Corpus Annotation and Inference with Episodic Logic Type Structure

Gene Kim

May 2, 2018
Language understanding is a growing area of interest in NLP.
Language understanding is a growing area of interest in NLP.

**QA:** AI2 Reasoning Challenge, RACE, bAbI, SQuAD, TriviaQA, NarrativeQA, FreebaseQA, WebQuestions,...

**Dialogue:** Amazon Alexa Challenge, work on Google Home and Microsoft Cortana

**Inference:** JOCI, SNLI, MultiNLI

**Semantic Parsing:** AMR

**Others:** GLUE
ULF Annotation Project

Project: Annotate a large, topically varied dataset of sentences with unscoped logical form (ULF) representations.

- ULF: captures semantic type structure and retains scoping and anaphoric ambiguity.

Goal: Train a reliable, general-purpose ULF transducer on the corpus.
ULF Annotation Project

Project: Annotate a large, topically varied dataset of sentences with unscoped logical form (ULF) representations.

- ULF: captures semantic type structure and retains scoping and anaphoric ambiguity.

Goal: Train a reliable, general-purpose ULF transducer on the corpus.

Example Annotation

“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 ((pres debate.v)
   (the.d (n+preds fact.n
    (= (that ((the.d |Earth|.n)
      ((pres prog) heat_up.v))))))))))
```
Hypotheses of Proposal

1. A divide-and-conquer approach to semantic parsing will ultimately lead to more precise and useful representations for reasoning over language.

2. An expressive logical representation with model-theoretic backing will enable reasoning capabilities that are not offered by other semantic representations available today.

3. Better language understanding and reasoning systems can be built by combining the strengths of statistical systems in converting raw signals to structured representations and symbolic systems in performing precise and flexible manipulations over complex structures.
Short Introduction of ULF

“Alice thinks that John nearly fell”
“He neglected three little bushes”

ULF
(|Alice| (((pres think.v) (that (|John| (nearly.adv-a (past fall.v)))))))
(he.pro ((past neglect.v) (three.d (little.a (plur bush.n)))))
Short Introduction of ULF

“Alice thinks that John nearly fell”
“He neglected three little bushes”

**ULF**

\[
(|\text{Alice}| \ ((\text{pres} \ \text{think.v}) \ (\text{that} \ (|\text{John}| \ (\text{nearly.adv-a} \ (\text{past} \ \text{fall.v})))))

(he.pro \ ((\text{past} \ \text{neglect.v}) \ (\text{three.d} \ (\text{little.a} \ (\text{plur} \ \text{bush.n})))))
\]

*Syntax-like*

**Nouns:** bush.n
**Verbs:** think.v, fall.v, neglect.v
**Adjectives:** little.a
**Adverbs:** nearly.adv
**Pronouns:** he.pro
**Names:** |Alice|, |John|
**Determiners:** three.d
Short Introduction of ULF

“Alice thinks that John nearly fell”
“He neglected three little bushes”

ULF
(|Alice| (((pres think.v) (that (|John| (nearly.adv-a (past fall.v)))))
(he.pro ((past neglect.v) (three.d (little.a (plur bush.n))))))

Syntax-like
Nouns: bush.n
Verbs: think.v, fall.v, neglect.v
Adjectives: little.a
Adverbs: nearly.adv
Pronouns: he.pro
Names: |Alice|, |John|
Determiners: three.d

Formal
Domain: $\mathcal{D}$, Situations: $\mathcal{S}$, Truth-value: 2
$\mathcal{N}: \mathcal{D} \rightarrow (\mathcal{S} \rightarrow 2)$
Individual constant($\mathcal{D}$): |Alice|, |John|
Individual variable($\mathcal{D}$): he.pro
$n$-place predicate($\mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow 2)$):
bush.n, think.v, fall.v, neglect.v, little.a
Predicate modifier($\mathcal{N} \rightarrow \mathcal{N}$): nearly.adv
Modifier constructor($\mathcal{N} \rightarrow (\mathcal{N} \rightarrow \mathcal{N})$): attr
Sentence nominalizer((($\mathcal{S} \rightarrow 2) \rightarrow \mathcal{D}$)): that
Role of ULF in Comprehensive Semantic Interpretation

“The boy wants to go”
Role of ULF in Comprehensive Semantic Interpretation

“The boy wants to go”

ULF

((the.d boy.n) ((pres want.v) (to go.v)))
“The boy wants to go”

ULF

((the.d boy.n) ((pres want.v) (to go.v)))

Scoping

(pres (the.d x (x boy.n) (x (want.v (to go.v)))))
Role of ULF in Comprehensive Semantic Interpretation

“The boy wants to go”

ULF

((the.d boy.n) ((pres want.v) (to go.v)))

Scoping

(pres (the.d x (x boy.n) (x (want.v (to go.v)))))

Deindexing

(|E|.sk at-about.p |Now17|)

((the.d x (x boy.n) (x (want.v (to go.v))))) ** |E|.sk)
“The boy wants to go”

ULF

((the.d boy.n) ((pres want.v) (to go.v)))

Scoping

(pres (the.d x (x boy.n) (x (want.v (to go.v)))))

Deindexing

(|E|.sk at-about.p |Now17|)

((the.d x (x boy.n) (x (want.v (to go.v))))) ** |E|.sk)

Coreference

(|E|.sk at-about.p |Now17|)

(((Manolin| (want.v (to go.v))) ** |E|.sk)
Inference using ULFs

Phrase structure + Coherent types

Everyone in the audience has been enjoying the sunny weather.

Len has been enjoying the sunny weather.

Implicative, attitudinal, and communicative verbs

He managed to quit smoking.

He quit smoking.

Counterfactuals

Gene wishes people liked to go out to eat ice cream in the winter.

People don’t like to go out to eat ice cream in the winter.

Questions and requests

When are you getting married?

You are getting married in the foreseeable future
Inference using ULFs

Phrase structure + Coherent types

- **Generalization/specializations**
  
  *Everyone in the audience has been enjoying the sunny weather.*
  
  $\rightarrow$ *Len has been enjoying the sunny weather.*
Inference using ULFs

Phrase structure + Coherent types

- Generalization/specializations
  Everyone in the audience has been enjoying the sunny weather.
  → Len has been enjoying the sunny weather.

- Implicative, attitudinal, and communicative verbs
  He managed to quit smoking. → He quit smoking.
Inference using ULFs

Phrase structure + Coherent types

- **Generalization/specializations**
  > Everyone in the audience has been enjoying the sunny weather.
  > → Len has been enjoying the sunny weather.

- **Implicative, attitudinal, and communicative verbs**
  > He managed to quit smoking. → He quit smoking.

- **Counterfactuals**
  > Gene wishes people liked to go out to eat ice cream in the winter.
  > → People don’t like to go out to eat ice cream in the winter.
Inference using ULFs

Phrase structure + Coherent types

- **Generalization/specializations**
  
  *Everyone in the audience has been enjoying the sunny weather.*
  
  → *Len has been enjoying the sunny weather.*

- **Implicative, attitudinal, and communicative verbs**

