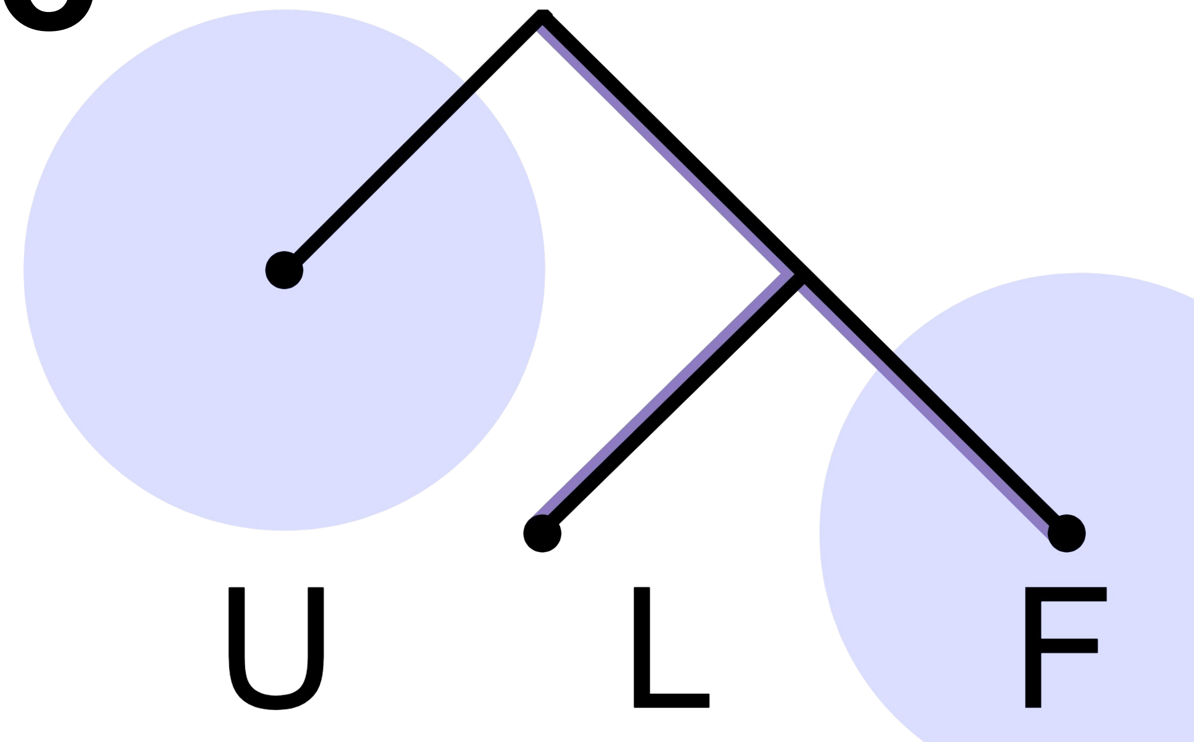
Monotonic Inference for Underspecified Episodic Logic

Mandar Juvekar
University of Rochester

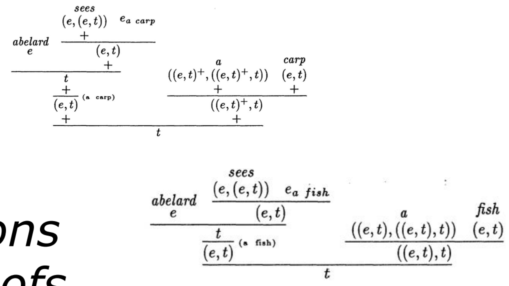
Natural Logic Meets Machine Learning
17 July 2020



Gene Louis Kim
UR

Lenhart K. Schubert
UR

Sánchez Valencia



Lambek Derivations
Tableau-style proofs

"abelard sees a carp"
"every carp is a fish"

"abelard sees a fish"

Replace Lambek derivations
and sentences with ULFs

(|Abelard| (see.v (a.d fish.n)))

(|Abelard| (see.v (a.d carp.n)))

ULF

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| ((past place.v) |John| (under.p (k arrest.n))))

"Adam placed John under arrest."

