"This puppy is for me"*. Note that semantically the 1-place predicate `(for.p me.pro)` is formed by applying the 2-place predicate `for.p` to the (individual-denoting) term `me.pro`. (Viewing $n$-place predicates as successively applied to their arguments, each time reducing the adicity, is in keeping with the traditions of Schönfinkel, Church, Curry, Montague, and others – hence "curried" predicates.) If we apply `(for.p me.pro)` to another argument, such as `|Snoopy|` (the name of a puppy), we obtain a truth value. So semantically, `adv-a` is a *type-shifting operator* of type (*predicate* → (*predicate* → *predicate*))), where the predicates are 1-place and thus of type (*entity* → *truth value*). Of course, the name `adv-a` is intended to suggest "adverbial", in recognition of the grammatical distinction between predicative and adverbial uses of prepositional phrases.

In the preceding discussion we glossed over *intensionality*. For example, (2) is a counterfactual conditional, and the consequent clause *"I would be able to succeed"* is not evaluated in the actual world, but in a possible world where the (patently false) antecedent is imagined to be true. ULF and deeper LFs derived from it are based on a semantics where sentences are evaluated in *possible situations (episodes)*, whose maxima are possible worlds. Details about syntactic forms and semantic types in episodic logic have been provided in many past publications (Hwang, 1992; Hwang and Schubert, 1994; Schubert and Hwang, 2000).

There are some further type-shifting operators in the examples: '`to`' (synonym: `ka`) in (2) shifts a verbal predicate to a *kind (type) of action or attribute*, which is an abstract individual; '`k`' in (4) shifts a nominal predicate to a *kind* of thing (so the subject here is the abstract kind, flowers, whose instances consist of sets of flowers; and '`that`' in (6) produces a reified *proposition* (again an abstract individual) from a sentence meaning. Through these type shifts, we are able to maintain a simple, classical view of predication, while allowing greater expressivity than the most widely employed logical forms, for example enabling generalized quantification (as in (6)), modification, reification, and other forms of intensionality.

The positioning of `(adv-a (for.p me.pro))` within the verbal predicate it modi-

fies, rather than in the expected prefix-operator position, already indicates a certain looseness in the ULF syntax, as opposed to the rigidity of formal logic. This is unproblematic because we restrict the way operators may combine with operands so that type consistency is assured – and in fact in subsequent processing, any `(adv-a ...)` constituents of a verbal predicate are moved so as to immediately precede that predicate. There are a number of further kinds of looseness in ULFs, but we defer further discussion to the current ULF annotation tutorial.

### 1.1.2 Role of ULF in comprehensive semantic interpretation

ULFs are underspecified – loosely structured and ambiguous – in several ways. But their surface-like form, and the type structure they encode, make them well-suited to reducing underspecification, both using well-established linguistic principles and machine learning (ML) techniques that exploit the distributional properties of language. The scope of the thesis is not expected to encompass much of this further processing, but we want to reiterate some reasons for regarding ULFs as a suitable basis.

Heuristic algorithms that resolve scope ambiguities and make event structure explicit have been developed for and applied to ULF in the past. Though these algorithms are not sufficiently reliable, they set a baseline for future work on disambiguation aided by ML techniques. The following points address the utility of ULFs as preliminary structures enabling systematic reduction of underspecification.

**Word sense disambiguation (WSD):** One obvious form of underspecification is word sense ambiguity. But while, for example, `(weak.a (plur creature.n))` in (4) does not specify which of the dozen WordNet senses of *weak* or three senses of *creature* is intended here, the type structure is perfectly clear: A predicate modifier is being applied to a nominal predicate. Certainly standard statistical WSD techniques (Jurafsky and Martin, 2009) can be applied to ULFs, but this should not in general be done for isolated sentences, since word senses tend to be used consistently over longer passages. We should mention here that adjectives appearing in predicative position (e.g., *able* in (2)) or in attributive position (e.g., *little* in (3)) are type-distinct, but ULF leaves this distinction to automatic processing, since the semantic type of an adjective is unambiguous from the way it appears in ULF.

**Predicate adicity:** A slightly subtler issue is the adicity of predicates. We do not assume unique adicity of word-derived predicates such as `run.v`, since such predicates can have intransitive, simple transitive and other variants (e.g., *run quickly* vs. *run an experiment*). But adicity of a predicate in ULF is always clear from the syntactic context in which it has been placed – we know that it has all its arguments in place, forming a truth-valued formula, when an argument (the "subject") is placed

on its left, as in English.

**Scope ambiguity:** While some of the underspecification in ULFs is deterministically resolvable, *unscoped* constituents can generally "float" to more than one possible position. The three types of unscoped elements in ULF are *determiner phrases* derived from noun phrases (such as *very few people* and *the Earth* in (6)), the tense operators `pres` and `past`, and the coordinators `and.cc, or.cc` and some variants of these. The positions they can "float" to in postprocessing are always pre-sentential, and determiner phrases leave behind a variable that is then bound at the sentential level. This view of scope ambiguity was first developed in (Schubert and Pelletier, 1982) and subsequently elaborated in (Hurum and Schubert, 1986) and reiterated in various publications by Hwang and Schubert. The accessible positions are constrained by certain restrictions well-known in linguistics. For example, in the sentence *"Browder ... claims that every oligarch in Russia was forced to give Putin 50 percent of his wealth"*, there is no wide-scope reading of *every*, to the effect *"For every oligarch in Russia, Browder claims ... etc."*; the subordinate clause is a "scope island" for strong quantifiers like *every* (as well as for tense). The important point here is that ULF allows exploitation of such structural constraints, since it still reflects the surface syntax. Now, firm linguistic constraints still leave open multiple scoping possibilities, and many factors influence preferred choices, with surface form (e.g., surface ordering) playing a prominent role (Manshadi et al., 2013). So again the proximity of ULF to surface syntax should be helpful in applying ML techniques to determining preferred scopings.

**Anaphora:** Another important aspect of disambiguation is coreference resolution. Again there are important linguistic constraints ("binding constraints") in this task. For example, in *"John said that he was robbed", he* can refer to John; but this is not possible in *"He said that John was robbed"*, because in the latter, *he* C-commands *John*, i.e., in the phrase structure of the sentence, it is a sibling of an ancestor of *John*. ULF preserves this structure, allowing use of such constraints. Preservation of structure also allows application of ML techniques (Poesio et al., 2016), but again this should be done over passages, not individual sentences, since coreference "chains" can span many sentences. When coreference relations have been established as far as possible and operators have been scoped, the resulting LFs are quite close in form to first-order logic, except for incorporating the additional expressive devices (generalized quantifiers, modification, attitudes, etc.) that we have already mentioned and illustrated. In writings on episodic logic this is called this the *indexical logical form*, or ILF.

**Event/situation structure:** The most important aspect of logical form that remains implicit in ILF is event/situation structure. Much of the past work on EL

has been concerned with the principles of *de-indexing*, i.e., making events and situations – *episodes* in EL terminology – explicit (Hwang, 1992; Hwang and Schubert, 1994; Schubert, 2000). The relationship to Davidsonian event semantics and Reichenbachian tense-aspect theory is explained in these references. Our compositional approach to tense-aspect processing leads to construction of a so-called *tense tree*, and yields multiple, related reference events for sentences such as *"By 8pm tonight, all the employees will have been working for 15 hours straight".* The relevant point here is that the compositional constuction and use of tense-trees is possible only if the logical form being processed reflects the original clausal structure – as ULF and ILF indeed do.

**Canonicalization:** Finally, canonicalization of ELF into "minimal" propositions, with top-level Skolemization (and occasionally $\lambda$-conversions), is straightforward. A simple example was seen in the Introduction, and some more complex examples are shown in prior publications (Schubert and Hwang, 2000; Schubert, 2014; Schubert, 2015).

When episodes have been made explicit (and optionally, canonicalized), the result is *episodic logical form* (ELF); i.e., we have sentences of EL, as described in our previously cited publications. These can be employed in our Epilog inference engine for reasoning that combines linguistic semantic content with world knowledge. A variety of complex Epilog inferences are reported in (Schubert, 2013), and (Morbini and Schubert, 2011) contained examples of self-aware metareasoning. Further in the past, Epilog reasoned about snippets from the Little Red Riding Hood story: *If the wolf tries to eat LRRH when there are woodcutters nearby, what is likely to happen?*"; answer chain: *The wolf would attack and try to subdue LRRH; this would be noisy; the woodcutters would notice, and see that a child is being attacked; that is a wicked act, and they would rush to help her, and punish or kill the wolf* (Hwang, 1992; Schubert and Hwang, 2000). However, the scale of such world-knowledge-dependent reasoning has been limited by the difficulty of acquiring large amounts of inference-enabling knowledge. (The largest experiments, showing the competitiveness of Epilog against state-of-the art theorem provers were limited to formulas of first-order logic (Morbini and Schubert, 2009).) In the proposed work we therefore focus on inferences that are important but not heavily dependent on world knowledge.

Thus ULFs comprise a "primal" logical form whose resemblance to phrase structure and whose constraints on semantic types provide a basis for the multi-faceted requirements of deriving less ambiguous, nonindexical, canonical LFs suitable for reasoning. However, as we have pointed out, ULFs are themselves inference-enabling, and this will be important for the evaluation plan.

### 1.1.3 Inference with ULFs

An important insight of NLog research is that language can be used directly for inference, requiring only phrase structure analysis and upward/downward entailment marking (polarity) of phrasal contexts. This means that NLog inferences are *situated* inferences, i.e., their meaning is just as dependent on the utterance setting and discourse state as the linguistic "input" that drives them.

This insight carries over to ULFs, and provides a separate justification for computing ULFs, apart from their utility in the process of deriving deep, context-independent, canonicalized meaning representations from language. Our evaluation of the English-to-ULF parser that we are proposing to develop will be formulated in terms of certain classes of situated inferences enabled by ULFs.

ULFs in principle provide a more reliable and more general basis for situated inference than mere phrase structure, because of the coherent semantic type structure they encode. Greater reliability also leads to the possibility of spontaneous forward inferencing, as opposed to inference guided by propositions to be confirmed or disconfirmed (as in most textual entailment and NLog studies to date). This is important, because human language understanding seems to involve continuous forward inferencing. As a simple example, if according to your paper or newsfeed *"Police reported that the vehicle struck several cars"*, you will conclude that the reported event almost certainly happened, and further, that the cars involved were all damaged. Now, the first of these inferences requires only a small amount of knowledge about communication, to the effect that *reporting* (in your preferred media) typically involves reporting of facts; whereas the latter depends on very specific world knowledge. Our demonstration of ULF utility in forward inference will focus on the former kinds of inference (and related ones), since accumulation of sufficient world knowledge for enabling the latter kinds of inference remains out of reach in the short run.

Here, briefly, are some kinds of inferences we can expect ULFs to support:

- *NLog inferences based on generalizations/specializations.* For example, *"Every NATO member sent troops to Afghanistan"*, together with the knowledge that France is a NATO member and that Afghanistan is a country entails that *France sent troops to Afghanistan* and that *France sent troops to a country.* Such inferences are within the scope of NLog-based and ULF-based methods, and can help in finding inference paths to candidate entailments; but they will not be our focus as they rarely seem worthwhile as spontaneous forward inferences from sentences in discourse (we are particularly interested in dialogue settings).
- *NLog inferences based on implicatives.* For example, *"She managed to quit smoking"* entails that *"She quit smoking"* (and the negation of the premise leads to the opposite conclusion). We already demonstrated such inferences in our framework

for various headlines (Stratos et al., 2011), such as the inference from *Oprah is shocked that Obama gets no respect* (Fox News 2011) to *Obama gets no respect*. Such inferences are surely important – and immediate – in language understanding, and will be included in our evaluations.

- *Inferences based on attitudinal and communicative verbs.* Some such inferences, for instance for *knowing-that* and *finding-out-that*, fall under the previous heading, but others do not. For example, *"John denounced Bill as a charlatan"* entails that *John probably believes that Bill is a charlatan*, that *John asserted to his listeners (or readers) that Bill is a charlatan*, and that *John wanted his listeners (or readers) to believe that Bill is a charlatan*. Such inferences would be hard to capture within NLog, since they are partially probabilistic, require structural elaboration, and depend on constituent types.
- *Inferences based on counterfactuals.* For example, *"If I were rich, I would pay off your debt"* and *"I wish I were rich"* both implicate that *the speaker is not rich*. This depends on recognition of the counterfactual form, which is distinguished in ULF.
- *Inferences from questions and requests.* For example, *"When are you getting married"* enables the inference that the addressee will get married (in the foreseeable future), and that the questioner wants to know the expected date of the event, and expects that the addressee probably knows the answer, and will supply it. Similarly an apparent request such as *"Could you close the door?"* implies that the speaker wants the addressee to close the door, and expects he or she will do so. There are subtleties in the distinction between questions and requests that can be captured in ULF and made use of.

# Chapter 2

# Background & Related Work

In the Introduction, we related our proposal to the development of AMR. Despite its lack of concern for inference, this development was an inspiration to us in terms of both the quest for broad coverage and methods of learning and evaluating semantic parsers. But there has also been much activity in developing semantic parsers that derive logical representations, raising the possibility of making inferences with those representations (Popescu et al., 2004; Kate and Mooney, 2006; Kwiatkowski et al., 2011; Liang et al., 2011; Poon, 2013; Tellex et al., 2011; Artzi and Zettlemoyer, 2013; Howard et al., 2014; Artzi et al., 2015; Konstas et al., 2017). The techniques and formalisms employed are interesting (e.g., learning of CCG grammars that generate $\lambda$-calculus expressions), but the targeted tasks have generally been question-answering in domains consisting of numerous monadic and dyadic ground facts ("triples"), or instruction-following by robots. Acquisition of knowledge via language, or inferences about beliefs, intentions, etc., have generally not been addressed in any broad way.