  *He managed to quit smoking.*  →  *He quit smoking.*

- **Counterfactuals**

  *Gene wishes people liked to go out to eat ice cream in the winter.*
  
  → *People don’t like to go out to eat ice cream in the winter.*

- **Questions and requests**

  *When are you getting married?*  
  
  → *You are getting married in the foreseeable future*
The advantages of our chosen representation include:

- It is not so far removed from constituency parses, which can be precisely generated.

- It enables principled analysis of structure and further resolution of ambiguous phenomena. Full pipeline exists for understanding children’s books.

- It enables structural inferences, which can be generated spontaneously (forward inference).
Outline

1 Introduction

2 Survey of Related Work
   - TRIPS
   - The JHU Decompositional Semantics Initiative
   - Parallel Meaning Bank
   - LinGO Redwoods Treebank
   - Abstract Meaning Representation

3 Research Project Description and Progress
   - Motivation - Lexical Axiom Extraction in EL
   - Annotation Environment and Corpus Building
   - Corpus Building
   - Learning a Statistical Parser
   - Evaluating the Parser
1 Introduction

2 Survey of Related Work
   - TRIPS
   - The JHU Decompositional Semantics Initiative
   - Parallel Meaning Bank
   - LinGO Redwoods Treebank
   - Abstract Meaning Representation

3 Research Project Description and Progress
   - Motivation - Lexical Axiom Extraction in EL
   - Annotation Environment and Corpus Building
   - Corpus Building
   - Learning a Statistical Parser
   - Evaluating the Parser
The TRIPS Parser

- Generates parses in underspecified semantic representation with scoping constraints
- Node grounded in an ontology
- Uses a bottom-up chart parser with a hand-built grammar, a syntax-semantic lexicon tied to an ontology, and preferences from syntactic parsers and taggers
- Deployed in multiple tasks with minimal modifications

Figure 1: Parse for “They tried to find the ice bucket” using the vanilla dialogue model of TRIPS.
TRIPS Logical Form (Allen et al., 2008) descriptively covers a lot of language phenomena (e.g. generalized quantifiers, lambda abstractions, dialogue semantics, thematic roles).
TRIPS Logical Form (Allen et al., 2008) descriptively covers a lot of language phenomena (e.g. generalized quantifiers, lambda abstractions, dialogue semantics, thematic roles).

Formally, TRIPS LF is an underspecified semantic representation which subsumes Minimal Recursion Semantics and Hole Semantics (Allen et al., 2018).
TRIPS Logical Form (Allen et al., 2008) descriptively covers a lot of language phenomena (e.g. generalized quantifiers, lambda abstractions, dialogue semantics, thematic roles).

Formally, TRIPS LF is an underspecified semantic representation which subsumes Minimal Recursion Semantics and Hole Semantics (Allen et al., 2018).

- Easy to manage underspecification
- Computationally efficient
- Flexible to different object languages
- At present there are no direct, systematic inference methods for TRIPS LF
Decomp

Building up a model of language semantics through user annotations of focused phenomena.

- Quick and easy to judge by every day users
- Train precise model on large corpus
- Build up general model of semantics distinction at a time

So far investigated

- Predicate-argument extraction (White et al., 2016)
- Semantic proto-roles for discovering thematic roles (Reisinger et al., 2015)
- Selection behavior of clause-embedding verbs
- Event factuality (Rudinger et al., 2018)
PredPatt (White et al., 2016) lays a foundation for this as a minimal predicate-argument structure. Built on top of universal dependencies.

PredPatt extracts predicates and arguments from text.

\[
?a \text{ extracts } ?b \text{ from } ?c
\]

\[
?a: \text{ PredPatt}
\]

\[
?b: \text{ predicates}
\]

\[
?c: \text{ text}
\]

\[
?a \text{ extracts } ?b \text{ from } ?c
\]

\[
?a: \text{ PredPatt}
\]

\[
?b: \text{ arguments}
\]

\[
?c: \text{ text}
\]
Decomp

PredPatt (White et al., 2016) lays a foundation for this as a minimal predicate-argument structure. Built on top of universal dependencies.

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

Model and theory agnostic
Parallel Meaning Bank

- Annotates full documents
- Human-aided machine annotations
- 2,057 English sentences so far
- Discourse representation structures
Parallel Meaning Bank

- Annotates full documents
- Human-aided machine annotations
- 2,057 English sentences so far
- Discourse representation structures

Discourse Representation Structures

- Anaphora resolution
- Discourse structures
- Presupposition
- Donkey anaphora
- Mappable to FOL

Donkey Anaphora

Every child who owns a dog loves it.
Figure 2: Screenshot of the PMB Explorer with analysis of the sentence “The farm grows potatoes.”
PMB Assessment

Pros
- Natively handles discourses.
- Sufficient annotation speed for corpus construction.
- Formally interpretable representation which can be used with FOL-theorem provers.

Cons
- Insufficient formal expressivity for natural language.
- Approach requires a large amount of engineering – automatic generation which is integrated with a highly-featured annotation editor.
- Hand-engineered grammars do not scale well to addition of linguistic phenomena.
The LinGO Redwoods Treebank:

- HPSG grammar and Minimal Recursion Semantics representation
- Hand-built grammar (ERG)
- Semi-manually annotated by pruning parse forest
- 87% of a 92,706 sentence dataset annotated
Redwoods Treebank Project

The LinGO Redwoods Treebank:
- HPSG grammar and Minimal Recursion Semantics representation
- Hand-built grammar (ERG)
- Semi-manually annotated by pruning parse forest
- 87% of a 92,706 sentence dataset annotated

Minimal Recursion Semantics (MRS):
- Flat semantic representation
- Designed for underspecification
- MRS used as a meta-language for ERG – does not define object-language semantics.

Figure 3: Example of the sentence “Do you want to meet on Tuesday” in simplified, dependency graph form. Example from Oepen et al. (Oepen et al., 2002).
Treebanking

1. Generate candidate parses using an HPSG parser.
2. Prune parse forest to a single candidate using discriminants.
3. Accept or reject this parse.

Discriminants are saved for treebank updates.

The corpus includes WSJ, MT, and dialogue corpora.

Figure 4: Screenshot of Redwoods treebanking environment for the sentence “I saw a black and white dog.”
The ERG performance is a result of years of improvement.

