Monotonic Inference for Underspecified Episodic Logic

Mandar Juvekar
University of Rochester

Natural Logic Meets Machine Learning
17 July 2020
“abelard sees a carp”
“every carp is a fish”

Lambek Derivations
Tableau-style proofs

Replace Lambek derivations and sentences with ULFs

(\text{\text{\text{nAbelard}} (see.v (a.d fish.n)))}

(\text{\text{nAbelard}} (see.v (a.d carp.n)))
**Episodic Logic (EL)**
An extended FOL that closely matches the form and expressivity of natural language.

**Unscoped Logical Form (ULF)**
An underspecified form of EL. Specifies semantic type structure while leaving scope, anaphora, and word sense unresolved.
"Adam placed John under arrest."
Typical EL Inference

Unscoped episodic logical forms are fully resolved before inference
Premises: abelard sees a carp, every carp is a fish

Interpret:

\[ \text{MAJ}(\phi^-), \text{MIN}(\phi^+) \]
\[ \text{MAJ}_\sigma(-\text{MIN}_\sigma(\bot^+)^-) \]
\[ \text{MAJ}(\phi^-), \text{MIN}(\phi^+) \]
\[ \text{MIN}_\sigma(\text{MAJ}_\sigma(\top^-)^+) \]

Infer:

Conclude: abelard sees a fish
Key Observation
ULF provides the structural foundation for monotonic inference
“Ali does not know that I work with a dog”
Ali does not know that I work with a dog

“Ali does not know that I work with a dog”
“Ali does not know that I work with a dog”
“Ali does not know that I work with a dog”
We need semantic argument structure

“Some man holds no apple”

Some: (+,+), No: (-,-)
We need semantic argument structure

“Some man holds no apple”

Some: (+,+)      No: (-,-)

Some man touches no apple

Some man holds no apple

((Some man) (holds (no apple)))  Grammatical
We need semantic argument structure

“Some man holds no apple”

Some: (+,+), No: (-,-)

Some man touches no apple

Some man holds no apple

Some man clenches no apple
Proposal

Directly use ULFs as the basis for inference
Scope Marking

Sánchez Valencia

abelard sees a fish

abelard sees (a fish)
Scope Marking

Sánchez Valencia

ULF

1. every dog sees a fish
2. ((every.d dog.n) (see.v (a.d fish.n))) 
   ULF of 1.
3. (every.d \( x: (x \text{ dog.n}) \))
   (a.d \( y: (y \text{ fish.n}) \))
   \( (x \text{ see.v } y) \))
   Scope every.d above a.d
4. ((every.d dog.n)# (see.v (a.d fish.n)))
   Marking
Polarity Marking

Sánchez Valencia

abelard sees a carp

abelard sees \((a \text{ carp}^+)\)
Polarity Marking

Sánchez Valencia

ULF

1. (lAbelardl (see.v (a.d carp.n)))  Assumption
2. ((every.d carp.n) (be.v (= (a.d fish.n))))  Assumption
3. (a.d \(x\): (x carp.n)\(^+\) (lAbelardl (see.v \(x\)\(^+\))\(^+\)))  SLF of 1. w/ polarity
4. (lAbelardl (see.v (a.d carp.n)\(^+\)))  Pol marking 1., 3.
Inference Rules

Monotonicity

\((\text{every } x)^\# \text{ is a } y, F(x^+), X \xrightarrow{\bullet} Y\)

\((\text{every } x)^\# \text{ is a } y, F(y), X \xrightarrow{\bullet} Y\)
Inference Rules

Monotonicity

\((\text{every } x)^\# \text{ is a } y, F(x^+), X \downarrow Y\)

\((\text{every } x)^\# \text{ is a } y, F(y), X \downarrow Y\)
Inference Rules

**Monotonicity**

\[(\text{every } x)^\# \text{ is a } y, F(x^+), X \bullet Y\]

\[(\text{every } x)^\# \text{ is a } y, F(y), X \bullet Y\]