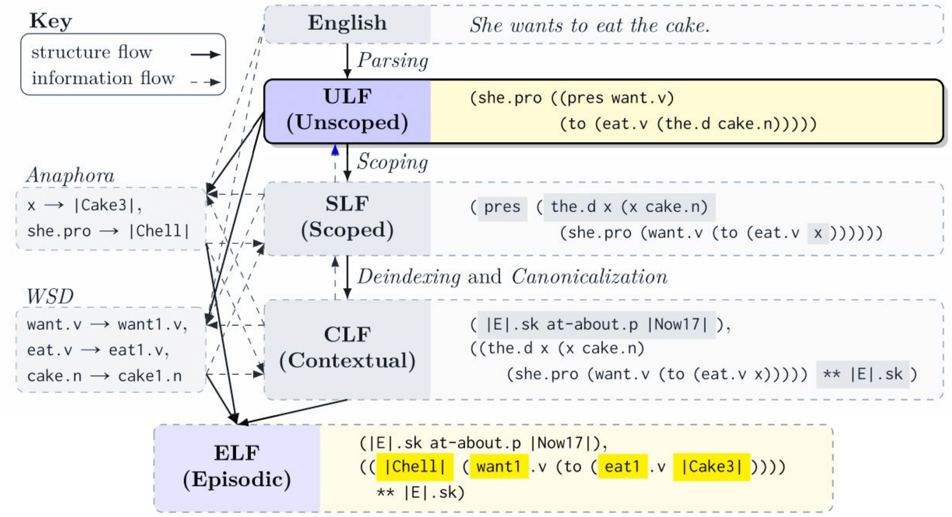
Typical EL Inference

Unscoped episodic logical forms are fully resolved before inference

Premises

abelard sees a carp, every carp is a fish

Interpret



Infer

$$\frac{MAJ(\phi^-), MIN(\phi'^+)}{MAJ_\sigma(\neg MIN_\sigma(\perp^+)^-)}$$

$$\frac{MAJ(\phi^-), MIN(\phi'^+)}{MIN_\sigma(MAJ_\sigma(\top^-)^+)}$$

Conclude

abelard sees a fish

Key Observation

ULF provides the structural foundation for monotonic inference

(|Ali| (do.aux-s not (know.v (that (i.pro (work.v (adv-a (with.p (a.d dog.n))))))))))