<table>
<thead>
<tr>
<th>Processing Stage</th>
<th>Stage Coverage</th>
<th>Running Total Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical Coverage</td>
<td>32%</td>
<td>32%</td>
</tr>
<tr>
<td>Able to Generate Parse</td>
<td>57%</td>
<td>18%</td>
</tr>
<tr>
<td>Contains Correct Parse</td>
<td>83%</td>
<td>15%</td>
</tr>
</tbody>
</table>

**Table 1**: Early stage ERG performance on the BNC in 2003.
The ERG performance is a result of years of improvement.

<table>
<thead>
<tr>
<th>Processing Stage</th>
<th>Stage Coverage</th>
<th>Running Total Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical Coverage</td>
<td>32%</td>
<td>32%</td>
</tr>
<tr>
<td>Able to Generate Parse</td>
<td>57%</td>
<td>18%</td>
</tr>
<tr>
<td>Contains Correct Parse</td>
<td>83%</td>
<td>15%</td>
</tr>
</tbody>
</table>

Table 1: Early stage ERG performance on the BNC in 2003.

Years of grammar improvement was critical for annotation success!
Abstract Meaning Representation

- Unified, graphical semantic representation based on PropBank arguments
- Canonicalized representation of meaning
- One-shot approach to capturing representation
- Editor with unix-style text commands for annotating
- 47,274 sentences annotated
- Formally equivalent to FOL w/o quantifiers

Logical format
\[ \exists w, g, b: \]
\[ \text{instance}(w, \text{want-01}) \land \text{instance}(g, \text{girl}) \land \]
\[ \text{instance}(b, \text{believe-01}) \land \]
\[ \text{arg0}(w, g) \land \text{arg1}(w, b) \land \text{arg0}(b, g) \]

AMR format
\[
(w / \text{want-01}
\quad :\text{arg0} (g / \text{girl})
\quad :\text{arg1} (b / \text{believe-01}
\quad :\text{arg0} g))
\]

Graph format

Figure 5: AMR representations for “The girl wanted to believe herself”. 
AMR Assessment

Pros
- Wide linguistic coverage.
- Sufficient annotation speed for corpus construction.

Cons
- Insufficient formal expressivity for natural language.
- Over-canonicalization for nuanced inference.

AMR-equivalent sentences (Bender et al., 2015)
- No one ate.
- Every person failed to eat.
- Dropping of tense, aspect, grammatical number, and more.
1 Introduction

2 Survey of Related Work
   - TRIPS
   - The JHU Decompositional Semantics Initiative
   - Parallel Meaning Bank
   - LinGO Redwoods Treebank
   - Abstract Meaning Representation

3 Research Project Description and Progress
   - Motivation - Lexical Axiom Extraction in EL
   - Annotation Environment and Corpus Building
   - Corpus Building
   - Learning a Statistical Parser
   - Evaluating the Parser
slam2.v
Gloss: “strike violently”
Frames: [Somebody slam2.v Something]
Examples: “slam the ball”

Axiom:
(∀x,y,e: [[x slam2.v y] ** e]
→ [[[x (violently1.adv (strike1.v y))] ** e]
and [x person1.n] [y thing12.n]])

- EL axioms from WordNet verb entries
- Rule-based system
- Generated lexical KB is competitive in a lexical inference task.
- Error analysis shows need for a better EL transducer
1. Annotation Environment and Corpus Building

2. Learning a Statistical Parser

3. Evaluating the Parser
First Pilot Annotations

Fall 2016

- Simple graph-building annotation tool inspired by the AMR Editor.
- Each annotated between 27 and 72 sentences.
- ULF ann. speed ≈ AMR ann. speed.

<table>
<thead>
<tr>
<th>Annotator</th>
<th>Minutes/Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beginner</td>
<td>12.67</td>
</tr>
<tr>
<td>Beginner (- first 10)</td>
<td>6.83</td>
</tr>
<tr>
<td>Intermediate</td>
<td>7.70</td>
</tr>
<tr>
<td>Expert</td>
<td>6.87</td>
</tr>
</tbody>
</table>

Table 2: Average timing of experimental ULF annotations.

Figure 6: Timing results from ULF experimental annotations.
Agreement of annotations was 0.48 :(
First Pilot Annotations - Limitations

Agreement of annotations was 0.48 :

Discrepancy sources (in order of severity):

1. Movement of large phrases, such as prepositional modifiers.
2. Ill-formatted text, such as fragments.
3. Some language phenomena were not carefully discussed in the preliminary guidelines.
Towards Simpler Annotations


2. To preserve surface word order and simplify annotations, we extend ULF.
   - Relaxation of well-formedness constraints
   - Lexical marking of scope
   - Introduction of syntactic macros
Second Pilot Annotations

Fall 2017
2 experts, 6 beginners

Changes from first pilot annotations:
- Layer-wise annotations, direct writing
- Introduction of ULF relaxations and macros
- Further development of ULF guidelines
- Shared annotation view
- Annotated Tatoeba rather than Brown corpus
Second Pilot Annotations

Fall 2017
2 experts, 6 beginners

Changes from first pilot annotations:
- Layer-wise annotations, direct writing
- Introduction of ULF relaxations and macros
- Further development of ULF guidelines
- Shared annotation view
- Annotated Tatoeba rather than Brown corpus

Annotation Count
- 270 sentence annotated
- 80 annotations timed
Second Pilot Annotations

Fall 2017
2 experts, 6 beginners

Changes from first pilot annotations:
- Layer-wise annotations, direct writing
- Introduction of ULF relaxations and macros
- Further development of ULF guidelines
- Shared annotation view
- Annotated Tatoeba rather than Brown corpus

Annotation Count
- 270 sentence annotated
- 80 annotations timed

Annotation Speeds
- 8 min/sent overall
- 4 min/sent for experts
- 11 min/sent for non experts
Second Pilot Annotations

Fall 2017
2 experts, 6 beginners

Changes from first pilot annotations:
- Layer-wise annotations, direct writing
- Introduction of ULF relaxations and macros
- Further development of ULF guidelines
- Shared annotation view
- Annotated Tatoeba rather than Brown corpus

Annotation Count
270 sentence annotated
80 annotations timed

Annotation Speeds
8 min/sent overall
4 min/sent for experts
11 min/sent for non experts
Relaxing ULF Constraints

We can allow omission of type-shifters from predicates to predicate-modifiers for certain pairs of types.

- nn - noun to noun modifier
- nnp - noun phrase to noun modifier
- attr - adjective to noun modifier
- adv-a - any predicate to monadic verb/adjective modifier
Relaxing ULF Constraints

We can allow omission of type-shifters from predicates to predicate-modifiers for certain pairs of types.

- **nn** - noun to noun modifier
- **nnp** - noun phrase to noun modifier
- **attr** - adjective to noun modifier
- **adv-a** - any predicate to monadic verb/adjective modifier

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