**Monotonicity (UMI)**

\[\phi[(\delta P1)^+], ((\text{every.d } P1) \text{ (be.v (= (a.d P2))))}\]

\[\phi[(\delta P2)]\]

where \(\delta\) is a determiner.
Inference Rules

**Monotonicity (UMI)**

\[
\phi[(\delta \ P1)^+], \ ((\text{every} \ d \ P1) \ (\text{be} \ v \ (= \ (\text{a} \ d \ P2)))) \\
\phi[(\delta \ P2)]
\]

where \( \delta \) is a determiner.

1. \((\text{l'Abelardl (see} \ v \ (\text{a} \ d \ \text{carp.n}))\)) Assumption

2. \(((\text{every} \ d \ \text{carp.n}) \ (\text{be} \ v \ (= \ (\text{a} \ d \ \text{fish.n}))))\)) Assumption

3. \((\text{a} \ d \ x: \ (x \ \text{carp.n})^+ \\
(\text{l'Abelardl (see} \ v \ x)^+)^+ \)) SLF of 1. w/ polarity

4. \((\text{l'Abelardl (see} \ v \ (\text{a} \ d \ \text{carp.n})^+))\)) Pol marking 1., 3.

5. \((\text{l'Abelardl (see} \ v \ (\text{a} \ d \ \text{fish.n}))\)) UMI 2., 4.
Inference Rules

**Conversion**

\[(\text{some } y)^\# \text{ is a } x, X \downarrow Y\]

\[(\text{some } x)^\# \text{ is a } y, X \downarrow Y\]

**Conversion (UCI)**

\[
\frac{(d_1 \ P) \ (\text{be.v} = (d_2 \ Q)))}{(d_1 \ Q) \ (\text{be.v} = (d_2 \ P)))}
\]

where \(d_1 \in \{\text{some.d, a.d, no.d}\}\) and \(d_2 \in \{\text{some.d, a.d}\}\).
Inference Rules

**Conversion**

\[(\text{some } y) \# \text{ is } a \ x, X \vdash Y\]

\[(\text{some } x) \# \text{ is } a \ y, X \vdash Y\]

**Conversion (UCI)**

\[
\frac{((d_1 \ P) \ (\text{be.v } (= (d_2 \ Q))))}{((d_1 \ Q) \ (\text{be.v } (= (d_2 \ P))))} \quad \text{where } d_1 \in \{\text{some.d, a.d, no.d}\} \quad \text{and } d_2 \in \{\text{some.d, a.d}\}.
\]

1. \[((\text{every.d } S) \ (\text{be.v } (= (\text{a.d } P))))\] Assumption
2. \[((\text{some } S) \ (\text{be.v } (= (\text{a.d } M))))\] Assumption
3. \[((\text{some } S)^+ \ (\text{be.v } (= (\text{a.d } M))))\] Polarity marking, 2.
4. \[((\text{some } P) \ (\text{be.v } (= (\text{a.d } M))))\] UMI, 1., 3.
5. \[((\text{some } M) \ (\text{be.v } (= (\text{a.d } P))))\] Conversion, 4.
Generalized Inference

Rule Instantiation (EL)

“Every carp is a fish” \( (\text{every} \cdot d \ x: (x \text{ carp.n})^- \ (x \text{ fish.n})^+) \)

“Abelard sees a carp” \( (\text{a.d} \ y: (y \text{ carp.n})^+ \ (\text{lAbelardl (see.v y)})^+) \)
Generalized Inference

**Rule Instantiation (EL)**

1. Select logical fragments with opposing polarities

“Every carp is a fish”

“Abelard sees a carp”
Generalized Inference

**Rule Instantiation (EL)**

1. Select logical fragments with opposing polarities
   - “Every carp is a fish” (every.d x: (x carp.n)⁻ (x fish.n)⁺)
   - “Abelard sees a carp” (a.d y: (y carp.n)⁺ (lAbelardl (see.v y))⁺)
2. Matchably bind the two fragments (fail if unable)
   \[(x \rightarrow y)\]
Generalized Inference

