University of Rochester

Thesis Proposal Presentation

Corpus Annotation and Inference with Episodic Logic Type Structure

Gene Kim

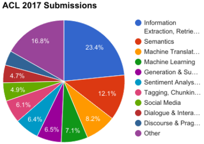
May 2, 2018

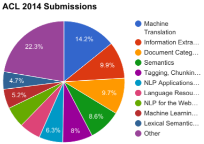
Introduction

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"

$$\Downarrow$$

```
(((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

```
(|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`

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)))
```

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)))))`

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)
```

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)
```

Coreference

```
(|E|.sk at-about.p |Now17|)
((|Manolin| (want.v (to go.v))) ** |E|.sk)
```

Inference using ULFs

Phrase structure + Coherent types

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.
  $\rightarrow$ *Len has been enjoying the sunny weather.*

- **Implicative, attitudinal, and communicative verbs**
  He managed to quit smoking. $\rightarrow$ *He quit smoking.*

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.*
- **Implicative, attitudinal, and communicative verbs**
  He managed to quit smoking. $\rightarrow$ *He quit smoking.*
- **Counterfactuals**
  Gene wishes people liked to go out to eat ice cream in the winter.
  $\rightarrow$ *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.
  $\rightarrow$ *Len has been enjoying the sunny weather.*

- **Implicative, attitudinal, and communicative verbs**
  He managed to quit smoking. $\rightarrow$ *He quit smoking.*

- **Counterfactuals**
  Gene wishes people liked to go out to eat ice cream in the winter.
  $\rightarrow$ *People don't like to go out to eat ice cream in the winter.*

- **Questions and requests**
  When are you getting married?
  $\rightarrow$ *You are getting married in the foreseeable future*

Summary of ULF Advantages

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

Outline

TRIPS

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

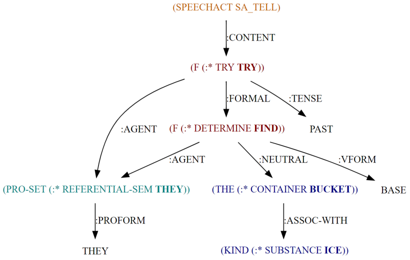


Figure 1: Parse for “They tried to find the ice bucket” using the vanilla dialogue model of TRIPS.

TRIPS LF

TRIPS Logical Form (Allen et al., 2008) descriptively covers of lot of language phenomena (e.g. generalized quantifiers, lambda abstractions, dialogue semantics, thematic roles).

TRIPS LF

TRIPS Logical Form (Allen et al., 2008) descriptively covers of 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 LF

TRIPS Logical Form (Allen et al., 2008) descriptively covers of 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)

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
```

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

Parallel Meaning Bank

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

Parallel Meaning Bank

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.

PMB Explorer

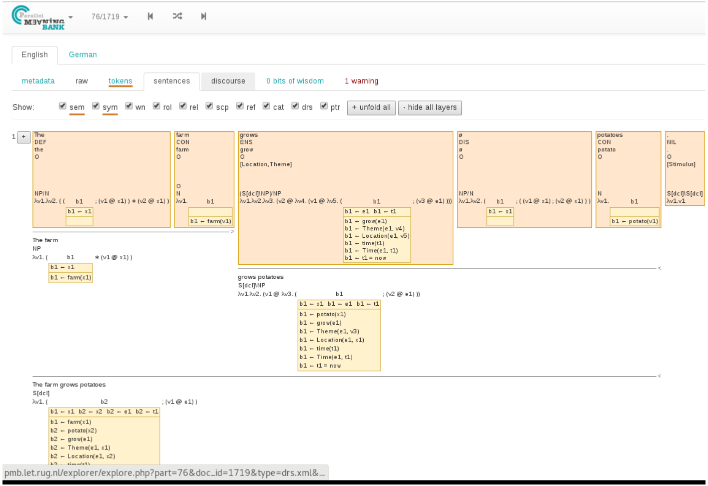


Figure 2: Screenshot of the PMB Explorer with analysis of the sentence "The farm grows potatoes."