```plaintext
((burning.a hot.a) (melting.n pot.n))
```
Lexical Scope Marking

Add a lexical marker for scoping position rather than lifting.

<table>
<thead>
<tr>
<th>Sentences</th>
<th>Mary confidently spoke up</th>
<th>Mary undoubtedly spoke up</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Without Lexical Marking</strong></td>
<td>(</td>
<td>Mary</td>
</tr>
<tr>
<td><strong>With Lexical Marking</strong></td>
<td>(</td>
<td>Mary</td>
</tr>
</tbody>
</table>
Lexical Scope Marking

Add a lexical marker for scoping position rather than lifting.

| Sentences                          | Mary confidently spoke up  
|                                   | Mary undoubtedly spoke up  
| Without Lexical Marking           | (|Mary| (confidently.adv (past speak_up.v)))  
|                                   | (undoubtedly.adv (|Mary| (past speak_up.v)))  
| With Lexical Marking              | (|Mary| (confidently.adv-a (past speak_up.v)))  
|                                   | (|Mary| (undoubtedly.adv-s (past speak_up.v)))  

Stays close to constituency bracketing
Sentence:  *Muiriel is 20 now*  
Bracketing: (Muiriel ((is 20) now))  
Full ULF: (|Muiriel| (((pres be.v) 20.a) now.adv-e))
Similar to C-macros, but accompanied by a few specially interpreted items.

Post-nominal modifiers

\[(n+p\text{reds } N \text{ Pred1 Pred2 } \ldots \text{ PredN}) \equiv \]
\[(\lambda x ((x \text{ N}) \text{ and } (x \text{ Pred1}) (x \text{ Pred2}) \ldots (x \text{ PredN})))\]

\[(n+p\text{reds } NP \text{ Pred1 Pred2 } \ldots \text{ PredN}) \equiv \]
\[(\text{the.d } (\lambda x ((x = \text{ NP}) \text{ and } (x \text{ Pred1}) (x \text{ Pred2}) \ldots (x \text{ PredN}))))\]
Similar to C-macros, but accompanied by a few specially interpreted items.

Post-nominal modifiers

\[(n+\text{preds } N \text{ Pred1 Pred2 ... PredN}) \equiv (\lambda x ((x N) \text{ and } (x \text{ Pred1 }) (x \text{ Pred2}) ... (x \text{ PredN})))\]

\[(n+\text{preds } NP \text{ Pred1 Pred2 ... PredN}) \equiv (\text{the.d } (\lambda x ((x = NP) \text{ and } (x \text{ Pred1 }) (x \text{ Pred2}) ... (x \text{ PredN}))))\]

*The table by the fireplace with three legs*

\[(\text{the.d } (n+\text{preds } \text{table.n } \text{by.p } (\text{the.d } \text{fireplace.n})))\]

\[(\text{with.p } ((\text{nquan } \text{three.a}) \text{ (plur } \text{leg.n})))\]
Relative Clauses

\[(\text{sub } C \ S[*h]) \equiv \ S[*h \leftarrow C]\]

\[S_{emb}[\text{that.rel}] \equiv (\lambda \ *r \ S_{emb}[\text{that.rel} \leftarrow \ *r])\]

*car that you bought*

\[(\text{n+preds } \text{car.n} \ (\text{sub } \text{that.rel} \ (\text{you.pro} \ ((\text{past } \text{buy.v}) \ *h))))\]
Relative Clauses
(sub C S[*h]) ≡ S[*h←C]
S_{emb}[that.rel] ≡ (λ *r S_{emb}[that.rel←*r])

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

\[
\downarrow\text{n+preds}
\]
(λ x ((x car.n) (x (sub that.rel (you.pro ((past buy.v) *h)))))})
Relative Clauses

\[(\text{sub } C \text{ S}[*h]) \equiv \text{S}[*h\leftarrow C]\]

\[S_{emb}[\text{that}.\text{rel}] \equiv (\lambda *r \text{ } S_{emb}[\text{that}.\text{rel}\leftarrow *r])\]

*car that you bought*

\[\text{(n+preds car.n (sub that.rel (you.pro ((past buy.v) *h))))}\]

\[\text{n+preds}\]

\[\text{(\lambda x (((x car.n) (x (sub that.rel (you.pro ((past buy.v) *h)))))}}\]

\[\text{sub}\]

\[\text{(\lambda x (((x car.n) (x (you.pro ((past buy.v) that.rel)))))}}\]
Macros

Relative Clauses

\((\text{sub } C \ S[*h]) \equiv S[*h{\leftarrow}C]\)

\(S_{emb}[\text{that.rel}] \equiv (\lambda \ *r \ S_{emb}[\text{that.rel}{\leftarrow}*r])\)

car that you bought

\((n+\text{preds} \ \text{car.n} \ (\text{sub} \ \text{that.rel} \ (\text{you.pro} \ ((\text{past buy.v}) \ *h))))\)

\(n+\text{preds}

(\lambda \ x \ ((x \ \text{car.n}) \ (x \ (\text{sub} \ \text{that.rel} \ (\text{you.pro} \ ((\text{past buy.v}) \ *h))))))\)

\(\text{sub}

(\lambda \ x \ ((x \ \text{car.n}) \ (x \ (\text{you.pro} \ ((\text{past buy.v}) \ \text{that.rel}))))))\)

\(\text{that.rel}

(\lambda \ x \ ((x \ \text{car.n}) \ and \ (x \ (\lambda \ *r \ (\text{you.pro} \ ((\text{past buy.v}) \ *r))))))\)
Relative Clauses

(sub C S[*h]) ≡ S[*h←C]

$S_{emb}[\text{that.rel}] ≡ (\lambda *r S_{emb}[\text{that.rel←*r}])$

car that you bought

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

→ n+preds

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

→ sub

(λ x ((x car.n) (x (you.pro ((past buy.v) that.rel))))))

→ that.rel

(λ x ((x car.n) and (x (\lambda *r (you.pro ((past buy.v) *r))))))

→ λ-conversion

(λ x ((x car.n) and (you.pro ((past buy.v) x))))
**Prenominal Possessive**

\[
((\text{NP} 's) \ N) \equiv (\text{the}.d \ ((\text{poss-by} \ \text{NP}) \ N))
\]

Example: \[((|\text{John}| 's) \ \text{dog}.n) \equiv (\text{the}.d \ ((\text{poss-by} \ |\text{John}|) \ \text{dog}.n))\]

**Possessive Determiners**

\[
(\text{my}.d \ N) \leftrightarrow (\text{the}.d \ ((\text{poss-by} \ \text{me}.pro) \ N)),
