Generating Discourse Inferences from Unscoped Episodic Logical Formulas

Gene Louis Kim, Benjamin Kane, Viet Duong, Muskaan Mendiratta, Graeme McGuire, Sophie Sackstein, Georgiy Platonov, and Lenhart Schubert

Presented by: Gene Louis Kim
August 2019
Unscoped episodic logical form (ULF) is an expressive initial representation of Episodic Logic, but inference with it has not been demonstrated with it.
Unscoped episodic logical form (ULF) is an expressive initial representation of Episodic Logic, but inference with it has not been demonstrated with it.

**Unscoped {Episodic} Logical Form (ULF)**

- **An underspecified Episodic Logic (EL)**
  - Extended FOL, closely matches expressivity of natural languages
    - modification, reification, generalized quantifiers, and more
Unscoped episodic logical form (ULF) is an expressive initial representation of Episodic Logic, but inference with it has not been demonstrated with it.

### Unscoped \{Episodic\} Logical Form (ULF)

- **An underspecified Episodic Logic (EL)**
  - Extended FOL, closely matches expressivity of natural languages
    - modification, reification, generalized quantifiers, and more
- **Starting point for EL parsing**

---

**Episodic Logic Pipeline**

Unscoped episodic logical form (ULF) is an expressive initial representation of Episodic Logic, but inference with it has not been demonstrated with it.
Unscoped episodic logical form (ULF) is an expressive initial representation of Episodic Logic, but inference with it has not been demonstrated with it.

Unscoped \{Episodic\} Logical Form (ULF)

- An underspecified Episodic Logic (EL)
  - Extended FOL, closely matches expressivity of natural languages
    - modification, reification, generalized quantifiers, and more
- Starting point for EL parsing
- Enables situated inferences
Unscoped episodic logical form (ULF) is an expressive initial representation of Episodic Logic, but inference with it has not been demonstrated with it.

**Unscoped {Episodic} Logical Form (ULF)**

- An underspecified Episodic Logic (EL)
  - Extended FOL, closely matches expressivity of natural languages
    - modification, reification, generalized quantifiers, and more
- Starting point for EL parsing
- Enables situated inferences
We select the following inference types for evaluation:

<table>
<thead>
<tr>
<th>Type</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>questions</td>
<td>“How soon can you get that done?”</td>
<td>“You can get that done”</td>
</tr>
<tr>
<td>requests</td>
<td>“Could you put your seat back up?”</td>
<td>“I want and expect you to put your seat back up”</td>
</tr>
<tr>
<td>counterfactuals</td>
<td>“I wish I had turned off the stove”</td>
<td>“I didn’t turn off the stove”</td>
</tr>
<tr>
<td>clause-taking verbs</td>
<td>“John suspects that I’m lying”</td>
<td>“John thinks that I am probably lying”</td>
</tr>
</tbody>
</table>
Introduction

We select the following inference types for evaluation:

- **questions**: “How soon can you get that done?”
  - “You can get that done”

- **requests**: “Could you put your seat back up?”
  - “I want and expect you to put your seat back up”
  - Via ULF
  - “I didn’t turn off the stove”

- **counterfactuals**: “I wish I had turned off the stove”

- **clause-taking verbs**: “John suspects that I’m lying”
  - “John thinks that I am probably lying”

**Properties of Inferences**
1. important for setting a natural discourse context
We select the following inference types for evaluation:

- **questions**
  - "How soon can you get that done?"
  - "You can get that done"

- **requests**
  - "Could you put your seat back up?"
  - "I want and expect you to put your seat back up"
  - Via ULF

- **counterfactuals**
  - "I wish I had turned off the stove"
  - "I didn't turn off the stove"

- **clause-taking verbs**
  - "John suspects that I'm lying"
  - "John thinks that I am probably lying"

**Properties of Inferences**

1. important for setting a natural discourse context
2. structurally-oriented - we can avoid turning evaluation into a classification problem
ULF? (syntax)

A minimal step across from syntax to semantics in Episodic Logic
ULF? (syntax)

A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULF

(⟨Alice| ((pres think.v)
  (that ⟨John| (nearly.adv-a (past fall.v)))))

Syntax (simplified)