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.

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

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.

```
_4:{
  _4:int_rel[SOA e2:_want2_rel]
  e2:_want2_rel[ARG1 x4:pron_rel, ARG4 _2:hypo_rel]
  _1:def_rel[BV x4:pron_rel]
  _2:hypo_rel[SOA e18:_meet_v_rel]
  e18:_meet_v_rel[ARG1 x4:pron_rel]
  e19:_on_temp_rel[ARG e18:_meet_v_rel, ARG3 x21:dofw_rel]
  x21:dofw_rel[NAMED :tue]
  _3:def_np_rel[BV x21:dofw_rel]
}
```

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).

Redwoods Annotations

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.

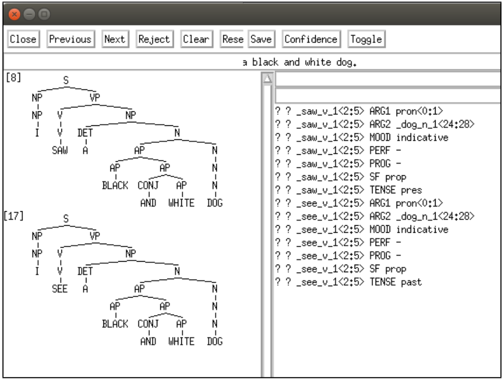


Figure 4: Screenshot of Redwoods treebanking environment for the sentence "I saw a black and white dog."

ERG Development Results

The ERG performance is a result of years of improvement.

Processing Stage	Stage Coverage	Running Total Coverage
Lexical Coverage	32%	32%
Able to Generate Parse	57%	18%
Contains Correct Parse	83%	15%

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

ERG Development Results

The ERG performance is a result of years of improvement.

Processing Stage	Stage Coverage	Running Total Coverage
Lexical Coverage	32%	32%
Able to Generate Parse	57%	18%
Contains Correct Parse	83%	15%

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

Years of grammar improvement was critical for annotation success!

Abstract Meaning Representation

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

$\exists w, g, b:$
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

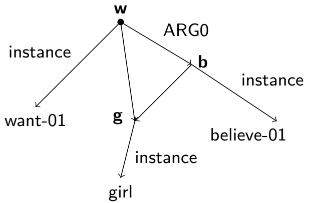


Figure 5: AMR representations for "The girl wanted to believe herself".

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.

Outline

Motivation - Lexical Axiom Extraction from WordNet

slam2.v
Gloss: "strike violently"
Frames: [Somebody slam2.v Something]
Examples: "slam the ball"

$$\Downarrow$$

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**

Research Plan Overview

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.

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

Table 2: Average timing of experimental ULF annotations.

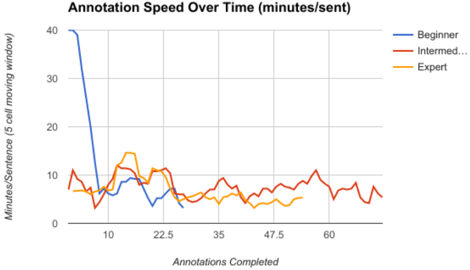


Figure 6: Timing results from ULF experimental annotations.

First Pilot Annotations - Limitations

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

1. Simplify annotation procedure with multi-layered 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))
```

⇓

```
((burning.a hot.a) (melting.n pot.n))
```

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)))`

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))`

Macros

Similar to C-macros, but accompanied by a few specially interpreted items.

Post-nominal modifiers

```
(n+preds N Pred1 Pred2 ... PredN) ≡
          (λ x ((x N) and (x Pred1) (x Pred2) ... (x PredN)))
(np+preds NP Pred1 Pred2 ... PredN) ≡
          (the.d (λ x ((x = NP) and (x Pred1) (x Pred2) ... (x PredN))))
```

Macros

Similar to C-macros, but accompanied by a few specially interpreted items.

Post-nominal modifiers