Rule Instantiation (EL)

1. Select logical fragments with opposing polarities

2. Matchably bind the two fragments (fail if unable)

3. Convert the formula with the negative polarity fragment

"Every carp is a fish"  (every.d x: (x carp.n)⁻ (x fish.n)⁺)

"Abelard sees a carp"  (a.d y: (y carp.n)⁺ (lAbelard (see.v y))⁺)

\[(x \to y)\]

\[(y \text{ carp.n}) \to \top\]

\[\top \to (y \text{ fish.n})^+\]
Generalized Inference

**Rule Instantiation (EL)**

1. Select logical fragments with opposing polarities
2. Matchably bind the two fragments (fail if unable)
3. Convert the formula with the negative polarity fragment

"Every carp is a fish"  
(y carp.n) \( \rightarrow \) T

"Abelard sees a carp"  
(\( x \rightarrow y \))

\( (x \rightarrow y) \)

(y carp.n) \( \rightarrow \) T

\( \top \rightarrow (y \text{ fish.n})^+ = (y \text{ fish.n})^+ \)
Generalized Inference

**Rule Instantiation (EL)**

1. Select logical fragments with opposing polarities

2. Matchably bind the two fragments (fail if unable)

3. Convert the formula with the negative polarity fragment

\[(\text{a.d } y: (y \text{ carp.n})^+)\]
\[(\text{!Abelardl (see.v } y))^+)\]

\[(y \text{ fish.n})^+]\]
Generalized Inference

Rule Instantiation (EL)

1. Select logical fragments with opposing polarities
2. Matchably bind the two fragments (fail if unable)
3. Convert the formula with the negative polarity fragment
4. Substitute converted formula for other match

"Abelard sees a fish"
Generalized Inference

**Rule Instantiation (EL)**

1. Select logical fragments with opposing polarities

2. Matchably bind the two fragments (fail if unable)

3. Convert the formula with the negative polarity fragment

4. Substitute converted formula for other match

\[
\text{MAJ: } (\text{every.d } x: (x \text{ carp.n})^- (x \text{ fish.n})^+) \\
\text{MIN: } (\text{a.d } y: (y \text{ carp.n})^+ (\neg \text{Abelardl (see.v } y)^+))
\]

```
"Abelard sees a fish" (\text{a.d } y: (y \text{ fish.n})^+ (\neg \text{Abelardl (see.v } y)^+)
```
Generalized Inference

**Rule Instantiation (EL)**

\[
\frac{\text{MAJ}(\phi^-), \text{MIN}(\phi'^+)}{\text{MAJ}_\sigma(\neg \text{MIN}_\sigma(\bot^+)^-)}
\]

\[
\frac{\text{MAJ}(\phi^-), \text{MIN}(\phi'^+)}{\text{MIN}_\sigma(\text{MAJ}_\sigma(\top^-)^+)}
\]

generalizes

**ULF Monotonic Inference**

\[
\phi[(\delta \ P1)^+], ((\text{every}.d \ P1) (\text{be}.v (= (a.d \ P2))))
\]

\[
\phi[(\delta \ P2)^-], ((\text{every}.d \ P1) (\text{be}.v (= (a.d \ P2))))
\]

where \(\delta\) is a determiner.
Benefits

● Reduce sources of parsing error
● Dynamically choose scoping assumptions
● Retain a record of assumptions and inferences
● Simple interface to surface form
Integration with ML

- ULF was designed for ease of ML-based parsing. Parser under review with similar performance to initial AMR parsers.
- ML-assisted ambiguity resolution (e.g. scopes, word sense, polarity)
- Retain semantic type and polarity coherence for interpretable inferences.
Thanks!