\]

where \text{my}.d and \text{me}.pro can be replaced by any corresponding pair of possessive determiner and personal pronoun.

**Under development**

Comparatives, Superlatives, Questions, Gaps, Discourse Markers
Plan to make major progress in annotations this summer with a handful of annotators. Try to get 3,000 annotations (cf. initial AMR corpus of 10,000 with 12 annotators for 3 months) primarily from Tatoeba dataset.

Current annotator state:

- 2-layer annotation
- Simple syntax and bracket highlighting
- Standalone reference for modals
- Quick-reference of examples from guidelines

Figure 7: Current ULF annotator state with example annotation process.
Current Annotator State

This page is an excerpt from the annotation guidelines listing the different modals:

- **can**
  - (pres can.aux-v)
    - if it means something like "presently able to"
      - "This rocket can reach Mars"
  - (pres can.aux-s)
    - if it simply refers to a possibility
      - "This mission can fail"

- **could**
  - (pres could.aux-v)
    - if it means something like "presently able to"
      - "I could easily climb over that fence"
  - (pres could.aux-s)
    - if it simply refers to a possibility
      - "The sea level could rise"
  - (past can.aux-v)
    - if it means roughly "able-to in the past"
      - "Pterodactyls could fly"
  - (past can.aux-s)
    - if it refers to a possibility from a past perspective
      - "He was well aware that he could fail"

- **may**
  - (pres may.aux-v)
    - if it means something like "presently permitted to"
      - "You may sit down"
  - (pres may.aux-s)
    - if it simply refers to a possibility
      - "The prisoner may escape"

- **might**
  - (pres might.aux-s)
    - if it simply refers to a possibility
      - "He might fail"
  - (past may.aux-s)
    - if it refers to a possibility from a past perspective

**Figure 8:** Screenshot of modals reference.
Current Annotator State

Figure 8: Screenshot of modals reference.

Figure 9: Screenshot of sanity checker output.

Sanity checking formula (after preprocessing):

\{(ONE.D ((PRES CAN.AUX-V) (FIND.V TIME.N)) \.) (ALWAYS.ADV-E)\}

Formula with predicted types:

\{(TYPES UNKNOWN) (ONE.D (TYPES PRED TENSED-VERB) ((TYPES TENSED-AUX) (PRES CAN.AUX-V)) ((TYPES VERB PRED) (FIND.V TIME.N))) \.\}

\{(TYPES UNKNOWN) (ALWAYS.ADV-E))\}

Ann segment: (ONE.D ((PRES CAN.AUX-V) (FIND.V TIME.N)) \.)

Predicted constituent types ((list of types) -- constituent)

((DET) -- ONE.D) ((PRED TENSED-VERB) -- ((PRES CAN.AUX-V) (FIND.V TIME.N)))

((SENT-PU NT) -- \.\)

Possibly failed conditions:

"Determiners take 1 nominal (noun) argument."

"Sentence punctuation take a single tensed sentence argument and is post-fixed."

NIL
In choosing our approach training a parser, we'll take advantage of everything we can. Here are some major features of the ULF parsing task.

- **Relatively small dataset size**
  
  $\leq 10,000$ sentences

- **Known restrictions in target type structure**
  
  (k he.pro) not allowed!

- **Close to constituent parse and surface form**

- **Enables structured inferences**
Learning a Statistical Parser

In choosing our approach training a parser, we'll take advantage of everything we can. Here are some major features of the ULF parsing task.

- **Relatively small dataset size**
  <10,000 sentences

- **Known restrictions in target type structure**
  (k he.pro) not allowed!

- **Close to constituent parse and surface form**

- **Enables structured inferences**

We propose using tree-to-tree machine translation method or a string-to-tree parsing method with further refinement using reinforcement learning on inference tasks.

**Figure 10:** Performance of neural vs phrase-based MT systems as a function of data size (Koehn and Knowles, 2017).
Tree-to-tree Method

Generate the constituency tree and the ULF in parallel using a Synchronous Tree Substitution Grammar (STSG) (Eisner, 2003; Gildea, 2003).
Tree-to-tree Method

Generate the constituency tree and the ULF in parallel using a Synchronous Tree Substitution Grammar (STSG) (Eisner, 2003; Gildea, 2003).

STSG learning steps:

1. **Align nodes between the two trees**
   - Can apply heuristic priors via Variational Bayes, e.g. string matching and lexical types

2. **Learning multi-node rules between the two trees**
   - Can speed up with rule-decomposition sampling with a Bayesian prior on rule size (Post and Gildea, 2009; Chung et al., 2014).

STSG rules

\[ X \Rightarrow a, b \]

\[ X \Rightarrow a_1 X^{[1]} a_2 X^{[2]} a_3, b_1 X^{[2]} b_2 X^{[1]} b_3 \]
Figure 11: Rules for the example sentence For John to sleep in is unusual.
Given the minimal reordering between surface English and ULFs, we may be able to use PCFGs directly. Just like standard constituent parsing.

- **Minor extensions to ULF compositions to handle reordering**,