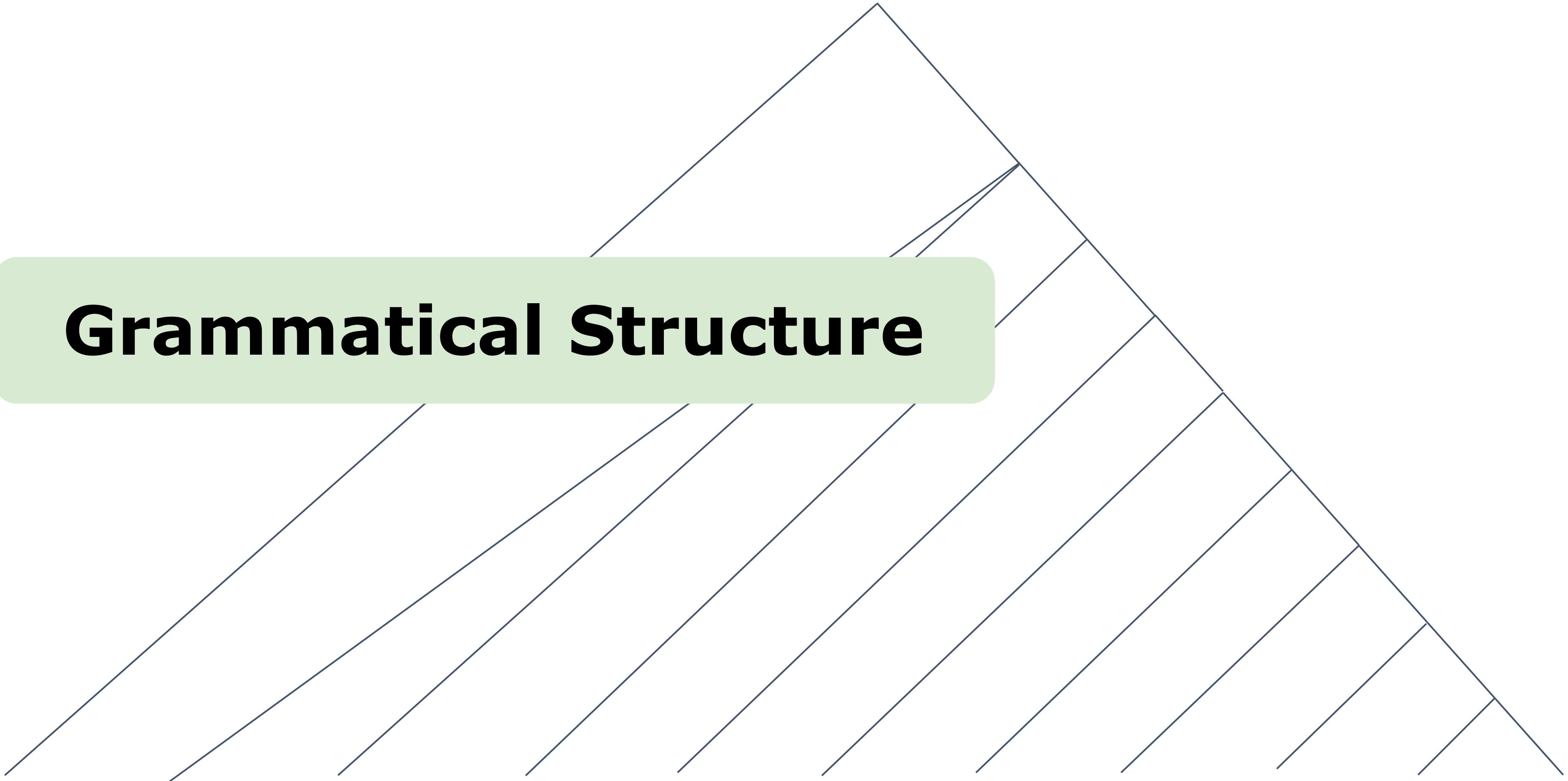
"Ali does not know that I work with a dog"

Preserved Word Order

(|Ali| (do.aux-s not (know.v (that (i.pro (work.v (adv-a (with.p (a.d dog.n)))))))))

"Ali does not know that I work with a dog"