Noteworthy examples of formal logic-based approaches, not targeting specific applications were done by Bos (2008) and Draiccio et al. (2013), whose hand-built semantic parsers respectively generate FOL formulas and OWL-DL expressions. But again these representations preclude generalized quantifiers, modification, reification, attitudes, etc., and the route from NL strings to logic (CCG parsing $\rightarrow$ DRS $\rightarrow$ FOL, in the case of Bos' Boxer) often produces flawed representations. We are not aware of any work on inference generation of the type we are targeting, based on these projects.

The rest of this chapter describes and discusses notable work in domain-general semantic parsing and annotation. Each one of these projects has had significant influences on the choices made in the design of this research project. This more detailed review sets the stage for concretely discussing choices made in our research

plan and distinguishing the direction of our research plan from existing work.

## 2.1 TRIPS

The TRIPS parser generates semantic parses in an underspecified semantic representation with scoping constraints (Allen and Teng, 2017; Allen et al., 2018). The nodes are grounded in an ontology, which is a single inheritance hierarchy built on a mixture of syntactic and semantic distinctions. There are levels of the ontology that roughly correspond to VerbNet (Schuler, 2006) and FrameNet (Baker et al., 1998) classes, and also include additional distinctions of temporal and entailment information. It has three equivalent formats, which roughly correspond to the three different types of formats that AMR comes in: logical, graphical, and PENMAN. Figure 2.1 shows an example of a TRIPS parse in graphical format.

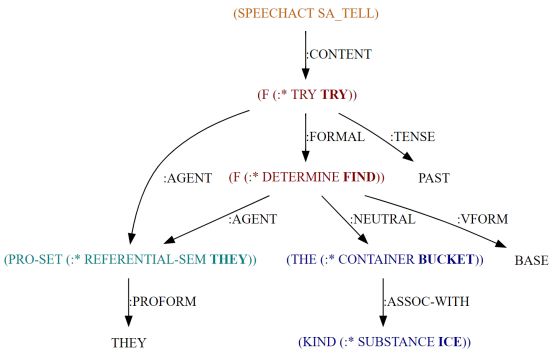



Figure 2.1: Parse for "They tried to find the ice bucket" using the vanilla dialogue model of TRIPS.

Despite the structural similarity to AMR, TRIPS includes richer information for quantifiers and speech acts and makes finer distinctions of word senses for adjectives and predicates. The TRIPS parser generates LFs using a bottom-up chart parser with a hand-built grammar, a syntax-semantic lexicon, and a semantic ontology. The parser's preferences can be further tuned using the results from tokenizers, POS taggers, NER taggers, and constituent parsers (Allen et al., 2018).

One of the greatest strengths of TRIPS is its ontology of English words which gives a unified relationship between words using semantic roles, aspectual class, and temporal/causal entailments. The ontology is critical in both parsing and inference. During parsing the ontology is used for semantic preferences in argument selection and word sense disambiguation. During inference the ontology can be used for generalizing and specializing words, making inferences about time and causality, and for making structured semantic similarity judgments between predications.

The TRIPS parser has been deployed in multiple tasks with minimal modifications: extracting knowledge from biology papers (Allen et al., 2015), conversational dialogue (Rhee et al., 2014), goal-oriented dialogue (Perera et al., 2017), and lexical knowledge acquisition (Allen et al., 2011).

### 2.1.1 TRIPS LF

Allen et al. (2008) describe a graphical logical form representation derived from the representation used by the TRIPS dialogue system (Allen et al., 2007).[1] Each node in the LF has three components: *specifier*, *type*, and *word* which represent the node function, conceptual class, and surface word, respectively. The specifiers are semantic functions enumerated in the LF specification and the type is a conceptual class from the ontology used in conjunction with the LF. The nodes are connected using thematic-role-inspired relations (e.g. `:AGENT`, `:AFFECTED`, `:FORMAL`, etc.). For example, consider the top-level node in Figure 2.1, `(SPEECHACT SA_TELL)`. `SPEECHACT` is the specifier, `SA_TELL` is the type, and the word is empty because this top-level is generated from the general sentence structure assuming deployment in a dialogue setting. It participates in one relation `:CONTENT` with the node `(F (:* TRY TRY))` as the parent of the relation.

TRIPS LF descriptively captures modal constructions, generalized quantifiers, lambda abstractions, and dialogue semantics making it able to express a large subset of natural language phenomena. These features are very useful when using TRIPS LFs since it allows a system to distinguish differences in these finer details.

Formally, TRIPS LF is an underspecified semantic representation which subsumes well-formed Minimal Recursion Semantics (MRS) and Hole Semantics (Allen et al., 2018). All of these are object-language agnostic, meta-level semantic representations (Copestake et al., 2005; Bos, 1996). They are very useful for managing underspecified, but constrained representations in computationally simple and efficient manners. The problem with working only on model-agnostic semantics is that without grounding these representations into an object language, model-theoretic notions such as interpretation, truth, satisfaction, and entailment cannot be applied. It follows then that one cannot make claims about soundness or completeness about inference systems built directly on these representations.

One can ground these model-agnostic representations into an object language, but must be careful in their choice of object language. In a project extracting lexical knowledge from WordNet, Allen et al. (2013) mapped TRIPS LF to OWL-DL (OWL Working Group, 2004) from which subsumption-based inference was performed between WordNet definitions. While expressivity limitations did not arise in the subset of WordNet encountered in the project, OWL-DL struggles with common linguistic phenomena such as non-intersective modification, reification, self-reference, and uncertainty. I expect that generalizing the project to more of WordNet without changing the choice of representation would run into these fundamental limitations

[1] The full documentation for the LF is available at `http://trips.ihmc.us/parser/LF%20Documentation.pdf`.

of OWL-DL. Of course, TRIPS doesn't need to be grounded into OWL-DL specifically. Though it would probably be a trickier endeavor, mapping TRIPS into a more expressive object language seems possible.

All this does not mean that inference systems that operate directly on model-agnostic representations cannot be useful. In fact, I've already mentioned instances where TRIPS LF has been used to great effect. We simply cannot make model-theoretic claims about the inference until the object language is defined. I will not point out these sorts of limitations in as much detail in the following works, but it should be noted that these criticisms apply just as much to following work that have insufficiently expressive or model-agnostic semantic representations.

## 2.2 The JHU Decompositional Semantics Initiative

Decomp[2] follows an approach to building up a model of language semantics by modeling every day language user annotations on focused phenomena. The idea is to pose semantic distinctions as questions that are quick and easy to judge by every day users. This allows the construction of a large corpora from which to learn models of human language judgments. Then by building these models up for more and more semantic distinctions one can construct a general model of language semantics.

Much of the work in Decomp is built on top of PredPatt (White et al., 2016), a predicate-argument extractor built on top of universal dependencies. These are very low-level predicates, just above the surface language. See the example below (taken from the PredPatt github page[3]) which shows the predicate-argument structures that PredPatt outputs.

```
PredPatt extracts predicates and arguments from text .
?a extracts ?b from ?c
    ?a: PredPatt
    ?b: predicates
    ?c: text
?a extracts ?b from ?c
    ?a: PredPatt
    ?b: arguments
    ?c: text
```

Notice that the predicates include prepositions in them so they are not decomposed in the way semantic representations usually are. The goal was to provide a

[2] https://www.decomp.net

[3] https://github.com/hltcoe/PredPatt

high-precision interface between syntax and semantics for reliable usage in downstream semantic tasks or further resolution. Zhang et al. (2017) showed that PredPatt has the best precision and recall among well-known OpenIE systems on a large scale benchmark based on PropBank.

Rudinger et al. (2018) use these methods to predict event factuality with surprising precision. This task has some overlap with the sorts of inferences that we plan to show with ULFs, though this work does not frame the evaluation as a forward-inference task. Still, the event factuality dataset released alongside this publication likely can be of use for our evaluations.

Though this initiative has many more interesting results associated with it, they are not directly relevant to the discussion of this proposal so I will stop my discussion here. The long term goals of this project aligns well with ours in generating high-fidelity semantics for language which enables commonsense reasoning. To paraphrase Ben Van Durme's comment during the CogSci Dinner on April 16th, he is trying to climb up the mountain of semantics one step at a time through modeling human judgments whereas I am trying to climb down the mountain of semantics from episodic logic to a tractable subset for feasible annotation and precise generation. I am very sympathetic to the efforts of this initiative. At the moment, however, the semantic descriptions from this project are completely decoupled from the model-theoretic basis which drives much of this proposal.

## 2.3 Groningen/Parallel Meaning Bank

The Groningen Meaning Bank (GMB) (Basile et al., 2012) was developed at the University of Groningen led by Johan Bos and annotates full documents with Discourse Representation Structures (Kamp, 1981). Their annotations combine linguistic information from a variety of sources for different levels of semantic annotation, such as the thematic roles from VerbNet (Schuler, 2006) and word sense from WordNet (Miller, 1995). GMB uses an annotation approach that they call *human-aided machine annotation*, where the annotation process first is automated then corrected by humans. This work was recently extended into the Parallel Meaning Bank (PMB) (Abzianidze et al., 2017) where they seek to extend this type of annotation to multiple languages with semantic projection across sentence translations. I will focus my discussion on the English portion of the PMB since it is an improved version of GMB.

### 2.3.1 Annotation Pipeline

The annotation pipeline is separated into layers from tokenizing to sentence and discourse level semantics that build on top of each other. These layers are automatically

generated, then corrected by annotators to obtain gold-annotations. As annotations are accumulated, they are used to retrain the automatic systems.

### 2.3.2 Layers of Annotation

The annotation is split into the following five layers: (1) segmentation, (2) syntactic analysis, (3) semantic tagging, (4) symbolization, and (5) semantic interpretation. Steps (2-4) can be performed concurrently, but the output of step (1) feeds into steps (2-4) and the outputs of steps (2-4) feed into step (5). Below I describe each step in more detail.

1. **Segmentation**
   A custom-built character-level IOB segmentation tagger (Evang et al., 2013) segments the text to a representation closely-tied to PMB semantics. For example, semantically transparent morphological structure is decomposed (e.g. *impossible* → *im* + *possible*) and multiword expressions which are opaque or part of a name are singly tokenized – *Las Vegas* and *2 pm* are both single tokens.[4]
2. **Syntactic Analysis**
   EasyCCG (Lewis and Steedman, 2014) generates Combinatory Categorical Grammar (Steedman, 2000) derivations for PMB's syntactic analysis.[5]
3. **Semantic Tagging**
   Deep residual networks (Bjerva et al., 2016) perform lexical semantic tagging on an 80 tag tagset. This tagset includes POS tags, named entity classes, semantic distinctions (e.g. `negation`, `equative`), and some discourse-level categorizations (e.g. `greeting`, `hesitation`).
4. **Symbolization**
   This step maps tokens to lemmatized and normalized non-logical, conceptual categorizations that are referred to during semantic interpretation to improve generalization. It performs general simplifications such as reducing *European* to `europe`, but its most important function is canonicalizing special purpose formats that occur in natural language such as dates and time (e.g. *2 pm* is symbolized to `14:00`). PMB currently relies on a rule-based lemmatizer, Morpha (Minnen et al., 2001) for this step.

[4]During an invited talk at the 11th Linguistic Annotation Workshop (2017) Johan admitted that trying to decompose phrases such as "Secretary of State" proved too difficult in GMB annotation.

[5]EasyCCG's independence from POS tags and lexicalized grammar turn out to be critical for integrating with PMB's custom tags and modular human corrections, respectively. These features are also desirable from a cross-lingual annotation perspective.

5. **Semantic Interpretation**
   Finally, the text is interpreted into Discourse Representation Structures (DRSs) from the CCG analysis with lexical meaning represented by the 3-tuple <CCG category, semantic tag, symbol> (outputs of steps (2-4)). The Boxer system (Bos, 2015), a hand-built semantic parser, compositionally constructs the semantic interpretations.