(S (NP Alice.nnp) (VP thinks.vbz
  (SBAR that.rb (S (NP John.nnp) (ADVP nearly.rb) (VP fell.vbd)))))
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULF

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

Syntax (simplified)

\[
(\langle S \text{ (NP Alice.nnp) (VP thinks.vbz) }
\quad (\langle \text{SBAR that.rb} \text{ (S (NP John.nnp) (ADVP nearly.rb) (VP fell.vbd)}) \rangle) \rangle)
\]

Proper Nouns  Verbs  Adverbs
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULFs

\[ (\text{Alice} \rightarrow ((\text{pres} \ \text{think.v}) \ \text{that} \ (\text{John} \ \text{nearl}y.a (\text{past} \ \text{fall.v})))) \]
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULFs

(|Alice| (((pres think.v) (that (|John| (nearly.adv-a (past fall.v)))))))

Entity(\(\mathcal{D}\)): |Alice|, |John|
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULFs

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

\begin{itemize}
  \item **Basic Ontological Types**
  \begin{itemize}
    \item \(D\) \hspace{1em} Domain
    \item \(S\) \hspace{1em} Situations
    \item \(2\) \hspace{1em} Truth-value
  \end{itemize}
  \begin{itemize}
    \item Monadic Predicate \(\mathcal{N} : D \rightarrow (S \rightarrow 2)\)
  \end{itemize}
\end{itemize}

\begin{itemize}
  \item Entity(\(D\)) : |Alice|, |John|
  \item \text{n-ary predicate}(\(D^n \rightarrow (S \rightarrow 2)\)) : think.v, fall.v
\end{itemize}
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULFs

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

Entity(\(\mathcal{D}\)): |Alice|, |John|

n-ary predicate(\(\mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow 2)\)): think.v, fall.v

Predicate modifier(\(\mathcal{N} \rightarrow \mathcal{N}\)): nearly.adv-a
ULF? (semantics)

A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULFs
(|Alice| (((pres think.v)
  (that (|John| (nearly.adv-a (past fall.v)))))))

Basic Ontological Types
- \( \mathcal{D} \): Domain
- \( \mathcal{S} \): Situations
- \( \mathbb{2} \): Truth-value

Monadic Predicate \( \mathcal{N} : \mathcal{D} \rightarrow (\mathcal{S} \rightarrow \mathbb{2}) \)

Entity(\( \mathcal{D} \)): |Alice|, |John|

\( n\text{-ary predicate}(\mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow \mathbb{2})) \): think.v, fall.v

Predicate modifier(\( \mathcal{N} \rightarrow \mathcal{N} \)): nearly.adv-a

Sentence reifier((\( \mathcal{S} \rightarrow \mathbb{2} \) \rightarrow \( \mathcal{D} \))): that
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”

ULFs

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

**Basic Ontological Types**

- Domain \( \mathcal{D} \)
- Situations \( \mathcal{S} \)
- Truth-value \( 2 \)

**Monadc Predicate**
\[ \mathcal{N} : \mathcal{D} \rightarrow (\mathcal{S} \rightarrow 2) \]

**Entity** (\( \mathcal{D} \)): \( |Alice|, \; |John| \)

**\( \mathcal{n\text{-ary predicate}} \)** (\( \mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow 2) \)): \text{think.v}, \text{fall.v}

**Predicate modifier** (\( \mathcal{N} \rightarrow \mathcal{N} \)): \text{nearly.adv-a}

**Sentence reifier** (\( (\mathcal{S} \rightarrow 2) \rightarrow \mathcal{D} \)): \text{that}

Also... \text{determiner}, \text{sentence modifier}, \text{connective}, \text{lambda abstract}, \text{predicate reifier}
1. Abstract away syntactic idiosyncrasies with interpretable predicates and functions
1. Abstract away syntactic idiosyncrasies with interpretable predicates and functions

**Predicates**

verb-phrase? - defined over the ULF semantic type system

wh-word? - defined as a list
1. Abstract away syntactic idiosyncrasies with interpretable predicates and functions

**Predicates**

- **verb-phrase?** - defined over the ULF semantic type system
- **wh-word?** - defined as a list
1. Abstract away syntactic idiosyncrasies with interpretable predicates and functions

**Predicates**

- **verb-phrase?** - defined over the ULF semantic type system
  
  - lexical-verb?
  