  e.g. \( \text{Formula} \rightarrow \text{Term}, \text{VPred} \) and \( \text{Formula}' \rightarrow \text{VPred}, \text{Term} \) for reordered variants.

- **Much more computationally efficient**

- **Can use known type-restrictions for model initialization**
Fine-tuning to a task can overcome both limitations in annotated corpus size and differences between the optimal trade-offs for the corpus learning and the task. For log-linear models we can use the REINFORCE algorithm (Williams, 1992) to tune to a particular task by propagating the signal back through the model to maximize expected reward.
Fine-tuning to a task can overcome both limitations in annotated corpus size and differences between the optimal trade-offs for the corpus learning and the task.

For log-linear models we can use the Reinforce algorithm (Williams, 1992) to tune to a particular task by propagating the signal back through the model to maximize expected reward.

**Reinforce Optimization and Update Functions**

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

- $X$: the set of inputs
- $\theta$: model parameters
- $y$: the output
- $\alpha, \beta$: hyperparameters for the convergence rate
Intrinsic Evaluations

- Evaluate the parser against a test set of the gold corpus annotations using a metric similar to smatch.
- Gives partial credit for each correct constituent of predication.
- EL-smatch developed for fully interpreted EL. We need to develop a modified version for ULF.

Extrinsic Evaluations

- Evaluate on inference tasks that require structural representations, but minimal world knowledge: implicatives, counterfactuals, questions, requests.
- Evaluate on Natural Logic-like inferences.
- Integrate the ULF parser into EL-based systems, e.g. lexical axiom acquisition
We performed a small pilot demonstration of inference over ULF last fall.

- Requests & counterfactuals
  
  *Can you call again later?*
  
  → *I want you to call again later*

  *If we knew what we were doing, it would not be called research*
  
  → *We don’t know what we’re doing*

- Inference engine built on 10 development sentences

- Sentence annotation and inference engine development done by separate people

- Evaluated on 136 ULFs
  
  65 from uniformly sampled sentences
  
  71 from keyword-based sampled sentences.
Table 3: 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.

\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
Sample & \# sent. & \# inf. & Corr. & Context\textsuperscript{a} & Incorr. & Precision\textsuperscript{b} & Recover\textsuperscript{c} & Precision\textsuperscript{d} \\
\hline
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 \\
\hline
\end{tabular}
\end{table}

\textsuperscript{a}Correctness is contextually dependent (e.g. “Can you throw a fastball?” → “I want you to throw a fastball.”).

\textsuperscript{b}[assuming context is wrong]/[assuming context is right] for context dependent inferences.

\textsuperscript{c}Recoverable with no loss of correct inferences.

\textsuperscript{d}Precision after loss-less recoveries.
Currently extending pilot inference to a larger and more varied dataset with more rigorous data collection methods.

- Attitudinal, counterfactual, request, and question inference.
  "Oprah is shocked that Obama gets no respect"
  → *Obama gets no respect*
  "When is your wedding?"
  → *You are getting married in the near future*
The phenomena we’re interested in are common, but relatively low-frequency. To reduce the annotator burden we perform pattern-based sentence filtering.

- Designed to minimize assumptions about the data we’re interested in.
- Hand-built tokenizers, sentence-delimiters, and sampling patterns for generating dataset. Take advantage of dataset features.
  
  e.g. *Discourse Graphbank end-of-sentence always triggers a newline, though not every newline is an end-of-sentence.*

- Syntactically augmented regex patterns.

  "<begin?><If><mid><was|were|had|<past>|<ppart><mid?><<futr>)+"  
  "<begin?><<futr><mid><if><mid><was|were|had|<past>|<ppart>)+"
## Sampling Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>impl</th>
<th>ctrlf1</th>
<th>request</th>
<th>question</th>
<th>interest</th>
<th>ignored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disc. Grphbnk</td>
<td>1,987</td>
<td>110</td>
<td>2</td>
<td>47</td>
<td>2,030</td>
<td>1,122</td>
</tr>
<tr>
<td>Proj. Gutenberg</td>
<td>264,109</td>
<td>31,939</td>
<td>2,900</td>
<td>60,422</td>
<td>303,306</td>
<td>275,344</td>
</tr>
<tr>
<td>Switchboard</td>
<td>37,453</td>
<td>5,266</td>
<td>472</td>
<td>5,198</td>
<td>49,086</td>
<td>60,667</td>
</tr>
<tr>
<td>UIUC QC</td>
<td>3,711</td>
<td>95</td>
<td>385</td>
<td>15,205</td>
<td>15,251</td>
<td>201</td>
</tr>
<tr>
<td>Tatoeba</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 4:** 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.
Inference Elicitation Procedure