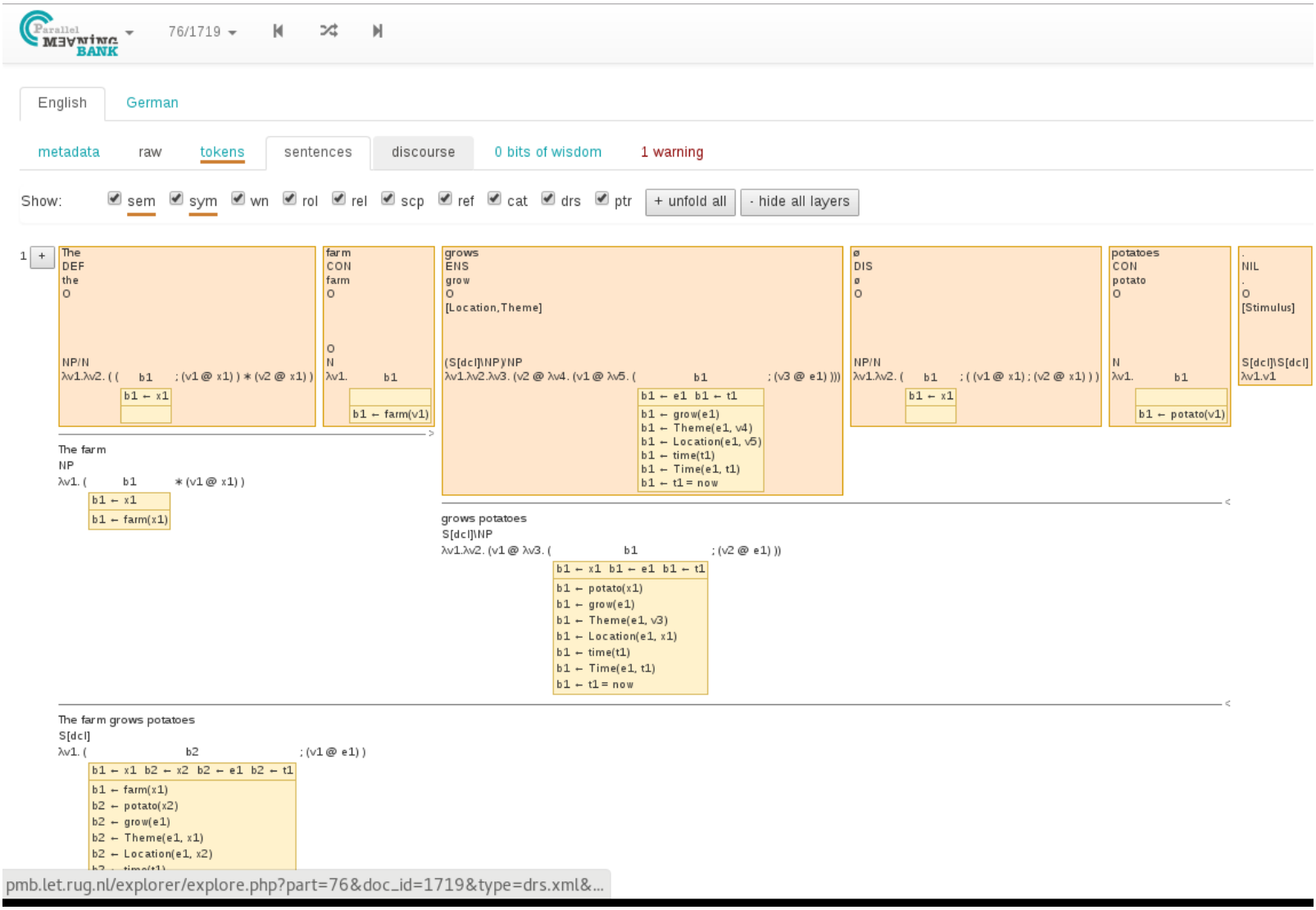



Figure 2.2: Screenshot of the PMB Explorer with analysis of the sentence "The farm grows potatoes."

### 2.3.3 PMB Explorer

A highly-featured annotation website was developed to assist in PMB's structured annotation approach. A screenshot of the PMB Explorer displaying an automatic analysis is shown in Figure 2.2. Below are notable features of the PMB Explorer:

- A modular, layer-wise annotation view – each can be marked correct separately.
- Correction tracker, revision history, and reversion.
- An integrated bug-tracker for annotator organization and communication.

### 2.3.4 Annotated Texts

The texts selected for annotation in the PMB are mostly freely distributable and total in 11.3 million tokens and 285,154 documents (Abzianidze et al., 2017). They include the following sources:

- Totoeba[6]
- Newscommentary corpus (Tiedemann, 2012)
- Recognizing Textual Entailment (RTE) corpus (Giampiccolo et al., 2007)
- Sherlock Holmes stories[7]
- The Bible (Christodouloupoulos and Steedman, 2015)

Table 2.1 shows the statistics of the gold annotations from the most recent (and first) release.[8] Note that there are three other languages that are annotated in this release as well. While no inter-annotator agreement is reported, the Explorer allows annotation revisiting and correction between annotators so the gold-annotations are likely highly uniform and accurate.

| Documents | Sentences | Tokens |
|---|---|---|
| 2,049 | 2,057 | 11,664 |

Table 2.1: Statistics of annotated English sentences in PMB v1.0.0, December 22, 2017 release.

### 2.3.5 Discourse Representation Structures

Discourse Representation Structure (DRSs) is a representation developed by Hans Kamp (Kamp, 1981; Kamp and Reyle, 1993) to handle anaphora, discourse structure, and presupposition. The key anaphoric issue that DRS overcame is "donkey

[6] https://tatoeba.org

[7] http://gutenberg.org, http://etc.usf.edu/lit2go

[8] http://pmb.let.rug.nl/data.php

anaphora" (e.g. "Every child who owns a dog loves it."). DRT overcomes the challenges of determining the quantifier interpretation and scope in such examples with principles that determine the nature of anaphoric bindings.

DRSs can also generate inferences via a mapping to FOL. The ability to perform this mapping, however, means that the expressivity of DRT is equal to that of FOL. FOL itself is not able to handle many important semantic phenomena such as generalized quantifiers and non-modal intensionality. There don't seem to be any methods in PMB to handle this limitation based on my conversations with Johan. In fact, he disclosed that PMB uses the `NIL` semtag to annotate generalized quantifiers as semantically ambiguous.

## 2.4 LinGO Redwoods Treebank

The LinGO Redwoods Treebank is a syntacto-semantic treebank with annotations of Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994) with corresponding Minimal Recursion Semantics (MRS) (Copestake et al., 2005). This long-running project has developed a suite of resources which support the annotation project. This includes LinGO English Resource Grammar (ERG) (Bub et al., 1997), a hand-built HPSG-grammar, a fast grammar parser (Callmeier, 2001), and the `[incr tsdb()]` grammar profiling and annotation environment (Oepen, 2001).[9]

### 2.4.1 Redwoods Treebank Overview

The Redwoods Treebank is annotated in ERG and has grown into a topically-varied collection of corpora over the years. The most recent release, the Eighth Growth, was released in 2013 and consists of

- the Verbmobil dialogue corpus,
- the LOGON Norwegian-English MT corpus (Oepen et al., 2004),
- the WeScience initiative English Wikipedia subset of 100 articles (Ytrstøl et al., 2009),
- a subset of the Brown corpus semantically tagged in the SemCor project (Landes et al., 1998),
- 23 of the 25 sections of the Penn Treebank Wall Street Journal corpus (DeepBank) (Flickinger et al., 2012),
- and a smattering of smaller sets of text such as the FraCaS entailment dataset (Cooper et al., 1996b) and hand-crafted linguistic unit test sentences.

[9] All the tools can be found at `http://www.delph-in.net`.

This release totals in over 92,706 sentences and 87,821 parses, giving it 95% coverage (not all parses are verified correct). The project has incomplete coverage due to the annotation method relying on a grammar to produce all the annotation choices.

### 2.4.2 Minimal Recursion Semantics (MRS)

MRS is a flat semantic representation which uses handles to encode scope effects and ambiguity. Its semantics is composed via unification by design since it was developed as a formalization of semantic composition in typed feature structure grammars (Copestake et al., 2001). In an effort to be an effective computational semantics, MRS is designed to satisfy the following criteria (Copestake et al., 2005):

- *Expressive Adequacy* – correctly express the full range of linguistic meanings
- *Grammatical Compatibility* – link cleanly to syntactic analysis;
- *Computational Tractability* – enable efficient processing of meaning; such as equivalence checking, and represent meaning in a computationally simple manner;
- *Underspecifiability* – allow the representation to leave ambiguous distinctions unresolved, which can be resolved in a simple and flexible manner.

Copestake et al. emphasize the last two criteria having found that previous work by computational linguists lack in those areas. Copestake et al. (1995) explains that MRS is not a complete model semantic theory, rather a representation for describing semantic structures in relation to HPSG. Thus it's unclear how we can make judgments about the formal semantic expressivity of MRS in relation to ERG at all. By design, then, MRSs are descriptively powerful while lacking a definite semantic representation and inference mechanism rooted in model semantics.

### 2.4.3 Annotation Procedure

Similar to the Parallel Meaning Bank, the Redwoods Treebank relies on human-guided machine annotations. The annotation procedure begins with generating the 500 highest-ranked analyses from a hand-built HPSG grammar (Flickinger et al., 2012). The correct analysis, if any, is manually selected using a series of decisions of whether to include a candidate lexical or relational analysis. For example, `_see_v_1<2:5> TENSE past` which describes the analysis of the text span (2,5), "saw", as the first sense of the verb "see" (`_see_v_1`) with a past tense. The annotator then makes a decision as to whether this analysis should be included or excluded from the final form. See the screenshot in Figure 2.3 for the environment state when only two candidate trees remain (there are no more trees below the frame) with over a dozen discriminants that the annotator can choose from to identify the correct analysis.

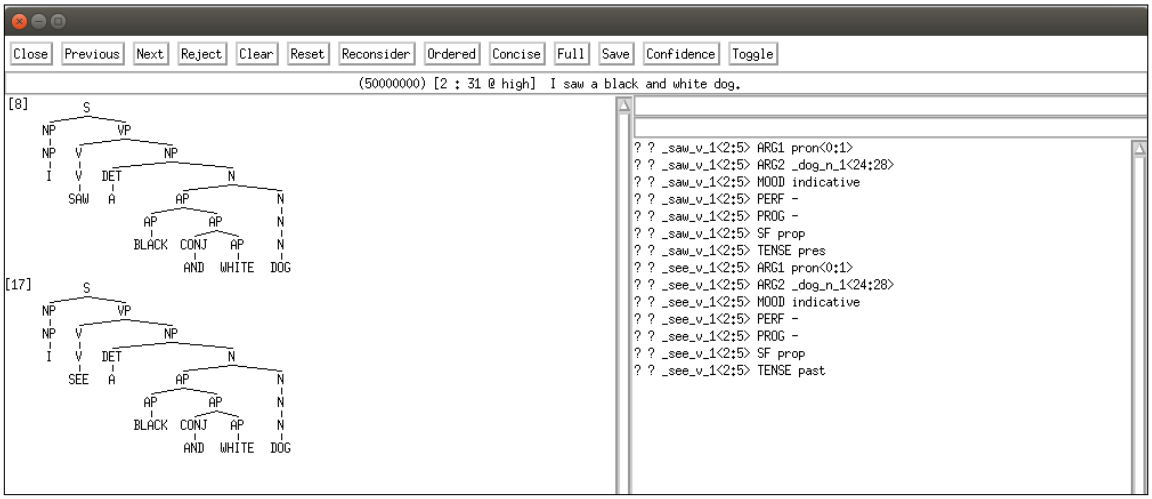



Figure 2.3: Screenshot of Redwoods treebanking environment. The left pane displays constituent parse trees of the remaining analyses to be disambiguated. The right pane displays the list of discriminants.

The `[incr tsdb()]` tool automatically prunes impossible analyses given the decision. The annotator decisions are saved so that upon updating the grammar, the annotation decisions can be rerun to identify and re-annotate ambiguities that arise from the changes in the grammar. This turns out to greatly reduce the re-annotation burden. Flickinger et al. (2012) report that this method required re-annotating 5-10% of the corpus in addition to the newly covered sentence when the grammar is improved. The limitation of this approach is that it assumes that the annotator was correct to begin with, so this cannot so easily be used to re-annotate the treebank efficiently to correct large-scale annotation mistakes identified during an error analysis.

### Challenges of a Hand-built Grammar

The annotation method of Redwoods Treebank is highly efficient but relies on a hand-built grammar. This grammar building turns out to take a long time as seen by the changes in coverage reported over the years of this project. The coverage statistics for the most recent release of the Redwoods Treebank is shown in Table 2.2. It is quite impressive with an 87% verified coverage on several domains. Still, 13% is a considerable portion of text to leave unanalyzed. In 2004 Baldwin et al. (2004) analyzed ERG coverage on a sample of the British National Corpus (BNC) (Burnard, 2000). This is four years after the first paper describing ERG (Copestake and Flickinger, 2000). The results of the analysis can be summarized as follows:

| Segment | # of sentences | Raw coverage | Verified coverage |
|---|---|---|---|
| Verbmobil | 12,393 | 96.46% | 92.18% |
| WSJ (00-21) | 43,541 | 93.87% | 85.17% |
| WeScience | 11,558 | 91.93% | 80.75% |
| SemCor | 3,000 | 93.81% | 85.11% |
| Total | 92,706 | 94.73% | 87.28% |

Table 2.2: ERG results for the most recent Redwoods release (the Eighth Growth). Only a selection of segments (corpora) are included in this table, but the total results includes all sentences in the growth.

- 32% of BNC sample have complete ERG lexical coverage
- Of those, 57% ERG could generate a parse for (18% of total sample)
- Of those, 83% contained a correct parse (15% of total sample)

When the results are broken down one by one the results don't look so bad, but when the errors are propagated ERG it is clear that ERG is performing dismally. This experiment shows the amount of work that it takes to hand-build a reliable grammar. A few years into development it could only generate a correct analysis for 15% of a sampling from the BNC. When developed for over a decade, a hand-built grammar can perform pretty well, but it is not a task that one should add to their project lightly. In order to overcome the 13% coverage gap of ERG, Zhang and Krieger (2011) developed a PCFG approximation of ERG which uses ERG rule names with feature structures and syntactic context as PCFG categories and a heuristic unifier to force unification. This PCFG parsed over 99% of sentences a 2% reduction in F1. This has not been used in the annotation process so far.