  - (verb-phrase? term?)

- **wh-word?** - defined as a list
1. Abstract away syntactic idiosyncrasies with interpretable predicates and functions

**Predicates**

verb-phrase? - defined over the ULF semantic type system

wh-word? - defined as a list

verb-phrase? :
  | lexical-verb?
  | (verb-phrase? term?)
  | (verb-modifier? verb-phrase?)
  | ...

Building ULF Inference Rules
Abstract away syntactic idiosyncrasies with interpretable predicates and functions

**Predicates**

*verb-phrase?* - defined over the ULF semantic type system

*wh-word?* - defined as a list

**Functions**

*negate-verb-phrase!*

“left the house” → “did not leave the house”

“could leave the house” → “could not leave the house”
1. Abstract away syntactic idiosyncrasies with interpretable predicates and functions

**Predicates**

- **verb-phrase?** - defined over the ULF semantic type system
  
  ```
  verb-phrase? : 
  | lexical-verb? 
  | (verb-phrase? term?) 
  | (verb-modifier? verb-phrase?) 
  | ...
  ```

- **wh-word?** - defined as a list

**Functions**

- **negate-verb-phrase!**
  - “left the house” → “did not leave the house”
  - “could leave the house” → “could not leave the house”

- **uninvert-sentence!**
  - “did you leave already” → “you did leave already”
Building ULF Inference Rules

2. Construct simple if-then rules
2. Construct simple if-then rules

if formula satisfies contains-wh? and ends with a question mark

“what did you buy?”

((sub what.pro
   ((past do.aux-s)
    you.pro (buy.v *h))) ?)
2. Construct simple if-then rules

if formula satisfies contains-wh? and ends with a question mark

strip the question mark

“what did you buy?”

((sub what.pro
   ((past do.aux-s)
    you.pro (buy.v *h))) ?)

→

((sub what.pro
   ((past do.aux-s)
    you.pro (buy.v *h)))

“what did you buy”
2. Construct simple if-then rules

if formula satisfies contains-wh? and ends with a question mark

strip the question mark

apply preprocessing markers

“what did you buy?”

((sub what.pro
  ((past do.aux-s
      you.pro (buy.v *h))) ?)

(sub what.pro
  ((past do.aux-s
      you.pro (buy.v *h)))

((past do.aux-s
  you.pro (buy.v what.pro)))

“did you buy what”
2. Construct simple if-then rules

if formula satisfies contains-wh? and ends with a question mark

strip the question mark

apply preprocessing markers

apply uninvert-sentence!

“what did you buy?”

((sub what.pro
   ((past do.aux-s)
    you.pro (buy.v *h))) ?)

((past do.aux-s)
 you.pro (buy.v what.pro))

(you.pro ((past do.aux-s)
  (buy.v what.pro)))

“you did buy what”
2. Construct simple if-then rules

if formula satisfies contains-wh? and ends with a question mark

strip the question mark

apply preprocessing markers

apply uninvert-sentence!

apply wh2some!

“what did you buy?”

((sub what.pro
   ((past do.aux-s)
    you.pro (buy.v *h))) ?)

(you.pro ((past do.aux-s)
          (buy.v something.pro)))

“you did buy something”
2. Construct simple if-then rules

if formula satisfies contains-wh? and ends with a question mark

- strip the question mark
- apply preprocessing markers
- apply uninvert-sentence!
- apply wh2some!

infer-wh-question-presupposition

"what did you buy?"

```plaintext
((sub what.pro
  ((past do.aux-s)
    you.pro (buy.v *h))) ?)
```

```
(you.pro ((past do.aux-s)
  (buy.v what.pro)))
```

```
(you.pro ((past do.aux-s)
  (buy.v something.pro)))
```

"you did buy something"
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

“How soon can you get that done?”
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

2. Elicit human inferences in each category
   a. Reduced noise with structured guidance

Human Annotator

“How soon can you get that done?”