In flux – Given a sentence, e.g. “If I were rich I would own a boat”, and a set of possible structure inference templates the annotator would:

1. **Select the inference template**
   
   (if \(<x>\) were \(<p>\) \(<x>\) would \(<q>\)) → (\(<x>\) is not \(<\text{pred}>\))

2. **Write down the result of the inference**

   “I am not rich”

Provide an option to write an inference that doesn’t correspond to one of the inference templates in case we miss a possibility.

The enumerate possible structure templates by sampling pattern.
I proposed a research plan for developing a semantic parser for ULFs with the following present state.

Completed:
- Pilot annotations of ULFs and annotation method development
- Preliminary ULF inference demonstration

On-going:
- Collection of the first annotation release
- Careful demonstration of ULF inference capabilities

Future:
- Training a parser on the ULF corpus
- Applying the ULF parser to more wide-scale demonstration of inference and usefulness.
Thank You!

Thank You!


Towards Simpler Annotations

New annotation procedure uses multiple stages so that each stage is a straight-forward task. Inspired by PMB.

New multi-stage approach

“Mary loves to solve puzzles”

1. Group syntactic constituents
(Mary (loves (to (solve puzzles))))

2. Run POS tagger over sentence
(nnp Mary) (vbz loves) (to to) (vb solve) (nns puzzles)

3. Correct POS tags and convert to dot-extensions
(Mary.nnp (loves.vbz (to.to (solve.vb puzzles.nns))))

4. Convert POS extensions to logical types, separate out morpho-syntactic operators
(|Mary| ((pres love.v) (to (solve.v (plur puzzle.n)))))

5. Add any implicit operators
(|Mary| ((pres love.v) (to (solve.v (k (plur puzzle.n)))))
Axiomatization Procedure

- WordNet Entry
  - Examples
  - Frames
  - Tagged Gloss

1) Argument Structure Inference

2) Semantic Parsing of Gloss

Refined Frames

Semantic Parse

3) Axiom Construction

Axiom
Motivation - Example Axiomatization

WordNet entry

*slam2.v*

Tagged gloss:

(VB strike1) (RB violently1)

Frames:

[Somebody slam2.v Something]
[Somebody slam2.v Somebody]

Examples: ("slam the ball")
Motivation - Example Axiomatization

WordNet entry

*slam2.v*

Tagged gloss:

(VB strike1) (RB violently1)

Frames:

[Somebody slam2.v Something]

[Somebody slam2.v Somebody]

Examples: (“slam the ball”)

Refined Frames:

[Somebody slam2.v Something]

1) Argument Structure Inference

1. Extend frames with example and gloss analysis.
2. Remove/merge redundant frames
2) Semantic Parsing of Gloss
1. Preprocess gloss into a sentence.
2. Parse sentence with a rule-based transducer.
3. Word sense disambiguation with POS tags.

Parse:
(Me.pro (violently1.adv (strike1.v It.pro))))
Motivation - Example Axiomatization

Refined Frames:
[Somebody slam2.v Something]

Parse:
(Me.pro (violently1.adv
 (strike1.v It.pro)))

Axiom Construction
1. Correlate frame and parse arguments.
2. Constrain argument types from frames.
3. Assert entailment from frame to gloss with type constraints.

Axiom:
\[
(\forall x_1 (\forall y_1 (\forall e \left[ x_1 \text{ slam2.v } y_1 \right] e)\left[ x_1 \text{ violently1.adv } y_1 \right] e)\left[ x_1 \text{ person1.n } y_1 \text{ thing12.n }\right])
\]
Motivation - Example Axiomatization

Refined Frames:
[Somebody slam2.v Something]

Parse:
(Me.pro (violently1.adv (strike1.v It.pro)))

3) Axiom Construction
1. Correlate frame and parse arguments.
2. Constrain argument types from frames.
3. Assert entailment from frame to gloss with type constraints.

Axiom:
(\forall x1 (\forall y1 (\forall e [[x1 slam2.v y1] ** e] 
[[[x1 (violently1.adv (strike1.v y1))] ** e] 
and [x1 person1.n] [y1 thing12.n]])))
Motivation - Evaluation

1. Agreement with manually-constructed gold standard axioms.
   - 50 synsets
   - 2,764 triples

2. Verb inference generation.
   - 812 verb pairs manually annotated with entailment (Weisman et al., 2012).
   - Simplified axioms.
   - Max 3-step forward inference.
   - Comparison with previous systems.

---

**Gold standard evaluation.**

<table>
<thead>
<tr>
<th>Measure</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL-smatch</td>
<td>0.85</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>Full Axiom</td>
<td>0.29</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Verb entailment evaluation.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Approach</td>
<td>0.43</td>
<td>0.53</td>
<td>0.48</td>
</tr>
<tr>
<td>TRIPS</td>
<td><strong>0.50</strong></td>
<td>0.45</td>
<td>0.47</td>
</tr>
<tr>
<td>Supervised</td>
<td>0.40</td>
<td><strong>0.71</strong></td>
<td><strong>0.51</strong></td>
</tr>
<tr>
<td>VerbOcean</td>
<td>0.33</td>
<td>0.15</td>
<td>0.20</td>
</tr>
<tr>
<td>Random</td>
<td>0.28</td>
<td>0.29</td>
<td>0.28</td>
</tr>
</tbody>
</table>
The greatest source of failure in the system was errors in the sentence-level EL interpretation.