```
(n+preds N Pred1 Pred2 ... PredN) ≡
         (λ x ((x N) and (x Pred1) (x Pred2) ... (x PredN)))
(np+preds NP Pred1 Pred2 ... PredN) ≡
         (the.d (λ x ((x = NP) and (x Pred1) (x Pred2) ... (x PredN))))
```

The table by the fireplace with three legs

```
(the.d (n+preds table.n (by.p (the.d fireplace.n))
                        (with.p ((nquan three.a) (plur leg.n)))))
```

Macros

<u>Relative Clauses</u>

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

$\texttt{S}_{emb}\texttt{[that.rel]} \equiv \texttt{(}\lambda\ \texttt{*r S}_{emb}\texttt{[that.rel}\leftarrow\texttt{*r])}$

car that you bought

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

Macros

Relative Clauses

(sub C S[*h]) $\equiv$ S[*h$\leftarrow$C]

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

car that you bought

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

$\Downarrow$ n+preds

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

Macros

Relative Clauses

(sub C S[*h]) $\equiv$ S[*h$\leftarrow$C]

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

car that you bought

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

$\Downarrow$ n+preds

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

$\Downarrow$ sub

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

Macros

<u>Relative Clauses</u>

`(sub C S[*h])` $\equiv$ `S[*h` $\leftarrow$ `C]`

S_{emb}`[that.rel]` $\equiv$ `(`λ `*r` S_{emb}`[that.rel` $\leftarrow$ `*r])`

car that you bought

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

$\Downarrow$ `n+preds`

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

$\Downarrow$ `sub`

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

$\Downarrow$ `that.rel`

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

Macros

<u>Relative Clauses</u>

(sub C S[*h]) $\equiv$ S[*h$\leftarrow$C]

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