“You can get that done”
“I want and expect you to get that done”
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

2. Elicit human inferences in each category
   a. Reduced noise with structured guidance

Human Inference Elicitation
1. Select inference category
   a. request, question, counterfactual, clause-taking
2. Select inference structure
   a. e.g. (if <x> were <pred>, <x> would <q>)
      → (<x> is not <pred>)
3. Write fluent sentence corresponding to inference antecedent
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

2. Elicit human inferences in each category
   a. Reduced noise with structured guidance

Human Inference Elicitation
1. Select inference category
   a. request, question, counterfactual, clause-taking

2. Select inference structure
   a. e.g. (if \(x\) were \(\text{pred}\), \(x\) would \(q\))
   \[ \rightarrow (\langle x \rangle \text{ is not } \langle \text{pred} \rangle) \]

3. Write fluent sentence corresponding to inference antecedent

Dataset of 698 elicited inferences over 406 sentences.
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

2. Elicit human inferences in each category
   a. Reduced noise with structured guidance

3. Collect gold ULF annotations of each sentence

   "How soon can you get that done?"

   "You can get that done"
   "I want and expect you to get that done"
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead
2. Elicit human inferences in each category
   a. Reduced noise with structured guidance
3. Collect gold ULF annotations of each sentence

ULF Annotator with
- Syntax highlighter
- Sanity checker

“How soon can you get that done?”

“You can get that done”
“I want and expect you to get that done”
Inference Evaluation

1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

2. Elicit human inferences in each category
   a. Reduced noise with structured guidance

3. Collect gold ULF annotations of each sentence

4. **Use inference rules to make conclusions**

   "You can get that done"
   "I want and expect you to get that done"

\[
((\text{sub (how.mod-a soon.a)})
\quad ((\text{pres can.aux-v}) \ \text{you.pro}
\quad (\text{get.v that.pro done.a *h})))
\]

“Automatic” ➔ “Inferred ULFs”
Inference Evaluation

1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

2. Elicit human inferences in each category
   a. Reduced noise with structured guidance

3. Collect gold ULF annotations of each sentence

4. Use inference rules to make conclusions

5. **Automatically generate English from ULF inferences**

---

Inferred ULFs

```
((sub (how.mod-a soon.a) ((pres can.aux-v) you.pro (get.v that.pro done.a *h))) ?)
```

Inferred Sentences

- “How soon can you get that done?”
- “You can get that done”
- “I want and expect you to get that done”

Automatic
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead
2. Elicit human inferences in each category
   a. Reduced noise with structured guidance
3. Collect gold ULF annotations of each sentence
4. Use inference rules to make conclusions
5. **Automatically generate English from ULF inferences**

The ULF-to-English translation
1. Analyze the ULF type of each clause,
2. Incorporate morphological inflections based on the type analysis,
3. Filter out purely logical operators, and
4. Map logical symbols to surface form counterparts.

---

"How soon can you get that done?"

"You can get that done"
"I want and expect you to get that done"
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead
2. Elicit human inferences in each category
   a. Reduced noise with structured guidance
3. Collect gold ULF annotations of each sentence
4. Use inference rules to make conclusions
5. Automatically generate English from ULF inferences
6. Evaluate
   a. precision using human judgments
   b. recall using automatic matching
Human elicited inferences are incomplete, so precision is only measured with human evaluation.
Human Evaluation (Precision)

Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
- 3 or 4 human judgments per inference
Human Evaluation (Precision)

Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
- 3 or 4 human judgments per inference
- 3 Categories: correct, incorrect, context-dependent
Human Evaluation (Precision)

Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
- 3 or 4 human judgments per inference
- 3 Categories: correct, incorrect, context-dependent
- Correctness(*) = reasonable inference in conversation, allowing for a bit of awkwardness in phrasing

"You can get that done"
"I want and expect you to get that done"
Human Evaluation (Precision)

Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
- 3 or 4 human judgments per inference
- 3 Categories: correct, incorrect, context-dependent
- Correctness(*) = reasonable inference in conversation, allowing for a bit of awkwardness in phrasing
- Evaluated grammaticality
Human Evaluation (Precision)

Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
- 3 or 4 human judgments per inference
- 3 Categories: correct, incorrect, context-dependent
- Correctness(*) = reasonable inference in conversation, allowing for a bit of awkwardness in phrasing
- Evaluated grammaticality