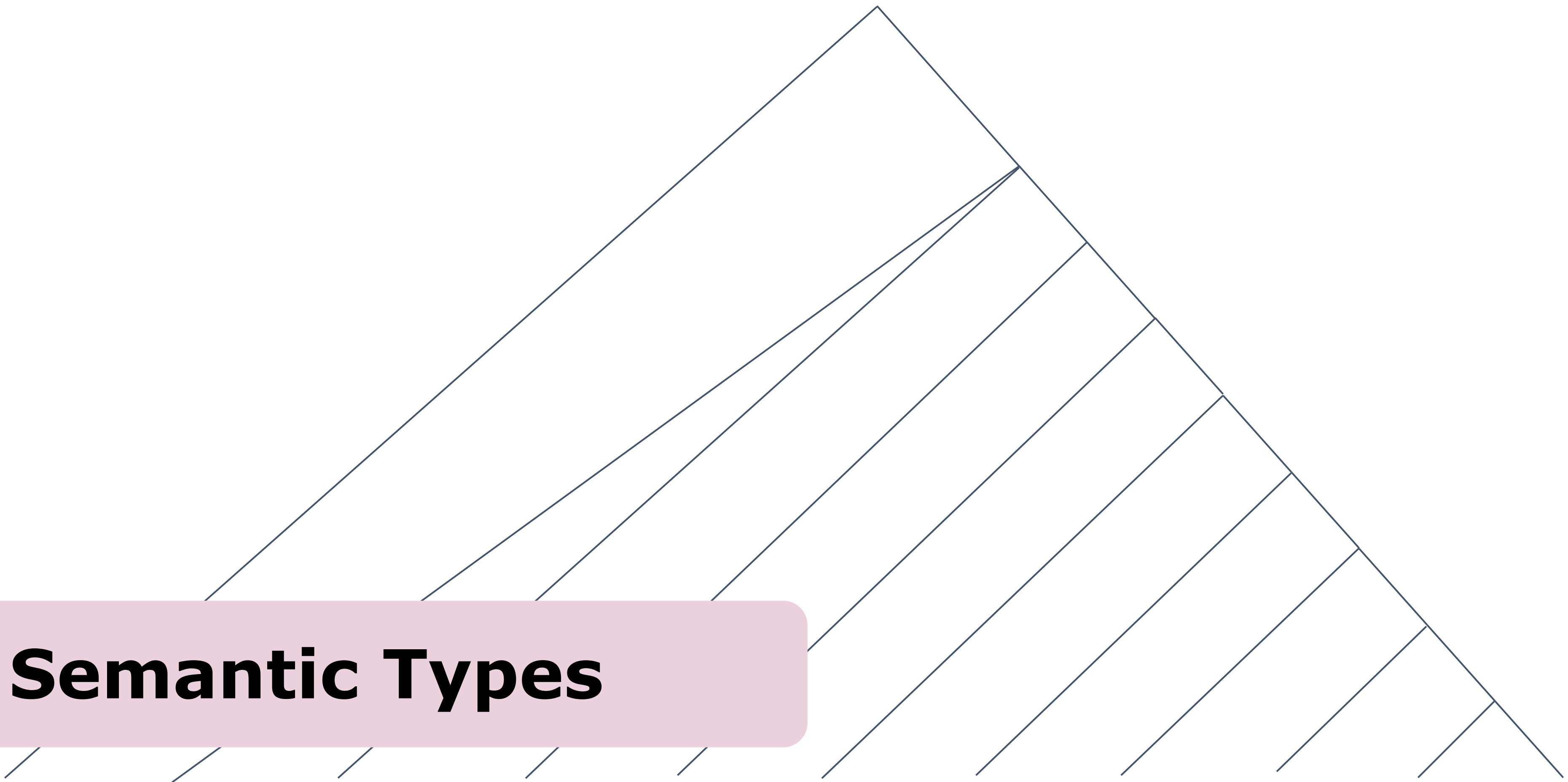
Grammatical Structure



(|Ali| (do.aux-s not (know.v (that (i.pro (work.v (adv-a (with.p (a.d dog.n)))))))))

"Ali does not know that I work with a dog"

Semantic Types



(|Ali| (do.aux-s not (know.v (that (i.pro (work.v (adv-a (with.p (a.d dog.n)))))))))

e ⟨t′,t′⟩ ⟨t′,t′⟩ ⟨e,⟨e,t′⟩⟩ ⟨t′,e⟩ e ⟨e,⟨e,t′⟩⟩ ⟨⟨e,t′⟩,e⟩ ⟨e,⟨e,t′⟩⟩ ⟨⟨e,t′⟩,e⟩ ⟨e,t′⟩

“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

((Some man) (holds (no apple)))

Grammatical

(Some x: (x man)
(no y: (y apple)
(x holds y)))