car that you bought

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

$\Downarrow$ n+preds

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

$\Downarrow$ sub

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

$\Downarrow$ that.rel

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

$\Downarrow$ λ-conversion

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

Macros

Prenominal Possessive

`((NP 's) N)` $\equiv$ `(the.d ((poss-by NP) N))`

Example: `((|John| 's) dog.n)` $\equiv$ `(the.d ((poss-by |John|) dog.n))`

Possessive Determiners

`(my.d N)` $\leftrightarrow$ `(the.d ((poss-by me.pro) N))`,

where `my.d` and `me.pro` can be replaced by any corresponding pair of possessive determiner and personal pronoun.

Under development

Comparatives, Superlatives, Questions, Gaps, Discourse Markers

First Annotation Release

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

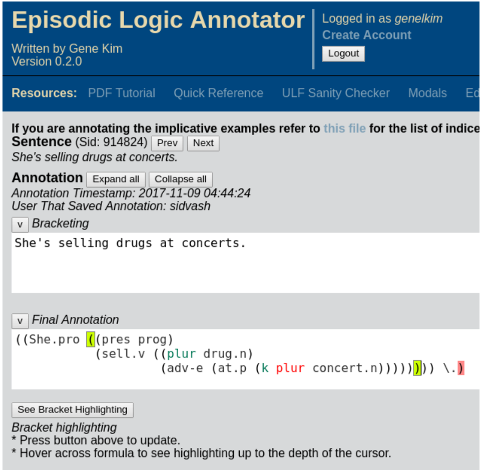
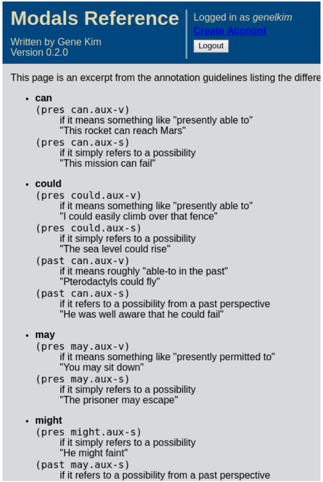


Figure 7: Current ULF annotator state with example annotation process.

Current Annotator State



Modals Reference

Written by Gene Kim
Version 0.2.0

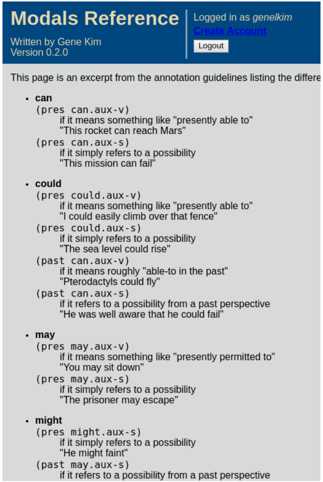
Logged in as *genelkim*
Create Account
Logout

This page is an excerpt from the annotation guidelines listing the differe

- **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 faint"
  `(past may.aux-s)`
  if it refers to a possibility from a past perspective

Figure 8: Screenshot of modals reference.

Current Annotator State



Modals Reference

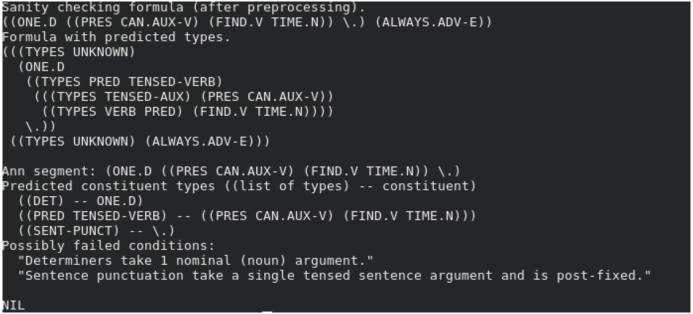
Written by Gene Kim
Version 0.2.0

Logged in as *genelkim*
Create Account
Logout

This page is an excerpt from the annotation guidelines listing the differe

- **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 faint"
  `(past may.aux-s)`
  if it refers to a possibility from a past perspective

Figure 8: Screenshot of modals reference.