The greatest source of failure in the system was errors in the sentence-level EL interpretation.

1 in 3 EL interpretations of glosses contained errors!

- Pretty good considering the problem, but not good enough to rely on in down-stream tasks.
1. **Segmentation**
   - impossible

2. **Syntactic Analysis**
   - CCG derivations with EasyCCG

3. **Semantic Tagging**
   - POS, NER, semantic, and discourse tags.

4. **Symbolization**
   - $2 \text{pm}$

5. **Semantic Interpretation**
   - Using the Boxer system

---

**Annotation Website**
- Layer-wise annotation view
- Edit template
- Dynamic re-analysis after rule edits
- Shared annotation view for reviews and corrections
- Edit tracker, revision history, and reversion
- An integrated bug-tracker for annotator organization and communication
- Automatic corpus statistics generation
Annotation Layers

1. Segmentation
   
   $impossible \rightarrow im\ possible$
Annotation Layers

1. Segmentation
   *impossible* $\rightarrow$ *im possible*

2. Syntactic Analysis
   CCG derivations with EasyCCG
Annotation Layers

1. Segmentation
   *impossible* → *im possible*

2. Syntactic Analysis
   CCG derivations with EasyCCG

3. Semantic Tagging
   POS, NER, semantic, and discourse tags.
Annotation Layers

1. Segmentation
   \textit{impossible} \rightarrow \textit{im possible}

2. Syntactic Analysis
   CCG derivations with EasyCCG

3. Semantic Tagging
   POS, NER, semantic, and discourse tags.

4. Symbolization
   \textit{2 pm} \rightarrow \textit{14:00}
Annotation Layers

1. Segmentation
   \textit{impossible} \rightarrow \textit{im possible}
2. Syntactic Analysis
   CCG derivations with EasyCCG
3. Semantic Tagging
   POS, NER, semantic, and discourse tags.
4. Symbolization
   \textit{2 pm} \rightarrow \textit{14:00}
5. Semantic Interpretation
   Using the Boxer system

 Annotation Website

- A layer-wise annotation view
- A edit template
- Dynamic re-analysis after rule edits
- Shared annotation view for reviews and corrections
- Edit tracker, revision history, and reversion
- An integrated bug-tracker for annotator organization and communication
- Automatic corpus statistics generation
Redwoods Summary

Pros
- Linguistically justified analysis.
- Good coverage of linguistic phenomena.
- Underspecification designed for applicability in context of more sentences.

Cons
- No general inference mechanism – existing ones are subsets of FOL or ad hoc.
- Uncertain formal interpretation of semantics.
- Hand-engineered grammars do not scale well to addition of linguistic phenomena.
- Approach requires a large amount of engineering – ERG grammar, HPSG parser, discriminant generator, storer, and applier.
AMR created without a formal analysis. Johan Bos published a model-theoretic analysis of AMR with the following results (Bos, 2016).

- Standard annotation of AMRs captures FOL without quantification.
- Polarity operators can be used to allow one $\forall$-quantification.
- AMR syntax may be extended to allow more $\forall$-quantifications.
AMR Semantics

AMR created without a formal analysis. Johan Bos published a model-theoretic analysis of AMR with the following results (Bos, 2016).

- Standard annotation of AMRs captures FOL without quantification.
- Polarity operators can be used to allow one $\forall$-quantification.
- AMR syntax may be extended to allow more $\forall$-quantifications.

Bender et al. (2015) show over-canonicalization.

**AMR-equivalent sentences**

- No one ate.
- Every person failed to eat.
Hermjakob (2013) built a special editor for AMR representations with the following core features:

- Unix-style text commands.
- Templates for beginner annotators.
- Point-and-click editing and automatic generation of certain cases for speedier annotations.
- Links to AMR roles, NER types, and suggestions.

Sentences can be annotated in about 10 minutes.

**Figure 12:** Screenshot of the AMR Editor editing the sentence “The girl wants to believe herself.”
AMR Annotations

The AMR project has annotated 47,274 sentences (21,065 publicly available).\(^1\) Numbers computed from AMR download website: http://amr.isi.edu/download.html

\(^2\)The rest of the sentences are only available to Deep Exploration and Filtering of Test (DEFT) DARPA program participants.
The AMR project has annotated 47,274 sentences (21,065 publicly available).\(^1\)

- *The Little Prince* corpus: 1,562 sentences.
- Bio AMR corpus: 6,452 sentences.
  - 3 full cancer-related PubMed articles
  - the result sections of 46 PubMed papers, and
  - 1,000 sentences from each of the BEL BioCreative training corpus and the Chicago Corpus.

**NOTE:** 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.

---

\(^1\)Numbers computed from AMR download website: [http://amr.isi.edu/download.html](http://amr.isi.edu/download.html)

\(^2\)The rest of the sentences are only available to Deep Exploration and Filtering of Test (DEFT) DARPA program participants.