<table>
<thead>
<tr>
<th></th>
<th>cf</th>
<th>cls</th>
<th>req</th>
<th>q-pre</th>
<th>q-act</th>
<th>oth</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct*</td>
<td>11/27</td>
<td>2/5</td>
<td>17/19</td>
<td>13/21</td>
<td>31/39</td>
<td>13/16</td>
<td>68.5%</td>
</tr>
<tr>
<td>Incorrect*</td>
<td>9/27</td>
<td>3/5</td>
<td>0/19</td>
<td>3/21</td>
<td>3/39</td>
<td>3/16</td>
<td>16.5%</td>
</tr>
<tr>
<td>Context*</td>
<td>7/27</td>
<td>0/5</td>
<td>2/19</td>
<td>5/21</td>
<td>5/39</td>
<td>0/16</td>
<td>15.0%</td>
</tr>
<tr>
<td>Grammar</td>
<td>20/27</td>
<td>1/5</td>
<td>19/19</td>
<td>12/21</td>
<td>33/39</td>
<td>14/16</td>
<td>78.0%</td>
</tr>
</tbody>
</table>

cf: counterfactual
cls: clause-taking
req: request
q-pre: question presuppositional inferences
q-act: question act inferences
oth: other
Human Evaluation (Precision)

Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
- 3 or 4 human judgments per inference
- 3 Categories: correct, incorrect, context-dependent
- Correctness(*) = reasonable inference in conversation, allowing for a bit of awkwardness in phrasing
- Evaluated grammaticality

<table>
<thead>
<tr>
<th></th>
<th>cf</th>
<th>cls</th>
<th>req</th>
<th>q-pre</th>
<th>q-act</th>
<th>other</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct*</td>
<td>11/27</td>
<td>2/5</td>
<td>17/19</td>
<td>13/21</td>
<td>31/39</td>
<td>13/16</td>
<td>68.5%</td>
</tr>
<tr>
<td>Incorrect*</td>
<td>9/27</td>
<td>3/5</td>
<td>0/19</td>
<td>3/21</td>
<td>3/39</td>
<td>3/10</td>
<td>16.5%</td>
</tr>
<tr>
<td>Context*</td>
<td>7/27</td>
<td>0/5</td>
<td>2/19</td>
<td>5/21</td>
<td>5/39</td>
<td>0/10</td>
<td>15.0%</td>
</tr>
<tr>
<td>Grammar</td>
<td>20/27</td>
<td>1/5</td>
<td>19/19</td>
<td>12/21</td>
<td>33/39</td>
<td>14/16</td>
<td>78.0%</td>
</tr>
</tbody>
</table>

cf: counterfactual  
cls: clause-taking  
req: request  
q-pre: question presuppositional inferences  
q-act: question act inferences  
oth: other
Human Evaluation (Precision)

Human elicited inferences are incomplete, so precision is only measured with human evaluation.

- Sample of 127 inferences
  - all counterfactual and clause-taking examples and random sampling of others
- 3 or 4 human judgments per inference
- 3 Categories: correct, incorrect, context-dependent
- Correctness(*) = reasonable inference in conversation, allowing for a bit of awkwardness in phrasing
- Evaluated grammaticality

<table>
<thead>
<tr>
<th></th>
<th>cf</th>
<th>cls</th>
<th>req</th>
<th>q-pre</th>
<th>q-act</th>
<th>oth</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct*</td>
<td>11/27</td>
<td>2/5</td>
<td>17/19</td>
<td>13/21</td>
<td>31/39</td>
<td>13/16</td>
<td>68.5%</td>
</tr>
<tr>
<td>Incorrect*</td>
<td>9/27</td>
<td>3/5</td>
<td>0/19</td>
<td>3/21</td>
<td>3/39</td>
<td>3/16</td>
<td>16.5%</td>
</tr>
<tr>
<td>Context*</td>
<td>7/27</td>
<td>0/5</td>
<td>2/19</td>
<td>5/21</td>
<td>5/39</td>
<td>0/16</td>
<td>15.0%</td>
</tr>
<tr>
<td>Grammar</td>
<td>20/27</td>
<td>1/5</td>
<td>19/19</td>
<td>12/21</td>
<td>33/39</td>
<td>14/16</td>
<td>78.0%</td>
</tr>
</tbody>
</table>

cf: counterfactual
cls: clause-taking
req: request
q-pre: question presuppositional inferences
q-act: question act inferences
oth: other
Human Annotator