```
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-PUNCT) -- \.)
Possibly failed conditions:
  "Determiners take 1 nominal (noun) argument."
  "Sentence punctuation take a single tensed sentence argument and is post-fixed."

NIL
```

Figure 9: Screenshot of sanity checker output.

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**

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.

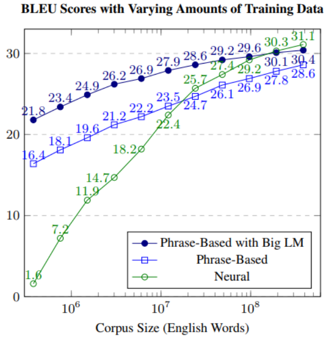


Figure 10: Performance of neural vs phrase-based MT systems as a function of data size (Koehn and Knowles, 2017).

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).

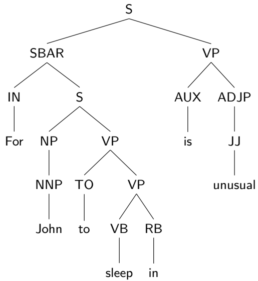
STSG rules

$$X \Rightarrow a, b$$

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

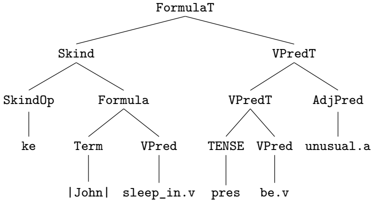
STSG Example

(a) **Constituency tree**



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

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



(c) **Possible Rules**

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

Figure 11: Rules for the example sentence *For John to sleep in is unusual*.

String-to-tree Method

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. `Formula` $\rightarrow$ `Term,VPred` and `Formula'` $\rightarrow$ `VPred,Term` for reordered variants.

- **Much more computationally efficient**

- **Can use known type-restrictions for model initialization**

Fine-tuning Models to Downstream Tasks

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 Models to Downstream Tasks

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

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

X: the set of inputs
θ: model parameters
y: the output
α,β: hyperparameters for the convergence rate

Evaluating the Parser

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

Pilot Inference Demo

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.

Pilot Inference Results

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: 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.

ULF Inference Demonstration

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"
  $\rightarrow$ *Obama gets no respect*
  "When is your wedding?"
  $\rightarrow$ *You are getting married in the near future*

Sampling Collection Procedure

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

Sampling Statistics

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 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>)` $\rightarrow$ `(<x> is not <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.

Conclusion

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!

References I

Allen, James F., 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. Venice, Italy: Association for Computational Linguistics, pp. 343–354. URL: http://dl.acm.org/citation.cfm?id=1626481.1626508.

Allen, James F. et al. (2018). "Effective Broad-Coverage Deep Parsing". In: *AAAI Conference on Artificial Intelligence*.

Bender, Emily M. et al. (2015). "Layers of Interpretation: On Grammar and Compositionality". In: *Proceedings of the 11th International Conference on Computational Semantics*. London, UK: Association for Computational Linguistics, pp. 239–249. URL: http://www.aclweb.org/anthology/W15-0128.

Bos, Johan (2016). "Expressive Power of Abstract Meaning Representations". In: *Computational Linguistics* 42.3, pp. 527–535. ISSN: 0891-2017. DOI: 10.1162/COLI_a_00257. URL: https://doi.org/10.1162/COLI_a_00257.

References II

Chung, Tagyoung et al. (2014). "Sampling Tree Fragments from Forests". In: *Computational Linguistics* 40, pp. 203–229.

Eisner, Jason (2003). "Learning non-isomorphic tree mappings for machine translation". In: *Proceedings of the 41st Meeting of the Association for Computational Linguistics, companion volume*. Sapporo, Japan, pp. 205–208.

Gildea, Daniel (2003). "Loosely Tree-Based Alignment for Machine Translation". In: *Proceedings of ACL-03*. Sapporo, Japan, pp. 80–87. URL: http://www.cs.rochester.edu/~gildea/gildea-acl03.pdf.

Hermjakob, Ulf (2013). *AMR Editor: A Tool to Build Abstract Meaning Representations*. URL: http://www.isi.edu/~ulf/amr/AMR-editor.html.

Koehn, Philipp and Rebecca Knowles (2017). "Six Challenges for Neural Machine Translation". In: *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, pp. 28–39. URL: http://aclweb.org/anthology/W17-3204.

References III

Oepen, Stephan et al. (2002). "LinGO Redwoods: A Rich and Dynamic Treebank for HPSG". In: *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*. Sozopol, Bulgaria.

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

Reisinger, Drew et al. (2015). "Semantic Proto-Roles". In: *Transactions of the Association for Computational Linguistics* 3, pp. 475–488. ISSN: 2307-387X. URL: https://transacl.org/ojs/index.php/tacl/article/view/674.

Rudinger, Rachel, Aaron Steven White, and Benjamin Van Durme (2018). "Neural Models of Factuality". In: *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

References IV

Weisman, Hila et al. (2012). "Learning Verb Inference Rules from Linguistically-motivated Evidence". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 194–204. URL: http://dl.acm.org/citation.cfm?id=2390948.2390972.

White, Aaron Steven et al. (2016). "Universal Decompositional Semantics on Universal Dependencies". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1713–1723. URL: https://aclweb.org/anthology/D16-1177.

Williams, Ronald J. (1992). "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". In: *Machine Learning* 8.3-4, pp. 229–256.

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))))
```