**Discussion**

The Redwoods Treebank project has many engineering successes that are reflective of the length of time that it has been under development and the great number of people collaborating on it. The successes of the project in flexibility of analysis, particularly in syntax, also reflects the motivations described by early papers to overcome the limitations of annotations that focus on one type of information (e.g. phrase structure trees, semantic dependency graphs) with a fixed set of labels while losing information that could be gained concurrently in an integrated manner.

However, the lack of an inference mechanism makes it suboptimal for generating inferences. They recommend using FOL theorem provers to generate inferences which would limit their semantic expressivity to FOL. MRS use in applications so far

have relied on task-specific, *ad hoc* methods: tree transformations for question generation (Yao and Zhang, 2010), semantic mapping rules for semantic-transfer machine translation (Copestake et al., 1995), or subsumption-based reasoning for recognizing textual entailment (Lien and Kouylekov, 2015)[10]

## 2.5 Abstract Meaning Representation

Abstract Meaning Representation (AMR) is a semantic representation that has made a big splash in the NLP community, leading to a large volume of papers published that present methods of parsing AMR from sentence text. AMR was developed as a semantic representation of language that abstracts away from morpho-syntactic idiosyncrasies (Banarescu et al., 2013). The idea was to introduce a simple, but unified – bringing together argument structure identification, preposition attachment, etc. – semantic representation of sentences, similar to constituent trees for syntax. They cover a wide range of linguistic phenomena in their representations, which is admirable. I expect that the expressive limitations of AMR representations will make it difficult to be used in a wide variety of applications and that the decoupling of the representation from surface text will make AMR recovery from the surface text difficult.

**Logical format**

$\exists w, g, b:$
instance(w, want-01) $\wedge$
instance(g, girl) $\wedge$
instance(b, believe-01) $\wedge$
arg0(w, g) $\wedge$ arg1(w, b) $\wedge$ arg0(b, g)

**AMR format**

```
(w / want-01
      :arg0 (g / girl)
      :arg1 (b / believe-01
            :arg0 g))
```

**Graph format**

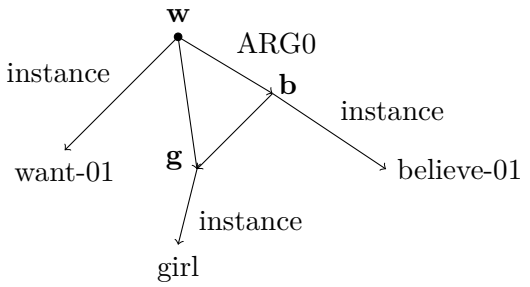



Figure 2.4: AMR representations for "The girl wanted to believe herself".

### 2.5.1 AMR Representation

AMRs are rooted, directed, acyclic graphs where the leaves act as node labels and relations and properties are defined between the nodes. AMR has a descriptively

[10]Lien and Kouylekov use a representation that is a combination of OWL-DL (OWL Working Group, 2004) and Horn-like rules but does not overcome the limitations of either representation.

equivalent neo-Davidsonian (Davidson, 1969) logical format which introduces a implicitly existentially quantified variable for every entity, event, property, and state, and a text format based on PENMAN inputs (Mathhiessen and Bateman, 1991) for convenient text-based interfacing. Figure 2.4 shows an example in all three formats.

AMR relations are include core PropBank (Kingsbury and Palmer, 2002; Palmer et al., 2005) arguments, select semantic relations from the thematic roles literature, and relation for special domains such as quantities, dates, and lists.

AMR generalizes across parts of speech and etymologically related words, which leads to the following four examples all being annotated with the concept "fear-01".

- My fear of snakes
- I am fearful of snakes
- I fear snakes
- I'm afraid of snakes

Banarescu et al. (2013) and the AMR Specifications (Banarescu et al., 2015) discuss all the phenomena that AMR handles in detail.

### 2.5.2 Semantics of AMR

Johan Bos provides a model-theoretic analysis of AMR in a squib published in Computational Linguistics (Bos, 2016). He showed that standard AMR representations are a subset of FOL with up to one universal quantifier[11] and presented a potential extension to AMR that would capture multiple universal quantifiers.

Providing AMRs with a model-theoretic interpretation is a step forward, but it remains a subset of FOL, which in itself is not expressive enough to appropriately express many important natural language phenomena, such as intensionality and reification of quantified sentences (Schubert, 2015). Bender et al. (2015) point out concrete limitations of AMR expressivity with the following groups of sentences that generate the same AMR representations despite non-trivial differences in meaning:

1. (a) Every person failed to eat.
   (b) No one ate.

2. (a) The boy is responsible for the work.
   (b) The boy is responsible for doing the work.
   (c) The boy has the responsibility for the work.

[11]The AMR specifications note that universal quantifiers are not represented by AMR. Bos overcomes this issue by using a series of polarity operators to capture the semantics of a universal quantifier.

In the first pair of sentences, sentence 1a requires some sort of attempt, intention, or expectation to eat, whereas sentence 1b does not require such a context. In the second pair of sentences, sentence 2a, 'work' may refer to the resulting object that already exists (such as a work of art) whereas sentences 2b and 2c cannot have such an interpretation.

### 2.5.3 AMR corpus

The AMR project has annotated a total of 47,274 sentences, of which 21,065 are available publicly or to Linguistic Data Consortium (LDC) members[12]. The rest of the sentences are only available to Deep Exploration and Filtering of Test (DEFT) DARPA program participants. The annotations break down into the following domains:

- *The Little Prince* corpus : 1,562 sentences.
- Bio AMR corpus : 6,452 sentences. This corpus consists of three cancer-related PubMed articles in full, the result sections of 46 PubMed paper, and 1000 sentences from each of the BEL BioCreative training corpus and the Chicago Corpus.
- LDC corpus : 39,260 sentences (13,051 general release). The source data of the DEFT-only corpus is not available, but the general release consists mostly of samplings from machine translation corpora with 200 sentences from weblogs and the WSJ corpus.

The three corpora do not all use the same version of AMR so they are not all useable at once with typical statistical training procedures.

### 2.5.4 AMR Editor

Hermjakob (2013) built a special editor for annotating AMR representations. The editor provides multiple input methods to fit an annotator's preferences

1. **Text Commands** – These are Unix-style text commands for editing the AMR graph. It includes a *last command* feature which is widely used in Unix shells.
2. **Templates** – The editor provides template versions of the text commands. Selecting a template opens a window with arguments slots to be filled out.
3. **Point and Click** – The annotator may point and click part of the annotation-in-progress to highlight a segment that they wish to delete or modify.

[12]Numbers computed from AMR download website: `http://amr.isi.edu/download.html`

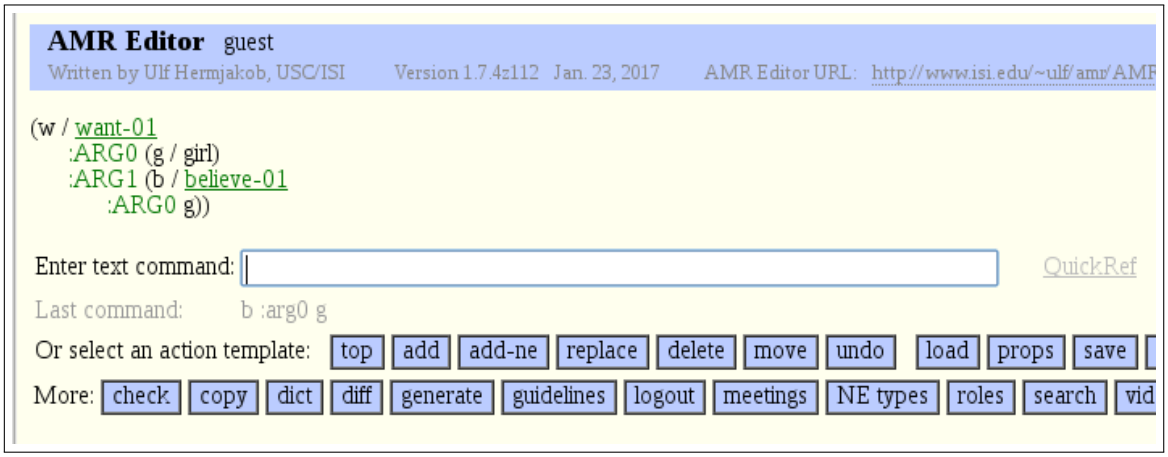



Figure 2.5: Screenshot of the AMR Editor editing the sentence "The girl wants to believe herself."

4. **Post Editing** – The AMR editor can automatically generate approximate AMR annotations from OntoNotes (Hovy et al., 2006) annotations.

The AMR editor also comes with annotator support in terms of links to the AMR Editor manual and FAQs, the AMR guidelines, the lists of roles in AMR, lists of named-entity types, and underlining some words with suggestions for concepts or properties. Figure 2.5 displays a screenshot of the editor in use.

### 2.5.5 Limitations

Beyond the semantic limitations already discussed in Section 2.5.2, AMR also have limitations intrinsic in its design that limit its usefulness even in tasks that AMR set out to assist. For example, the lack of grammatical number, tense, aspect, quotation marks, etc. would make precise natural language generation challenging, which Banarescu et al. (2013) list as one of the tasks they expect an AMR corpus would help. Schneider et al. (2015) argue that AMR does not capture tense because it does not generalize to other languages at a sentence-level. However, this seems like a strange design decision considering AMR is not supposed to be an interlingua and is closely tied to English vocabulary anyway.

# Chapter 3

# Research Project Description

## 3.1 Research Plan Overview

The major parts of our plan are the creation of an annotation environment, the collection of a significant body of ULF annotations, the training of a semantic parser on the annotated corpus, and the demonstration of the parser usage in a variety of inference tasks that the parser will enable.

1. **Annotation Environment and Corpus Building**

   The first stage of the research plan is to build an annotated corpus of ULF formulas from text. This involves developing an annotation method and interface for ULFs that enable fast and accurate annotations with minimal training. This stage is currently underway with two pilot annotation experiments completed, a 42-page annotation guideline, and a communal annotation interface for easy annotating and correction.

2. **Learning a Statistical Parser**

   The second stage of the research plan is to learn a statistical ULF semantic parser over the dataset for further use. I will present what we currently consider to be the most promising approach to this given the nature of the problem and the expected corpus size. We are optimistic about training an effective parser due to ULF's similarity to syntax and surface language.

3. **Evaluating the Parser & Beyond**

   The final stage of the research plan is to evaluate the parser internally via comparisons with a held-out portion of the corpus and as a resource for generating structured inferences such as those described in Section 1.1.3. At this

stage we will also measure the degree to which this parser improves knowledge acquisition tasks that were previously explored with brittle semantic parsers.

## 3.2 Completed and On-going Work

### 3.2.1 Lexical Axiom Extraction in Episodic Logic

Though this work does not directly fall into the topical framework of the research project proposed here, it was a core motivating factor in undertaking this project so a brief summary is included. Please see the published paper (Kim and Schubert, 2016) for full details on this work. We developed a rule-based approach to extract axioms in EL from WordNet verb entries by supplementing the gloss interpretation with WordNet verb frames and inferred structure from examples. We also developed a generalization to the *smatch* metric used in the AMR project, called *EL-smatch* to evaluate semantic matching EL formulas. Axioms generated by our approach proved competitive with the state-of-the-art in a verb entailment dataset and the axioms had an *EL-smatch* F1-score of 0.83.[1].

This axiomatization approach consists of three major steps:

1. Argument structure inference
2. Semantic parsing of the gloss
3. Axiom construction

Figure 3.1 shows the entire process for the example, *slam2.v*. The argument inference step refines the WordNet sentence frames using the provided examples. Specific pronouns associated with argument position are inserted as dummy arguments into the corresponding argument positions in the gloss, and the modified gloss is semantically parsed into EL. Axiom construction replaces the dummy arguments with variables and constructs a scoped axiom relating the entry word and the semantic parse of the gloss using the characterization operator '**'. In the simple example *slam2.v*, most of the subroutines used in each step have no effect. All transformations outside the scope of the BLLIP parser are performed with hand-written rules, which were fine-tuned using a development set of 550 verb synset entries.

Figure 3.2 illustrates an example of a simple EL forward inference chain using an axiom from WordNet for the sentence "John stumbles, but does not fall". Using the axiom for *stumble2.v*, and a hand written axiom schema asserting that statements

[1]For reference on what a good *smatch* score is, the current state-of-the-art AMR parsing in newswire is an F1-score of 0.71 (Zhou et al., 2016). Note that the two tasks are far too different to make a direct comparison of the system performances.

**WordNet entry**
*slam2.v*
Tagged gloss:
`(VB strike1) (RB violently1)`
Frames:
`[Somebody slam2.v Something]`
`[Somebody slam2.v Somebody]`
Examples: ("slam the ball")

**1. Argument Structure Inference**
Refined Frames:
`[Somebody slam2.v Something]`

**2. Semantic Parsing**
Parse: `(Me.pro (violently1.adv (strike1.v It.pro)))`

**3. Axiom Construction**
Axiom:
```
(∀x1 (∀y1 (∀e [[x1 slam2.v y1] ** e]
              [[[x1 (violently1.adv (strike1.v y1))] ** e]
               and [x1 person1.n] [y1 thing12.n]])))
```

Figure 3.1: Example gloss axiomatization process for WordNet entry *slam2.v*. The numbering corresponds to the subsections where these stages are discussed in detail.

conjoined with the connective "but" asserts the conjunction of the two statements as well. The semantics of abstract words, such as "but" need to be encoded by hand since dictionaries simply define abstract words in cycles. This second axiom is an axiom schema since it uses substitutional quantification over well-formed formulas, $\forall_{wff}$. Substitutional quantification is part of what allows EL to represent information about its own syntax and is used for meta-syntactic reasoning. Substitutional quantification and meta-reasoning in EL is explained in detail by Morbini & Schubert (2008). The inference process concludes that "John misses a step and nearly falls". This is an example of an inference that representations using an intersective approach to predicate modification cannot make since "John nearly falls" and "John does not fall" would contradict each other.