"You can get that done"
"I want and expect you to get that done"

Automatic Eval

Automatic

Inferred Sentences

Inferred ULFs

Automatic Evaluation (Recall)
The automatic evaluation has extra steps to deal with paraphrases.
1. Generate ULF inferences using generalized ULF predicates and transformations:
   infer-wh-question-presupposition
1. Generate ULF inferences using generalized ULF predicates and transformations:
   `infer-wh-question-presupposition`

2. Rewrite inferences into possible other forms [In ULF]:
   “I want you to get that done” + “I expect you to get that done” → “I want and expect you to get that done”
1. Generate ULF inferences using generalized ULF predicates and transformations:
   infer-wh-question-presupposition

2. Rewrite inferences into possible other forms [In ULF]:
   “I want you to get that done” + “I expect you to get that done” → “I want and expect you to get that done”

3. Translate to English:
   (i.pro (((pres want.v) and.cc (pres expect.v)) you.pro (to (get.v that.pro done.a))))
   → “I want and expect you to get that done”
1. **Generate ULF inferences using generalized ULF predicates and transformations:**

   \[ \text{infer-wh-question-presupposition} \]

2. **Rewrite inferences into possible other forms [In ULF]:**

   “I want you to get that done” + “I expect you to get that done” \rightarrow “I want and expect you to get that done”

3. **Translate to English:**

   \[ (\text{i.pro } ((\text{pres want.v}) \text{ and.cc } (\text{pres expect.v}))) \text{ you.pro } (\text{to } (\text{get.v that.pro done.a}))) \]

   \rightarrow \text{“I want and expect you to get that done”}

4. **Select closest match between the gold inferences and the available rewrite sentences**

   a. Allow 3 character edit distance between gold inference and inferred sentence to allow minor English generation errors.
1. Generate ULF inferences using generalized ULF predicates and transformations: 
   infer-wh-question-presupposition

2. Rewrite inferences into possible other forms [In ULF]:
   “I want you to get that done” + “I expect you to get that done” → “I want and expect you to get that done”

3. Translate to English:
   \[(i\.pro (((\text{pres want.v}) \text{and.cc} (\text{pres expect.v})) \text{you.pro} (\text{to (get.v that.pro done.a)})))\]
   → “I want and expect you to get that done”

4. Select closest match between the gold inferences and the available rewrite sentences
   a. Allow 3 character edit distance between gold inference and inferred sentence to allow minor English generation errors.
Automatic Evaluation (Recall)

Sentence: “How soon can you get that done?”

Inferred ULFs:
(sub (adv-e (how.mod-a soon.a)))
(pres can.aux-v) you.pro (get.v that.pro done.a *h)) ?)

Gold Inferences:
“You can get that done”
“I want and expect you to get that done”

<table>
<thead>
<tr>
<th></th>
<th>cf</th>
<th>cls</th>
<th>req</th>
<th>q</th>
<th>oth</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>1/13 (8%)</td>
<td>1/33 (3%)</td>
<td>33/97 (34%)</td>
<td>69/316 (22%)</td>
<td>7/130 (5%)</td>
<td>112/662 (18%)</td>
</tr>
</tbody>
</table>

cf: counterfactual  
cls: clause-taking  
req: request  
oth: other  
q: question
Results are low…
but consider simple baseline’s performance
Takeaways
The form close to syntax allowed evaluation over English using reliable generation.
Takeaways

- The form close to syntax allowed evaluation over English using reliable generation.