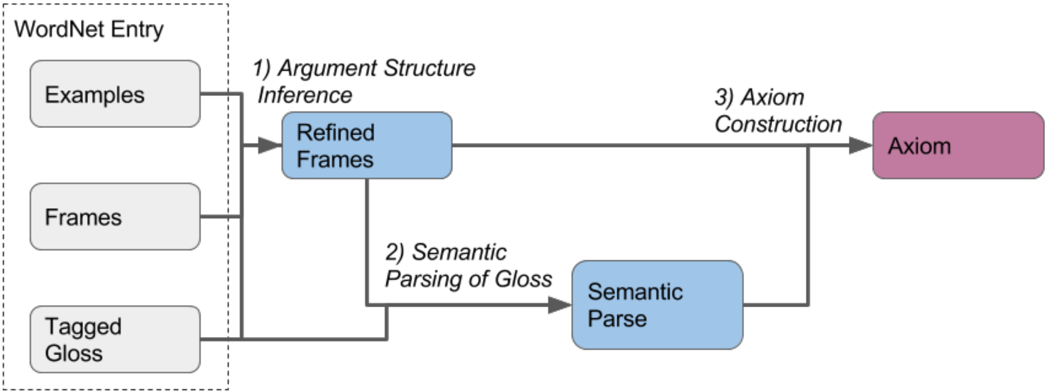
⇓ 4. Convert POS extensions to logical types, separate out morpho-syntactic operators

```
(|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



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")

1) Argument Structure Inference

1. Extend frames with example and gloss analysis.
2. Remove/merge redundant frames

Refined Frames:

`[Somebody slam2.v Something]`

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")

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.

Refined Frames:

`[Somebody slam2.v Something]`

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)))
```

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:

```
(∀x1 (∀y1 (∀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

Gold standard evaluation.

Measure	Precision	Recall	F1
EL-smatch	0.85	0.82	0.83
Full Axiom	0.29	-	-

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.

Verb entailment evaluation.

Method	Precision	Recall	F1
Our Approach	0.43	0.53	0.48
TRIPS	**0.50**	0.45	0.47
Supervised	0.40	**0.71**	**0.51**
VerbOcean	0.33	0.15	0.20
Random	0.28	0.29	0.28

Motivation - Parsing Errors

The greatest source of failure in the system was errors in the sentence-level EL interpretation.

Motivation - Parsing Errors

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.

PMB Annotations

PMB Annotations

Annotation Layers

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

PMB Annotations

Annotation Layers

1. Segmentation
   impossible → im possible
2. Syntactic Analysis
   CCG derivations with EasyCCG

PMB Annotations

Annotation Layers

1. Segmentation
   impossible → im possible
2. Syntactic Analysis
   CCG derivations with EasyCCG
3. Semantic Tagging
   POS, NER, semantic, and discourse tags.

PMB Annotations

Annotation Layers

1. Segmentation
   impossible $\rightarrow$ *im possible*
2. Syntactic Analysis
   CCG derivations with EasyCCG
3. Semantic Tagging
   POS, NER, semantic, and discourse tags.
4. Symbolization
   2 pm $\rightarrow$ `14:00`

PMB Annotations

Annotation Layers

1. Segmentation
   impossible → im possible
2. Syntactic Analysis
   CCG derivations with EasyCCG
3. Semantic Tagging
   POS, NER, semantic, and discourse tags.
4. Symbolization
   2 pm → `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 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.

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.

AMR Editor

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.

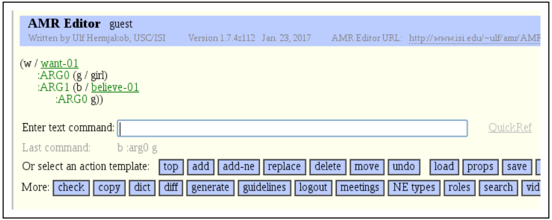


Figure 12: Screenshot of the AMR Editor editing the sentence “The girl wants to believe herself.”

AMR Annotations

he AMR project has annotated 47,274 sentences (21,065 publicly available)[1,2].

[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.

AMR Annotations

he AMR project has annotated 47,274 sentences (21,065 publicly available)[1,2].

- *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
  - 1000 sentences from each of the BEL BioCreative training corpus and the Chicago Corpus.
- LDC corpus : 39,260 sentences (13,051 general release). Mostly of samplings from machine translation corpora with 200 sentences from weblogs and the WSJ 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`

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