In the error analysis we found that most of the semantic parsing errors arose from a failure in the sentence parser or preprocessing directly preceding the sentence parser. That is, 17 out of the 52 axioms had errors arising from the sentence parser. These errors arose from either linguistic patterns that we did not encounter in our development set or in complex sentences that we didn't expect to succeed in (e.g. *take a walk for one's health or to aid digestion, as after a meal*).

*Axioms*

A1. *stumble2.v* : *miss a step and fall or nearly fall*

`(∀x,e: [[x stumble2.v] ** e]`
`[[(∃ z: [z step2.n] [x miss4.v z]) ∧`
`[[x fall23.v] ∨ [x (nearly.adv fall23.v)]]] ** e])`

A2. *If two statements are conjoined by "but", then both statements are true (i.e. conjunction)*

$(\forall_{wff}$ `x,y (∀ e: [[x but.cc y] ** e] [[x ∧ y] * e]))`

*Inference*

Sentence: "John stumbles, but does not fall"

| | | |
|---|---|---|
| I1. | `[[John.name stumble2.v] but.cc ¬[John.name fall23.v]]` | Parsed sentence |
| I2. | `[[John.name stumble2.v] ∧ ¬[John.name fall23.v]]` | I1 & A2 |
| I3. | `[John.name stumble2.v],¬[John.name fall23.v]` | I2 |
| I4. | `[(∃ z: [z step2.n] [John.name miss4.v z]) ∧ [[John.name fall23.v] ∨ [John.name (nearly.adv fall23.v)]]]` | I3 & A1 |
| I5. | `[(∃ z: [z step2.n] [John.name miss4.v z]) ∧ [John.name (nearly.adv fall23.v)]]` | I3 & I4 |

*Contradiction using OWL-DL*

| | | |
|---|---|---|
| I6. | `[John.name (nearly.adv fall23.v)]` | I5 |
| I7. | $\forall_{of}$.(John.name) $\sqcap$ $\forall_{of}$-1.(nearly.adv) $\sqcap$ fall23.v | I6 (Represent in OWL-DL) |
| I8. | $\forall_{of}$.(John.name) $\sqcap$ fall23.v | I7 |
| I9. | `[John.name fall23.v]` | I8 (Represent in EL) |
| I10. | $\square$ | I3 & I9 (Contradiction) |

Figure 3.2: If John stumbles, but doesn't fall, we can infer from the axioms extracted from WordNet verbs that he misses a step and nearly falls. This inference would lead to a contradiction with representations that use an intersective approach to predicate modification, such as OWL-DL.

We ultimately want to extend this gloss interpretation project to nouns and adjectives, but given the error rate in EL parses with the hand-built semantic parsing system we can't expect to generate widely usable axioms. This becomes even more problematic given the relative complexity of noun glosses compared to verb glosses. For example, consider the WordNet entry to *tree1.n*, the first noun sense:

> *tree1.n: (a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms)*

Many noun glosses have this sort of highly structured description including sub-

parts, common features, or uses (in cases of artifacts). Notice that getting the right operator argument structure in light of the nesting occurring in this sentence is a challenge. These are exactly the features that our project aims to focus on and produce with high-precision. An accurate EL transducer is also critical for deploying the axiom in a wide range of tasks. The axioms can only be used to their full potential when EL interpretations exist for the relevant text.

### 3.2.2 Pilot Annotations

We have performed two pilot annotations, one in Fall 2016 and one in Fall 2017. These were used to measure ULF annotation difficulty and training time and to get feedback on the annotation interface.

**Pilot Annotation 1, Fall 2016**

We performed our first exploratory annotation effort early in the project after developing an initial version of the annotation guidelines and a simple annotation tool inspired by the AMR Editor (Hermjakob, 2013). We had three annotators of varying degrees of expertise in EL: one expert, one intermediate, and one beginner. The sentences we annotated were randomly sampled from the Brown corpus, filtered to limit the sentence length to 17 words (the average sentence length in the corpus). Each annotator annotated between 27 and 72 sentences from the same set of sentences to measure correlation.

We found that annotators can learn to annotate quickly after practicing on a fairly small set of sentences and that the sentences could be annotated fast enough for building a corpus to be feasible. Figure 3.3 shows a plot of timing results by annotator and the number of annotation completed with a 5-cell moving window average to control for variable individual sentence difficulty. We can see that the intermediate and expert annotators have pretty consistent annotation speeds as a function annotations completed, as we would expect from their familiarity with the formulation. The beginner on the other hand, whose only annotation experience was with artificial and short sentences from the annotation tutorial, developed an annotation speed comparable to the intermediate and expert with 10 examples.

| **Annotator** | **Minutes/Sentence** |
|---|---|
| Beginner | 12.67 |
| Beginner (- first 10) | 6.83 |
| Intermediate | 7.70 |
| Expert | 6.87 |

Table 3.1: Average timing of experimental ULF annotations.

Table 3.1 shows that a practiced annotator averages less than 8 minutes per sentence. This is promising since it is less than the 10 minutes/sentence annotation

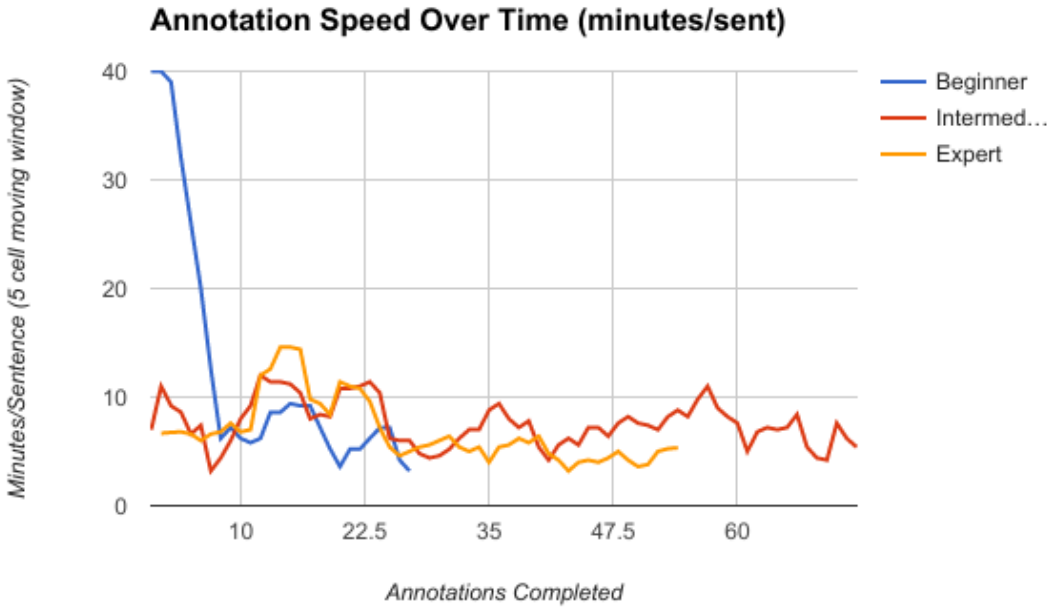



Figure 3.3: Timing results from ULF experimental annotations.

speed reported on the AMR project website[2], which has been successful in annotating over 40,000 sentences.

We also found that the ULF representations are capable of expressing the variety of linguistic phenomena that occur in the Brown corpus. There were only a couple of cases where the expert and intermediate annotators were unable to generate a confident interpretation in terms of normal sentence semantics. The annotators were mainly stumped in cases of ill-formed sentences such as fragments.

The expert and intermediate annotators had 53 annotations that were common between them, from which we computed a simple interannotator agreement with the following formula (based on the triple representation used in *EL-smatch*):

$$(2T_b)/(T_1 + T_2)$$

where

$T_b$ : number of triples in both annotations

$T_1$ : number of triples in annotation 1

[2]http://amr.isi.edu/editor.html. Visited 4/24/2017.

$T_2$ : number of triples in annotation 2

The 53 annotation pairs had an interannotator agreement of 0.48. This is insufficient, but expected from exploratory annotations. A review of the discrepancies showed that they were caused by the following issues (in order of severity):

1. The annotators could not consistently annotate phenomena that require movement of large phrases, such as prepositional modifiers which usually post-modify in English, but form prefix-operators in ULF.
2. Ill-formatted text, such as fragments, were not discussed in the preliminary guidelines.
3. Language phenomena that were not carefully discussed in the guidelines, so the standard representation was unclear to the annotator.

Additionally, we found that the AMR style annotator was not helpful because the similarity between ULF and surface form means that it's easier to directly annotate on the surface word than building up a graph.

Based on these results, we changed the annotator style to that closer to GMB (Bos et al., 2017). The annotation collection was changed to be collaborative so that all annotators have a shared view and experts can review and correct annotations. This setup reduces annotation error and streamlines the annotator training process. Additionally, each sentence annotation was broken up into multiple stages. Here is an example of the stages for the sentence "Mary loves to solve puzzles":

1. Group syntactic constituents (NPs, ADJPs, VPs, etc) using round brackets:
   `(Mary (loves (to (solve puzzles))))`;
2. Run a POS tagger over the original sentence:
   `(nnp Mary) (vbz loves) (to to) (vb solve) (nns puzzles)`;
3. Make any necessary corrections to tags, and then use them as dot-extensions in the bracketed sentence (the tag to dot-extension will be automated):
   `(Mary.nnp (loves.vbz (to.to (solve.vb puzzles.nns))))`; *No corrections needed*
4. Convert POS extensions to logical-types, and separate tense and plural as operators:
   `(|Mary| ((pres love.v) (to (solve.v (plur puzzle.n)))))`;
   (`|_|` ↔ name (proper noun); `.v` ↔ verbal predicate; `.n` ↔ nominal predicate; `to` without an extension is a special reifying operator);

5. Add any necessary implicit operators (typically, type-shifting operators):
   `(|Mary| ((pres love.v) (to (solve.v (k (plur puzzle.n))))))`;
   (`k` converts a predicate that is true of ordinary singular or plural entities into a kind – i.e. an abstract individual whose instances are ordinary entities; it is applied whenever we have a common noun phrase lacking a determiner (a so-called "bare noun phrase").)

The multi-stage annotation enables us to incorporate multi-stage gold-standardizing as done in the Parallel Meaning Bank, so we could take a more incremental approach to annotation if deemed necessary. In addition, we introduced syntactic sugar, *macros*, to ULF to minimize phrase movement and implicit operator insertion. These *macros* are described in more detail after the pilot annotation descriptions.

**Pilot Annotation 2, Fall 2017**

This pilot annotation used a team of eight annotators and annotated the Tatoeba dataset. 270 sentences were annotated, 80 of which were timed. The average annotation speed was 8 minutes per sentence with 4 minutes per sentence among the two experts and 11 minutes per sentence among the three trainees that participated in the timed annotations. We did not time the reviewing time of annotations done by non-experts. They took less time than outright annotations if any correction was needed at all.

After this annotation pilot we decided to reduce the annotation interface to just syntactic parsing and final conversion to logical form. Annotators are still taught using the five-step process outlined above, but during annotation time this is no longer required. We found the overhead for separating out each step was not worth the simplification.

### 3.2.3 Macro Development

In order to simplify annotations and reduce possibility of annotator error, we developed relaxations/modifications to the ULF syntax that makes the representation closer to surface text but can be used to deterministically recover the original ULF format. In order to accomplish this, we have so far introduced three different types of modifications to ULF:

1. relaxation of well-formedness constraints
2. lexical marking of scope
3. introduction of syntactic *macros*

Most of these changes to the ULF representation correspond to particular linguistic phenomena, rather than a general shift in representation. Although this leads to a large collection of operators for the annotator to learn to use, their relation to well-known linguistic phenomena make them simple to learn and remember by annotators with linguistic backgrounds. I introduce particular phenomena here to show how the general mechanisms work, but do not enumerate all the developed simplifications.

The first and simplest relaxation of the well-formedness constraints is a type of operator dropping. This occurs when predicates act as predicate modifiers, such as in compounding. For example, consider the phrase "burning hot melting pot". The desired ULF interpretation is

```
((attr ((adv-a burning.a) hot.a)) ((nn melting.n) pot.n)).
```

Since `burning.a` and `melting.n` are predicates, not predicate modifiers, they must be type-shifted to map them to the correct semantic type. `adv-a` maps a monadic predicate to a monadic verb/adjective-modifier, whereas `nn` maps a noun predicate to a noun predicate modifier. Similarly, `attr` is an adjective predicate to noun-predicate modifier. The need for a predicate-to-predicate modifier function and the one that is required for the correct mapping can be determined with a simple syntactic analysis. This is because there are no valid predications with the pairs of types listed. Thus we can simply add the type-shifter that corresponds to the pairs of types encountered. Below is a list of the predicate-modifier constructors and the types involved.