- The underlying semantic coherence allows the construction of inference rules, though with an additional interface to handle the syntax.
Conclusions

- We presented the first known method of generating inferences from ULF.
Conclusions

● We presented the first known method of generating inferences from ULF.
● We presented a method of collecting human elicitations of restricted categories of structural inferences, allowing a novel forward inference evaluation.
Conclusions

- We presented the first known method of generating inferences from ULF.
- We presented a method of collecting human elicitations of restricted categories of structural inferences, allowing a novel forward inference evaluation.
- 68.5% of the inferences were judged as reasonable by human judges.
Conclusions

- We presented the first known method of generating inferences from ULF.
- We presented a method of collecting human elicitations of restricted categories of structural inferences, allowing a novel forward inference evaluation.
- 68.5% of the inferences were judged as reasonable by human judges.
- Our experiments demonstrate some of the advantages of using a semantic representation closer to the syntactic form such as ULF—reliable translation to English and access to syntactic signals—though this comes at the cost of a more complicated interface with the semantic patterns.
Conclusions

- We presented the first known method of generating inferences from ULF.
- We presented a method of collecting human elicitations of restricted categories of structural inferences, allowing a novel forward inference evaluation.
- 68.5% of the inferences were judged as reasonable by human judges.
- Our experiments demonstrate some of the advantages of using a semantic representation closer to the syntactic form such as ULF—reliable translation to English and access to syntactic signals—though this comes at the cost of a more complicated interface with the semantic patterns.
- Improvements in the human elicitation procedure and implementation of the inference system (e.g. clause-taking verbs) are clear areas of future work. A larger and more refined dataset of inference elicitations will likely allow the development of a robust inference system.
We would like to thank the paper reviewers for their thoughtful feedback. This work was supported by DARPA CwC subcontract W911NF-15-1-0542.
Poor performance on counterfactual and clause-taking categories due to few examples

“he said he would give a ruble to anyone who found a hare” → “A hare”

Needs improved sampling and larger dataset
Analysis & Discussion

Poor performance on counterfactual and clause-taking categories due to few examples

“he said he would give a ruble to anyone who found a hare” → “A hare”

Needs improved sampling and larger dataset

Annotator disagreements on usage of certainty words

probably, likely, [absence of any], etc.

Move this to a separate likelihood metric of inferences
Disagreements on the boundary of request and questions

“Could you open the door?” ?→? “You know whether you could open the door”
Analysis & Discussion

Disagreements on the boundary of request and questions

“Could you open the door?” → “You know whether you could open the door”

Some remaining ULF to English errors

Me have a wife (subject/object pronouns)

It will entail a radical departure from current policies. (certain pluralizations)
Evaluation of rewrite module:
Evaluation of rewrite module:
- Sampled 100 sentences from final inferred sentences that were closest to gold inferences
Evaluation of rewrite module:
- Sampled 100 sentences from final inferred sentences that were closest to gold inferences
- Evaluated whether the sentence is a valid rewriting of the original inference(s)
Evaluation of rewrite module:
- Sampled 100 sentences from final inferred sentences that were closest to gold inferences
- Evaluated whether the sentence is a valid rewriting of the original inference(s)
- A valid rewriting does not introduce any new semantic information
Evaluation of rewrite module:
- Sampled 100 sentences from final inferred sentences that were closest to gold inferences
- Evaluated whether the sentence is a valid rewriting of the original inference(s)
- A valid rewriting does not introduce any new semantic information

91/100 sentences were valid rewritings
Inference Evaluation

1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

● Filtering patterns
   ○ (if-then) “if something <past tense/participle> something <future marking word> something”
   ○ (inverted if-then) “something <future marking word> something if something <past tense/participle>”

“How soon can you get that done?”
1. Gather multi-genre dataset
   a. Sentences filtered by superficial patterns to reduce annotation overhead

2. Elicit human inferences in each category
   a. Reduced noise with structured guidance

3. Collect gold ULF annotations of each sentence

4. Use inference rules to make conclusions

5. Automatically generate English from ULF inferences

6. Evaluate
   a. precision using human judgments
   b. recall using automatic matching
Episodic Logic

- Extended FOL
- Closely matches expressivity of natural languages
  - Predicates, connectives, quantifiers, equality $\rightarrow$ FOL
  - Predicate and sentence modification (*e.g.* very, gracefully, nearly, possibly)
  - Predicate and sentence reification (*e.g.* Beauty is subjective, That exoplanets exist is now certain)
  - Generalized quantifiers (*e.g.* most men who smoke)
  - Intensional predicates (*e.g.* believe, intend, resemble)
  - Reference to events and situations (*Many children had not been vaccinated against measles; this situation caused sporadic outbreaks of the disease*)

- Suitable for deductive, uncertain, and Natural-Logic-like inference
- A fast and comprehensive theorem prover, EPILOG, is already available.
ULF? (semantics)

A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”, “Could you dial for me?”