- `nn` - noun to noun modifier
- `nnp` - noun phrases to noun predicate modifier
- `attr` - adjective to noun predicate modifier
- `adv-a` - any predicate to monadic verb/adjective modifier

Lexical marking of scope has been applied to adverbs, which may be sentence or verb-level modifiers, but either type can appear at various positions in the sentence. Rather than requiring the annotator to actually place the adverb at the scope that it acts, we simply require them to append a tag that indicates at which scope it acts. Consider the following sentences:

(a) "Mary confidently spoke up"

(b) "Mary undoubtedly spoke up"

In sentence (a), "confidently" is a predicate modifying adverb, whereas in sentence (b) "undoubtedly" is a sentence modifying adverb. Rather than requiring the

ULF annotations (b) and (c) which require movement of the adverb, we use `-v` and `-s` tags to distinguish between the verb and sentence-level adverbs as shown in annotations (d) and (e). If there's an inconsistency, we can lift the adverb to the appropriate location.

(b) `(|Mary| (confidently.adv <past speak_up.v>))`

(c) `(undoubtedly.adv (|Mary| <past speak_up.v>))`

(d) `(|Mary| (confidently.adv-v <past speak_up.v>))`

(e) `(|Mary| (undoubtedly.adv-s <past speak_up.v>))`

Syntactic macros are new operators we introduce that map to a particular syntactic transformation or expansion, similar to a macro is in the C programming language. These are introduced for phenomena with more complex phrase shifting patterns, such as relative clauses, or common phenomena in English that lead to complex, but derivational semantic representations, such as relative clauses and genitives.

Consider the relative clause example "The car that you bought". The fully explicit ULF formula for this is the following.

```
(the.d (:l x ((x car.n) and (you.pro (past buy.v) x))))
```

To make this annotation closer to the surface form we introduce two operators: `n+preds` and `sub`, and a specially interpreted atom `that.rel`. `n+preds` maps to a lambda that jointly asserts the predicates listed in its arguments. `sub` inserts its first argument in place of every occurrence of `*h` in its second argument. Then `that.rel` is regarded as a special variable `*r` which is lambda-abstracted at the level of its lowest-embedding sentential form. Using these we can annotate "The car that you bought" as

```
(the.d (n+preds car.n (sub that.rel (you.pro (past buy.v) *h))))
```

via definition of `n+preds` this is equivalent to

```
(:l x ((x car.n) (x (sub that.rel (you.pro (past buy.v) *h)))))
```

via definition of `sub` this is equivalent to

```
(:l x ((x car.n) (x (you.pro (past buy.v) that.rel)))))
```

via `that.rel` interpretation this becomes

```
(:l x ((x car.n) and (x (:l *r (you.pro (past buy.v) *r)))))
```

via lambda-conversion this becomes the fully explicit ULF.

```
(:l x ((x car.n) and (you.pro (past buy.v) x)))
```

The full benefit of this alternate representation does not become apparent in such a simple example. These macros appropriately handle more complicated pied piping and multiple post-nominal modifiers in a simple manner with almost no reordering of the surface word order.

### 3.2.4 Annotation Release 1

We are currently preparing for an annotation push this summer to kick off the collection of the first major annotation release. We have a handful of annotators available for about 20 hours a week to participate in this annotation project. We're expecting to collect about 3,000 ULFs over the summer since the last pilot annotation averaged 8 minutes per annotation. For comparison, the initial AMR corpus was built by 12 LDC annotators, working during a quarter of a year, managing to produce around 10,000 AMRs with each sentence taking 10 minutes to code on average (Hermjakob, 2013). These annotations will primarily be from the Tatoeba dataset, but will expand into more complex datasets as we verify that the ULF guidelines cover the necessary phenomena.

### 3.2.5 Pilot Inference Demo

We created a small set of inference rules applicable to ULFs, for a subset of the suggested inference types, namely requests and counterfactuals. This was done by Len Schubert for a development set of 10 Tatoeba sentences showing these phenomena. I then independently annotated two random sets of Tatoeba sentences without seeing the exact rules that Len wrote. Of course we agreed on the sorts of inferences that we'd be testing (e.g. what question the request is asking, if any; what are the counterfactual sentences embedded in a counterfactual construction, if any). The first set, containing 65 sentence-derived ULFs, and obtained unselectively (i.e. uniformly randomly across Tatoeba), generated 5 correct inferences from 3 ULFs, and no incorrect ones. The second set, selected on the basis of containing a keyword such as *could* or *wish*, contained 71 ULFs. These produced 45 good inferences from 39 formulas, 8 context dependently correct inferences from 4 formulas, and 13 incorrect inferences from 10 formulas. Considering the small size of the development set, these are promising results. Moreover the errors turned out to be systematically due to a form of counterfactual not seen in the development set "If I were *to* ..." and use of sentence-initial *can* in non-requests. The inference rules could easily be amended to avoid these errors. These preliminary results of course had the advantage of using "expert" ULFs, rather than automatically generated ones. Also they do not yet tell us much about recall – except that the great majority sentences containing one of the keywords did indeed generate a warranted inference.

### 3.2.6 Attitudinal, Counterfactual, Request, and Question Inference Demonstration

We are currently developing a method to gather a more wide-ranging dataset of structural inferences on complement-taking verbs, counterfactual constructions, requests,

| *Sample* | *# sent.* | *# inf.* | *Corr.* | *Contxt*[a] | *Incorr.* | *Precision*[b] | *Recover*[c] | *Precision*[d] |
|---|---|---|---|---|---|---|---|---|
| General | 65 | 5 | 5 | 0 | 0 | 1.00 | 0 | 1.00 |
| Domain | 71 | 66 | 45 | 8 | 13 | 0.68/0.80 | 8 | 0.80/0.92 |
| Total | 136 | 71 | 50 | 8 | 13 | 0.70/0.81 | 8 | 0.82/0.93 |

Table 3.2: Results for the preliminary inference experiment on counterfactuals and requests. The general sample is a set of randomly sampled sentences, and the domain sample is a set of keyword-sampled sentences that we expect to have the sorts of phenomena we're generating inferences from. All sentences are sampled from the Tatoeba dataset.

[a]Correctness is contextually dependent (e.g. "Can you throw a fastball?" → "I want you to throw a fastball.").

[b][assuming context is wrong]/[assuming context is right] for context dependent inferences.

[c]Recoverable with no loss of correct inferences.

[d]Precision after loss-less recoveries.

and questions.

This work aims to concretely demonstrate how ULFs can be used for four of the five inference types described in Section 1.1.3. As the heading for this subsection suggests the inferences we're looking for are from attitudinal verb, counterfactual, request, and question constructions. These inferences are important for language understanding tasks, particularly dialogue comprehension, and include entailments, presuppositions, and pragmatic forces. We treat any defeasible inference with a sentence-level operator of appropriate likelihood (e.g. `probably.adv-s`, `possible.adv-s`) so that within the probabilistic inference framework, these can be canceled by more concrete evidence of their negations.

Due to the structural nature of these inferences, we can generate a reasonable dataset for evaluating forward inferences. This, in general, it very difficult to do because questions often have multiple equally correct answers. For structural inferences, we can make a minimum edit constraint between the source and consequent. Consider an example from Section 1.1.3: "She managed to quit smoking" entails "She quit smoking". The particular details are being filled into structure of the entailed sentence directly from the source sentence the same way that "She managed to complete the run even with the ankle injury she sustained" entails "She completed the run even with the ankle injury she sustained". Though paraphrases or less descriptive entailments can be described, a canonical entailed sentence can be defined.

**Dataset Construction.** We chose a variety of text sources for constructing this dataset to reduce genre-effects and provide good coverage of all the phenomena we

are investigating. Below I describe the datasets included for this collection. Some of these datasets include annotations that we ignore other than to identify sentence and token -boundaries.

- **Tatoeba** [3]

  The Tatoeba dataset consists of crowd-sourced translations from a community-based educational platform. People can request the translation of a sentence from one language to another on the website and other members will provide the translation. Due to this pedagogical structure, the sentences are fluent, simple, and highly-varied. The English portion downloaded on May 18, 2017 contains 687,274 sentences.

- **Discourse Graphbank (Wolf, 2005)**

  The Discourse Graphbank is a discourse annotation corpus created from 135 newswire and WSJ texts. We use the discourse annotations to perform sentence delimiting. This dataset is on the order of several thousand sentences.

- **Project Gutenberg** [4]

  Project Gutenberg is an online repository of texts with expired copyright. We downloaded the top 100 most popular books from the 30 days prior to February 26, 2018. We then ignored books that have a non-standard writing style: poems, plays, archaic texts, instructional books, textbooks, and dictionaries. We developed a custom tokenizer and sentence delimiter by hand on this dataset largely based on regex patterns. This dataset totals to 578,650 sentences.

- **Switchboard (Calhoun et al., 2010)**

  Switchboard is a telephone dialogue corpus on a wide range of topics. There have been many layers of annotations made on this dataset. We perform tokenization, disfluency elimination, and sentence delimiting using a combination of these annotations. Our normalized version of the dataset consists of 109,753 sentence-like utterances.

- **UIUC Question Classification (Li and Roth, 2002)**

  The UIUC Question Classification dataset consists of questions from the TREC question answering competition. This dataset covers a wide range of question structures on a wide variety of topics, but focuses on factoid questions. This dataset consists of 15,452 questions.

[3] `https://tatoeba.org/eng/`

[4] `https://www.gutenberg.org`

We chose to hand-build tokenizers and sentence-delimiters for each dataset for a couple of reasons. First, we didn't want to introduce biases from models trained on particular genres. Since this dataset spans multiple genres, we wanted to avoid building a biased dataset due to a improperly tuned tokenizers/delimiters. Second, most of the datasets have pretty regular patterns that can be identified and written in when built by hand. This also meant that we could take advantage of available annotations concurrently with the rules. The transparency of the rules also have the benefit that we can interpretably fix errors in their performance.

Since the phenomena we want to focus on are relatively infrequent, we wrote sampling patterns to only keep sentences that superficially look like they could contain one of these phenomena. These patterns are written in linguistically-augmented regex patterns to avoid parsing model bias. The sampling method was designed to be human interpretable while minimizing false positives and keeping false negatives to essentially zero. Since we will be getting human annotators to mark actual inferences, some false positives are not problematic. We simply needed a filtering mechanism so that we can reduce the number of annotations needed to get a sufficient number of positive examples. Requests, for example, occur once in roughly every 100 to 1000 sentences, depending on the genre. We opted to use a syntactically augmented regex patterns. For example, here are two augmented regex patterns for if/then counterfactual construction and an inverted version:

```
"<begin?>(if|If)<mid>(was|were|had|<past>|<ppart>)<mid?>(<futr>) .+"
"<begin?>(<futr>)<mid>if<mid>(was|were|had|<past>|<ppart>) .+"
```

<begin?> indicates that it is either the beginning of the string or space separated from previous text (this qualification exists because this sentence itself may be embedded. <mid> words that are padded with spaces on the sides (i.e. separate tokens from what's defined next to it) and <mid?> is a variant that simply allows a space as well. <past> and <ppart> are alternative lists of past tense and past participle verb forms. <futr> is an alternatives list of different conjugations of "will".

**Inference Annotation.** Our preliminary plan for inference annotation on this dataset is to ask the annotator to select the structural inference pattern that hold for the given sentence and write down the corresponding inferred sentence. For example, say there is the sentence "If I were rich, I would own a boat". The annotator would select an inference template along the lines of `(if <x> were <pred>, <x> would <q>)` $\rightarrow$ `(<x> is not <pred>)` and write down the inference "I am not rich". This way we can get a fluent inference, but push the annotator to think about the inferences structurally. We will also include the option for annotators to add rules thats aren't displayed because some of the more pragmatic inferences are difficult to predict ahead of time but easy to notice. We are still in the prototyping stage for the inference annotation

| Dataset | impl | ctrftl | request | question | interest | ignored |
|---|---|---|---|---|---|---|
| Disc. Grphbnk | 1,987 | 110 | 2 | 47 | 2,030 | 1,122 |
| Proj. Gutenberg | 264,109 | 31,939 | 2,900 | 60,422 | 303,306 | 275,344 |
| Switchboard | 37,453 | 5,266 | 472 | 5,198 | 49,086 | 60,667 |
| UIUC QC | 3,711 | 95 | 385 | 15,205 | 15,251 | 201 |
| Tatoeba | - | - | - | - | - | - |

Table 3.3: Sample statistics for each dataset given the sampling method described in this section. Statistics for Tatoeba has not been generated because a cursory look over the samples indicated a good distribution of results. These statistics were generated as part of the dataset selection phase.

so the details are in flux.

For inferences from attitudinal verbs and counterfactual constructions, a large portion of the inferences have to do with varying degrees of factuality of the embedded sentences. For this we should be able to use annotations from the recently released event factuality dataset (Rudinger et al., 2018).

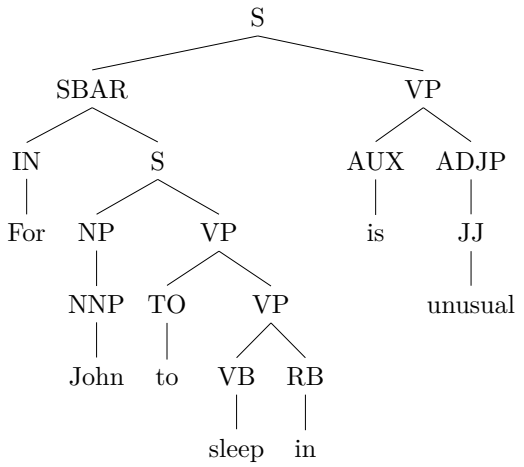
**Inference Evaluation.** The evaluation will be done using F1 over the exact inferences that were annotated. For this initial demonstration the inference rules for ULF will be written by hand using a development set of examples from the dataset just as in the pilot inference evaluation. Once we've trained a precise ULF parser, we could deploy it in automatic knowledge acquisition algorithms to learn these sorts of rules.

## 3.3 Next Steps

### 3.3.1 Learning the Semantic Parser.

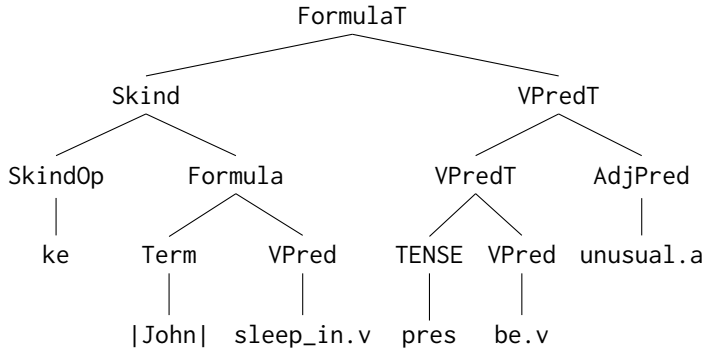
To learn the ULF parser, we plan to explore a tree-to-tree machine translation method and a string-to-tree parsing method; both are promising, given the similarity of ULF parsing to machine translation and to syntactic parsing. In both ULF parsing and machine translation, the input and output represent the same meaning with different vocabularies and syntax, but with overall similar structure. Our use of macros in ULFs for avoiding, e.g., explicit $\lambda$-abstraction and long-distance restructuring (in wh-questions, relative clauses, etc.) helps to strengthen the similarity of the target ULFs to the source sentences. We selected these approaches over the recently successful neural network approaches for our initial work because the annotation dataset will be relatively small, while neural network approaches rely on large datasets to perform best. By releasing the dataset to the public, we will encourage

(a) **Constituency tree**




(b) **Tree-form of ULF**

```
(((ke (|John| sleep_in.v))
  ((pres be.v) unusual.a))
```




(c) **Possible Rules**

```
S-FormulaT → SBAR-Skind VP-VPredT,
              SBAR-Skind VP-VPredT
SBAR-Skind → IN-SkindOp S-Formula,
              IN-SkindOp S-Formula
IN-SkindOP → For, ke
S-Formula → NNP-Term VP-VPred
NNP-Term → John, |John|
TO-VPred → to VP-VPred, VP-VPred
VP-VPred → sleep in, sleep_in.v
VP-VPredT → AUX-VPredT ADJP-JJ,
             AUX-VPredT ADJP-JJ
AUX-VPredT → is, (pres be.v)
JJ-AdjPred → unusual, unusual.a
```

Figure 3.4: Rules for the example sentence *For John to sleep in is unusual*. (a) and (b) are the constituency tree and tree representation of the ULF that correspond to the sentence, respectively. The non-terminals in (b) are logical type categories (those with 'T' appended are tensed variants). (c) lists a possible set of STSG rules that synchronously generates the two trees. Dashes are used to join the two tree non-terminal names. Notice that the learned rules collapse subtrees that only exist in only one (e.g. the "sleep in" constituency subtree and "pres be.v" logical subtree).

other researchers to develop progressively better ULF parsing methods.

The tree-to-tree translation method models the two representations as a Synchronous Tree Substitution Grammar (STSG) (Eisner, 2003; Gildea, 2003). This formalism consists of a grammar that generates two trees in parallel, with rules that expand a nonterminal into a pairs of corresponding tree fragments, one on each side of the transduction. See Figure 3.5 for the definition of a binary STSG. Examples of STSG rules that pair Penn Treebank style parse trees with ULF formulas are shown in Figure 3.4.

$$X \Rightarrow a, b$$

$$X \Rightarrow a_1 X^{[1]} a_2 X^{[2]} a_3, b_1 X^{[2]} b_2 X^{[1]} b_3$$

Figure 3.5: Binary STSG rules. The lowercase $a$ and $b$ stand for terminal symbols in the two languages, respectively and $X$ stand for non-terminals. The second rule describes the nonterminal case where the superscripts link the nonterminals on either side of the derivation. The nonterminals can be in either order for the nonterminal generation.

Learning STSG rules consists of two steps: (1) aligning the nodes between the syntactic and semantic trees and (2) learning larger, multi-node rules between the two trees. Node alignment will be learned through Variational Bayes to apply EM with heuristics based on string matching and available lexical types per token. These heuristics will considerably and reliably reduce the search space so that this will be successful on a small dataset. Larger rules are learned by extracting all pairs of tree fragments that are *consistently aligned*, that is, that have all nodes in the fragment on one side of the rule aligned to nodes in the fragment on the other sides, and vice versa. This can be sped up with rule-decomposition sampling with a Bayesian prior on the rule size (Post and Gildea, 2009; Chung et al., 2014).

But perhaps we will be able use string-to-tree parsing methods instead of STSG. The production rules in Figure 3.4 show that reordering between English syntax and ULF productions are rare. With minor extensions to the ULF composition types to allow for argument reordering (e.g., using `Formula` $\rightarrow$ `Term,VPred` and `Formula'` $\rightarrow$ `VPred,Term` to express the same semantics but with reversed productions) we could sufficiently capture the reordering between surface English and ULF formulas directly in a probabilistic context free grammar (PCFG). String-to-tree algorithms for PCFGs have significantly lower computational complexity than tree-to-tree algorithms for STSGs in general at the cost of not being able to do more complex reorderings.[5]

[5]The best parse can be decoded from binarized PCFGs in $\mathcal{O}((NT)^3)$ using Viterbi whereas STSGs can't be binarized so decoding is $\mathcal{O}((NT)^{(M+1)})$ where $N$ is the length of the sentence and $T$ is the number of non-terminals in the grammar and $M$ the maximum number of non-terminals

**Fine-tuning the Parsing Models to Inference Tasks.**
One of the worries in building a task-agnostic semantic parser is that the parser will be suboptimal for any particular task since the optimized training function differs from the task evaluation/reward function. We can overcome this by dynamically tuning the model to specific tasks through reinforcement learning. Any model that can be sampled from its distribution of outputs and is log-differentiable with respect to the model parameters can be optimized to maximize the expected reward using the REINFORCE algorithm (Williams, 1992):

$$\max_{\theta} \sum_{x_i \in X} E_{P(y_i|\theta,x_i)}[R(y_i)] \qquad \Delta\theta_i = \alpha(R(y) - \beta)(\frac{\partial}{\partial\theta_i} \ln(P(y|\theta, x)))$$

where $X$ is the set of inputs, $\theta$ are the model parameters, $y$ is the output, and $\alpha$ and $\beta$ are hyperparameters for the convergence rate. The efficacy of optimizing statistical models with REINFORCE for semantic parsing has been demonstrated by recent publications (Liang et al., 2017; Guu et al., 2017).[6] Researchers have also developed methods for making REINFORCE-based training more robust, such as experience replay (Mnih et al., 2015) and randomized beam search (Guu et al., 2017), that we will incorporate. Notice that log-linear models are log-differentiable so we can use this fine tuning approach with any log-linear neural network or grammar model.

### 3.3.2 Evaluation of the Semantic Parser

**Choosing a metric for internal evaluation.** In phrase-structure parse evaluations, a node of the test tree is scored as correct if it bears the same category label as the node of the reference tree that dominates the same word span. Since our primal LFs can be viewed as tree structures that correspond rather closely to a phrase structure tree, one may think that a similar evaluation metric could be employed here. However, we have no explicit category labels, and we will not in general have fixed sequences of atoms in the yield of a test and reference trees. The lack of category labels could be overcome by using the semantic categories of an LF grammar that describes semantically well-formed LFs. However, we are left with the potential mismatches at the leaf level, and therefore take our cue instead from the Smatch metric (Cai and Knight, 2013). Smatch in effect gives partial credit for each correct constituent (the predicate and each argument) of a predication, where

produced by an STSG rule.

[6]These publications learn semantic parsing models from distant supervision alone, which is feasible because they use restricted semantic representations to reduce the search space. We can think of the supervised training on ULF annotations as a smart initialization of parameters towards formulas that "look right".

the predications and constituents have been aligned via correspondences between the names of the unique variables associated with them.

We can follow a similar strategy by associating a unique variable with each nonatomic subexpression of a test LF and reference LF. Suppose that we are given a mapping between these variables as well as lambda-bound variables (and quantifier-bound variables, if any – though these generally appear only as a result of scope disambiguation). Under this mapping, we can score 1 unit for each immediate, correctly positioned constituent of each pair of corresponding subexpressions – where nonatomic constituents are identified by their associated variables. The main difference from matching AMR formulas is that we have a greater variety of expression types, e.g., quantifier phrases, modified predicates, and lambda-abstracts, but these can still be scored in the same way via their constituency.

The actual F1-score for a match depends on finding the variable correspondence that optimizes that score. Cai & Knight note that for AMR structures this is an NP-complete problem, but they were able to develop fast heuristic hill-climbing methods for the optimization problem that were quite accurate in relation to an exact (but relatively slow) integer linear programming approach. We expect to develop analogous algorithms that are equally fast and accurate. As a heuristic device, we will use semantic category parses based on an LF grammar for initial approximate structure alignment. (We already have an EL grammar and parser, but these are designed for fully scoped LFs, and thus will require adaptation.)

**Internal evaluation experiments.** We will use standard validation methods, holding out a portion (perhaps 10%) of the annotated corpora in training our semantic parser and evaluating on the held-out portion. We will perform these experiments with increasingly large annotated corpora in successive years, both because the growth of the corpora in the course of the project will make this possible, and because it is important determine how F1-scores improve as a function of training corpus size.

**External evaluation.** As explained in Section 3.2.6, we are propose evaluating on a novel forward inference task, one that focuses on phenomena that require minimal world knowledge, but fine-grained representations: implicative verbs, attitudinal and communicative verbs, counterfactuals, and questions and requests. As the examples discussed in 1.1.3 indicate, the inferences under consideration are instantly obvious and essential to understanding. The precision of such inferences is rather easy to evaluate via human judgments, but we will also perform evaluation of recall. This is in general hard for spontaneous inferences, because people are capable of reporting a wide variety of explanations, causal consequences, and stream-of-consciousness associations for any given sentence. This is presumably why

virtually all established inference challenges for general language are classificatory or multiple-choice. However, the inferences under consideration here are relatively easy to circumscribe, because of they are associated with very limited classes of key words and utterance forms. Thus the relevant types of inferences can be conveyed to experimental participants through explanation and examples. In fact, the reader who has perused the final four bullet points at the end of Section 1.1.3 probably has a pretty good idea already what inferences are at issue in these cases, and some further examples and explanations would consolidate that understanding. More objectively, the we found that students in AI classes have no difficulty catching on to the concept of implicatives in the context of a lecture on NLog – they are able to report implicative inferences quite readily for new examples. The same goes for inferences from communicative and attitudinal verbs, and the other categories enumerated in Section 1.1.3. The inference task can be channeled even more narrowly by requiring subjects to report just those inferences that they derive from the presence of particular words or utterance forms, specified as part of the task.

## 3.4 Conclusion

In this document I propose a research plan for developing a high-precision semantic parser for ULFs from a moderately sized annotated corpus. The approach of this research project is based on the hypothesis that a divide-and-conquer approach to semantic parsing will lead to more precise and useful semantic analyses than a one-shot approach. I discussed relevant semantic parsing and annotation projects and highlighted that this project fills a void in terms of tying the semantic representation to an expressive object language so that the descriptions and inferences can be supported by model theoretic analyses. Furthermore, I discussed completed and current work on effectively building up a corpus which involved borrowing techniques from the related works and developing novel logical form relaxations for simpler annotations without the loss of semantic precision. The project also includes measuring the structural inference capabilities of ULFs – to present all the uses of a ULF parser in NLP tasks. I wrap up the proposal with a clear plan on how we will train a high-precision semantic parser on a moderately-sized dataset by taking advantage of all the features of the data.

## References


[Abzianidze et al.2017] Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain, April. Association for Computational Linguistics.

[Allen and Teng2017] James Allen and Choh Man Teng. 2017. Broad coverage, domain-generic deep semantic parsing. In *AAAI Spring Symposium Series*.

[Allen et al.2007] James Allen, Mehdi Manshadi, Myroslava Dzikovska, and Mary Swift. 2007. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the Workshop on Deep Linguistic Processing*, DeepLP '07, pages 49–56, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Allen et al.2008] James F. Allen, Mary Swift, and Will de Beaumont. 2008. Deep semantic analysis of text. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, STEP '08, pages 343–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Allen et al.2011] James Allen, William de Beaumont, Nate Blaylock, George Ferguson, Jansen Orfan, and Mary Swift. 2011. Acquiring commonsense knowledge for a cognitive agent. In *AAAI Fall Symposium Series*.

[Allen et al.2013] James Allen, Will de Beaumont, Lucian Galescu, Jansen Orfan, Mary Swift, and Choh Man Teng. 2013. Automatically deriving event ontologies for a commonsense knowledge base. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 23–34, Potsdam, Germany, March. Association for Computational Linguistics.

[Allen et al.2015] James Allen, Will de Beaumont, Lucian Galescu, and Choh Man Teng. 2015. Complex event extraction using drum. In *Proceedings of BioNLP 15*, pages 1–11. Association for Computational Linguistics.

[Allen et al.2018] James F. Allen, Omid Bahkshandeh, William de Beaumont, Lucian Galescu, and Choh Man Teng. 2018. Effective broad-coverage deep parsing. In *AAAI Conference on Artificial Intelligence*.