ULFs

\[
\begin{align*}
|\text{Alice}| & \text{((pres think.v)} \\
& \text{(that } |\text{John}| \text{ (nearly.adv-a (past fall.v))))} \\
((\text{pres could.aux-v)} & \text{you.pro} \\
& \text{(dial.v } \text{ref1}.pro \text{ (adv-a (for.p me.pro)))))
\end{align*}
\]

\[
\begin{align*}
\text{Entity}(\mathcal{D}): |\text{Alice}|, |\text{John}|, \text{you.pro, ref1}.pro, \text{me.pro} \\
\text{n-ary predicate}(\mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow 2)): \text{think.v, fall.v, dial.v, for.p}
\end{align*}
\]

Basic Ontological Types

- Domain \( \mathcal{D} \)
- Situations \( \mathcal{S} \)
- Truth-value \( 2 \)

Monadic Predicate \( \mathcal{N} : \mathcal{D} \rightarrow (\mathcal{S} \rightarrow 2) \)
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”, “Could you dial for me?”

ULFs

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

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

Entity(\(\mathcal{D}\)): |Alice|, |John|, you.pro, \{ref1}.pro, me.pro

n-ary predicate(\(\mathcal{D}^n \to (\mathcal{S} \to 2)\)): think.v, fall.v, dial.v, for.p

Predicate modifier(\(\mathcal{N} \to \mathcal{N}\)): nearly.adv-a, (adv-a (for.p me.pro))
ULF? (semantics)

A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”, “Could you dial for me?”

ULFs

\[
\begin{align*}
\text{(|Alice| (pres think.v)} & \text{ (that (|John| (nearly.adv-a (past fall.v))))))} \\
\text{((pres could.aux-v) you.pro)} & \text{ (dial.v {ref1}.pro (adv-a (for.p me.pro))) ?)}
\end{align*}
\]

**Entity**(D): |Alice|, |John|, you.pro, {ref1}.pro, me.pro

**n-ary predicate**(\(\mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow 2)\)): think.v, fall.v, dial.v, for.p

**Predicate modifier**(\(\mathcal{N} \rightarrow \mathcal{N}\)): nearly.adv-a, (adv-a (for.p me.pro))

**Sentence reifier**((\(\mathcal{S} \rightarrow 2\) → \(\mathcal{D}\))): that
A minimal step across from syntax to semantics in Episodic Logic

“Alice thinks that John nearly fell”, “Could you dial for me?”

ULFs

\[
\begin{align*}
|Alice| \ ((\text{pres} \ \text{think.v}) \\
\quad \quad \quad (\text{that} \ ((|John| \ (\text{nearly.adv-a} \ (\text{past} \ \text{fall.v}))))) \\
((\text{pres} \ \text{could.aux-v}) \ \text{you.pro} \\
\quad \quad \quad (\text{dial.v} \ \{\text{ref1}.pro \ (\text{adv-a} \ (\text{for.p} \ \text{me.pro}))))) \ ?)
\end{align*}
\]

Entity(\(\mathcal{D}\)): |Alice|, |John|, you.pro, {ref1}.pro, me.pro

n-ary predicate(\(\mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow \mathcal{2})\)): think.v, fall.v, dial.v, for.p

Predicate modifier(\(\mathcal{N} \rightarrow \mathcal{N}\)): nearly.adv-a, (adv-a (for.p me.pro))

Sentence reifier((\(\mathcal{S} \rightarrow \mathcal{2}\) \rightarrow \(\mathcal{D}\))): that

Basic Ontological Types

- Domain (\(\mathcal{D}\))
- Situations (\(\mathcal{S}\))
- Truth-value (\(\mathcal{2}\))

Monadic Predicate: \(\mathcal{N} : \mathcal{D} \rightarrow (\mathcal{S} \rightarrow \mathcal{2})\)

Also... determiner, sentence modifier, connective, lambda abstract, predicate reifier