[Artzi and Zettlemoyer2013] Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1(1):49–62.

[Artzi et al.2015] Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon, Portugal, September. Association for Computational Linguistics.

[Baker et al.1998] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Baldwin et al.2004] Timothy Baldwin, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen. 2004. Road-testing the English Resource Grammar over the British National Corpus. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2047–2050.

[Banarescu et al.2013] Laura Banarescu, 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. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Banarescu et al.2015] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2015. *Abstract Meaning Representation (AMR) 1.2.2 Specifications*. Available at `https://github.com/amrisi/amr-guidelines/blob/master/amr.md`.

[Basile et al.2012] Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3196–3200, Istanbul, Turkey.

[Bateman1990] John A. Bateman. 1990. Upper modeling: organizing knowledge for natural language processing.

[Bender et al.2015] Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London, UK, April. Association for Computational Linguistics.

[Bjerva et al.2016] Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan, December. The COLING 2016 Organizing Committee.

[Bos et al.2017] Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer.

[Bos1996] Johan Bos. 1996. Predicate logic unplugged. In *Proceedings of the 10th Amsterdam Colloquium*, pages 133–143.

[Bos2008] Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, STEP '08, pages 277–286, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Bos2015] Johan Bos. 2015. Open-domain semantic parsing with boxer. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*, number 109, pages 301–304. Linköping University Electronic Press, Linköpings universitet.

[Bos2016] Johan Bos. 2016. Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535, September.

[Bub et al.1997] T. Bub, W. Wahlster, and A. Waibel. 1997. Verbmobil: the combination of deep and shallow processing for spontaneous speech translation. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 71–74 vol.1, Apr.

[Burnard2000] Lou Burnard. 2000. User reference guide for the British National Corpus. Technical report, Oxford University Computing Services.

[Cai and Knight2013] Shu Cai 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.

[Calhoun et al.2010] Sasha Calhoun, Jean Carletta, Jason M. Brenier, Neil Mayo, Daniel Jurafsky, Mark Steedman, and David Beaver. 2010. The NXT-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44:387–419.

[Callmeier2001] Ulrich Callmeier. 2001. Efficient parsing with large-scale unification grammars. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany.

[Christodouloupoulos and Steedman2015] Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

[Chung et al.2014] Tagyoung Chung, Licheng Fang, Daniel Gildea, and Daniel Štefankovič. 2014. Sampling tree fragments from forests. *Computational Linguistics*, 40:203–229.

[Cooper et al.1996a] Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, and Steve Pulman. 1996a. Using the framework. Technical Report LRE 62-051 D-16, The FraCaS Consortium.

[Cooper et al.1996b] Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996b. Using the framework. Technical report, The Fracas Consortium.

[Copestake and Flickinger2000] Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage English grammar using HPSG. In *IN PROCEEDINGS OF LREC 2000*, pages 591–600.

[Copestake et al.1995] Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. 1995. Translation using minimal recursion semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*.

[Copestake et al.2001] Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*,

ACL '01, pages 140–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Copestake et al.2005] Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, 3(2):281–332.

[Davidson1969] Donald Davidson, 1969. *The Individuation of Events*, pages 216–234. Springer Netherlands, Dordrecht.

[Draicchio et al.2013] F. Draicchio, A. Gangemi, V. Presutti, and A.G. Nuzzolese. 2013. FRED: From natural language text to rdf and owl in one click. In *P. Cimiano et al. (eds.) , ESWC 2013*, pages 263–267. Springer.

[Eisner2003] Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics, companion volume*, pages 205–208, Sapporo, Japan.

[Evang et al.2013] Kilian Evang, Valerio Basile, Grzegorz Chrupała, and Johan Bos. 2013. Elephant: Sequence labeling for word and sentence segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1426, Seattle, Washington, USA, October. Association for Computational Linguistics.

[Flickinger et al.2012] Dan Flickinger, Yi Zhang, and Valia Kordoni. 2012. DeepBank. a dynamically annotated treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

[Giampiccolo et al.2007] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Gildea2003] Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of ACL-03*, pages 80–87, Sapporo, Japan.

[Guu et al.2017] Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *Association for Computational Linguistics (ACL)*.

[Hermjakob2013] Ulf Hermjakob. 2013. Amr editor: A tool to build abstract meaning representations.

[Hovy et al.2006] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA, June. Association for Computational Linguistics.

[Howard et al.2014] Thomas M. Howard, Stefanie Tellex, and Nicholas Roy. 2014. A natural language planner interface for mobile manipulators. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6652–6659.

[Hurum and Schubert1986] Sven Hurum and Lenhart Schubert. 1986. Two types of quantifier scoping. In *Proc. 6th Can. Conf. on Artificial Intelligence (AI-86)*, pages 19–43, Montreal, Canada, May.

[Hwang and Schubert1993] C.H. Hwang and L.K. Schubert, 1993. *Episodic Logic: A situational logic for natural language processing*, pages 307–452. CSLI.

[Hwang and Schubert1994] Chung Hee Hwang and Lenhart K. Schubert. 1994. Interpreting tense, aspect and time adverbials: A compositional, unified approach. In *Proceedings of the First International Conference on Temporal Logic*, ICTL '94, pages 238–264, London, UK, UK. Springer-Verlag.

[Hwang1992] Chung Hee Hwang. 1992. *A logical approach to narrative understanding.* Ph.D. thesis, University of Alberta.

[Jurafsky and Martin2009] D. Jurafsky and J.H. Martin. 2009. *Speech and language processing.* Pearson/Prentice Hall, NJ, 2nd edition.

[Kamp and Reyle1993] Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in Linguistics and Philosophy.* Springer, Dordrecht.

[Kamp1981] Hans Kamp. 1981. A theory of truth and semantic representation. In J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof, editors, *Formal Methods in the Study of Language*, volume 1, pages 277–322. Mathematisch Centrum, Amsterdam.

[Kate and Mooney2006] Rohit J. Kate and Raymond J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 913–920, Sydney, Australia, July. Association for Computational Linguistics.

[Kim and Schubert2016] Gene Kim and Lenhart Schubert. 2016. High-fidelity lexical axiom construction from verb glosses. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 34–44, Berlin, Germany, August. Association for Computational Linguistics.

[Kingsbury and Palmer2002] Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *In Language Resources and Evaluation*.

[Konstas et al.2017] Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada, July. Association for Computational Linguistics.

[Kwiatkowski et al.2011] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[Landes et al.1998] Shari Landes, Claudia Leacock, and Randee I. Tengi. 1998. Building semantic concordances. In Christiane Fellbaum, editor, *WordNet: A Lexical Reference System and its Application*, chapter 8. MIT Press, Cambridge, MA.

[Lewis and Steedman2014] Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar, October. Association for Computational Linguistics.

[Li and Roth2002] Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Liang et al.2011] Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 590–599, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Liang et al.2017] Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural Symbolic Machines: Learning semantic parsers on Freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33, Vancouver, Canada, July. Association for Computational Linguistics.

[Lien and Kouylekov2015] Elisabeth Lien and Milen Kouylekov. 2015. Semantic parsing for textual entailment. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 40–49, Bilbao, Spain, July. Association for Computational Linguistics.

[MacCartney and Manning2008] Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK, August. Coling 2008 Organizing Committee.

[Manshadi et al.2013] Mehdi Manshadi, Daniel Gildea, and James Allen. 2013. Plurality, negation, and quantification:towards comprehensive quantifier scope disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 64–72, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Mathhiessen and Bateman1991] C.M.I.M. Mathhiessen and J.A. Bateman. 1991. *Text Generation and Systemic-Functional Linguistics*. Pinter, London.

[Miller1995] George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, November.

[Minnen et al.2001] Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223, September.

[Mnih et al.2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik,

Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518.

[Morbini and Schubert2008] Fabrizio Morbini and Lenhart Schubert. 2008. Metareasoning as an integral part of commonsense and autocognitive reasoning. In *AAAI-08 Workshop on Metareasoning*.

[Morbini and Schubert2009] Fabrizio Morbini and Lenhart Schubert. 2009. Evaluation of Epilog: A reasoner for Episodic Logic. In *Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning*, Toronto, Canada, June.

[Morbini and Schubert2011] Fabrizio Morbini and Lenhart Schubert. 2011. Metareasoning as an Integral Part of Commonsense and Autocognitive Reasoning. In Michael T. Cox and Anita Raja, editors, *Metareasoning: Thinking about thinking*. MIT Press, January.

[Oepen et al.2004] Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, and Paul Meurer. 2004. Som å kapp-ete med trollet? - towards mrs-based norwegian-english machine translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 11–20.

[Oepen2001] Stephan Oepen. 2001. [incr tsdb()] – competence and performance laboratory. user manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany.

[OWL Working Group2004] W3C OWL Working Group. 2004. *OWL Web Ontology Language Guide*. W3C Recommendation. Available at `https://www.w3.org/TR/2004/REC-owl-guide-20040210`.

[Palmer et al.2005] Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31.

[Perera et al.2017] Ian Perera, James Allen, Lucian Galescu, Choh Man Teng, Mark Burstein, Scott Friedman, David McDonald, and Jeffrey Rye. 2017. Natural language dialogue for building and learning models and structures. In *AAAI Conference on Artificial Intelligence*.

[Poesio et al.2016] M. Poesio, R. Stuckardt, and Y. Versley, editors. 2016. *Anaphora Resolution: Algorithms, Resources, and Applications*. Springer.

[Pollard and Sag1994] C. Pollard and I. A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago/CLSI Publications.

[Poon2013] Hoifung Poon. 2013. Grounded unsupervised semantic parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 933–943, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Popescu et al.2004] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates. 2004. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Post and Gildea2009] Matt Post and Daniel Gildea. 2009. Bayesian learning of a tree substitution grammar. In *Proc. Association for Computational Linguistics (short paper)*, pages 45–48, Singapore.

[Rhee et al.2014] Hyekyun Rhee, James Allen, Jennifer Mammen, and Mary Swift. 2014. Mobile phone-based asthma self-management aid for adolescents (masmaa): a feasibility study. *Patient Preference and Adherence*, 8:63.

[Rudinger et al.2018] Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans, LA.

[Schneider et al.2015] Nathan Schneider, Jeffrey Flanigan, and Tim O'Gorman. 2015. The logic of amr: Practical, unified, graph-based sentence semantics for nlp. NAACL HLT Tutorial.

[Schubert and Hwang2000] Lenhart K. Schubert and Chung Hee Hwang. 2000. Episodic logic meets little red riding hood: A comprehensive natural representation for language understanding. In Lucja M. Iwańska and Stuart C. Shapiro, editors, *Natural Language Processing and Knowledge Representation*, pages 111–174. MIT Press, Cambridge, MA, USA.

[Schubert and Pelletier1982] L.K. Schubert and F.J. Pelletier. 1982. From English to logic: Context-free computation of 'conventional' logical translations. *Am. J. of Computational Linguistics 8 [now Computational Linguistics]*, pages 26–44.

[Schubert2000] Lenhart K. Schubert. 2000. The situations we talk about. In Jack Minker, editor, *Logic-based Artificial Intelligence*, pages 407–439. Kluwer Academic Publishers, Norwell, MA, USA.

[Schubert2013] Lenhart Schubert. 2013. NLog-like inference and commonsense reasoning. In A. Zaenen, V. de Paiva, and C. Condoravdi, editors, *Perspectives on Semantic Representations for Textual Inference, special issue of Linguistic Issues in Language Technology (LiLT 9)*, volume 9, pages 1–26.

[Schubert2014] Lenhart Schubert. 2014. From treebank parses to Episodic Logic and commonsense inference. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 55–60, Baltimore, MD, June. Association for Computational Linguistics.

[Schubert2015] Lenhart Schubert. 2015. Semantic representation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4132–4138. AAAI Press.

[Schuler2006] Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

[Steedman2000] Mark Steedman. 2000. *The Syntactic Process*. The MIT Press.

[Stratos et al.2011] Karl Stratos, Lenhart K. Schubert, and Jonathan Gordon. 2011. Episodic Logic: Natural Logic + reasoning. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD)*.

[Tellex et al.2011] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI Conference on Artificial Intelligence*.

[Tiedemann2012] Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May. European Language Resources Association (ELRA).

[White et al.2016] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas, November. Association for Computational Linguistics.

[Williams1992] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, May.

[Wolf2005] Florian Wolf. 2005. *Coherence in natural language : data structures and applications*. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Brain and Cognitive Sciences.

[Yao and Zhang2010] Xuchen Yao and Yi Zhang. 2010. Question generation with minimal recursion semantics. In *Proceedings of the Third Workshop on Question Generation: Shared Task Evaluation Challenge*.

[Ytrstøl et al.2009] Gisle Ytrstøl, Dan Flickinger, and Stephan Oepen. 2009. Extracting and annotating Wikipedia sub-domains. In *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7)*, pages 185–196, Groningen, The Netherlands.

[Zhang and Krieger2011] Yi Zhang and Hans-Ulrich Krieger. 2011. Large-scale corpus-driven pcfg approximation of an hpsg. In *Proceedings of the 12th International Conference on Parsing Technologies*, ICPT '11, pages 198–208, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Zhang et al.2017] Sheng Zhang, Rachel Rudinger, and Ben Van Durme. 2017. An evaluation of predpatt and open ie via stage 1 semantic role labeling. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*, Montpellier, France, September.

[Zhou et al.2016] Junsheng Zhou, Feiyu Xu, Hans Uszkoreit, Weiguang QU, Ran Li, and Yanhui Gu. 2016. Amr parsing with an incremental joint model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 680–689. Association for Computational Linguistics.