Semantic

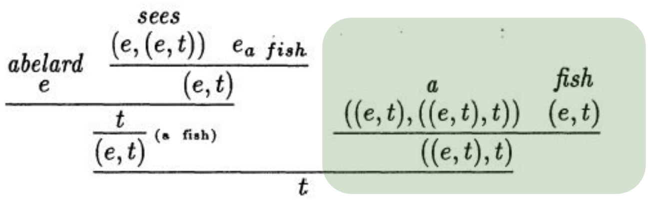
Proposal

Directly use ULFs as the basis for inference

Scope Marking

Sánchez Valencia

abelard sees a fish



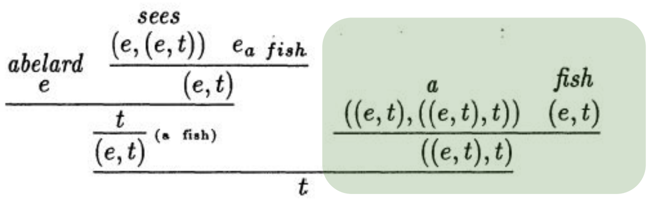
Label

- *abelard sees a fish*
- *abelard sees (a fish)*$^{\#}$ marking

Scope Marking

Sánchez Valencia

abelard sees a fish



Label

abelard sees a fish

abelard sees (a fish)$^{\#}$ marking

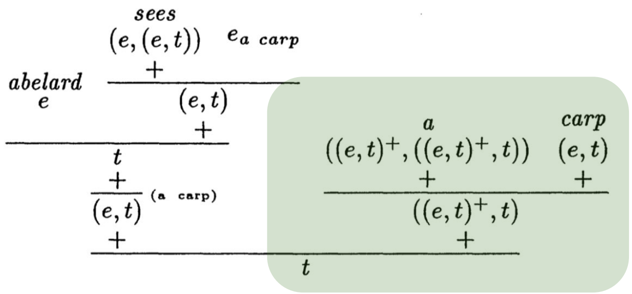
ULF

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 dog.n) (a.d y: (y fish.n) (x 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

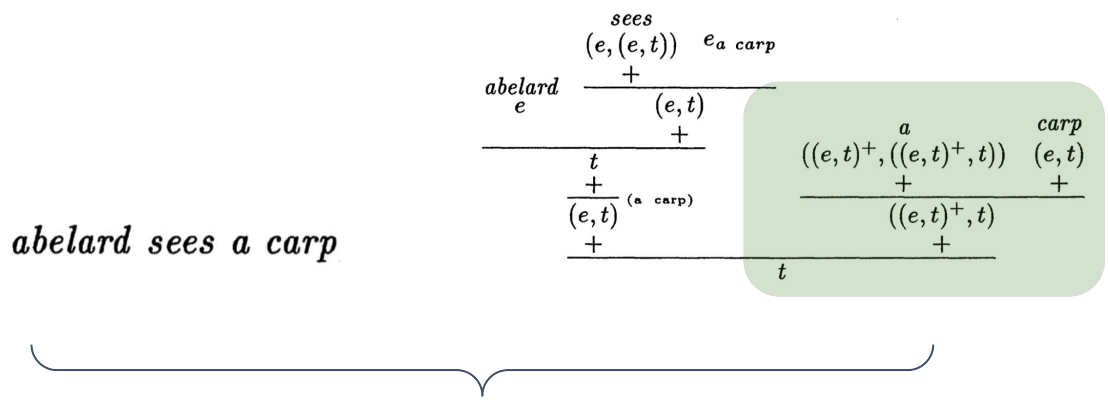
abelard sees a carp



$abelard\ sees\ (a\ carp^{+})^{\#}$

Polarity Marking

Sánchez Valencia



$abelard\ sees\ (a\ carp^+)^{\#}$

ULF

1. (|Abelard| (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)$^+$ (|Abelard| (see.v x)$^+$)$^+$) — SLF of 1. w/ polarity
4. (|Abelard| (see.v (a.d carp.n)$^+$)) — Pol marking 1.,3.

Inference Rules

Monotonicity

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

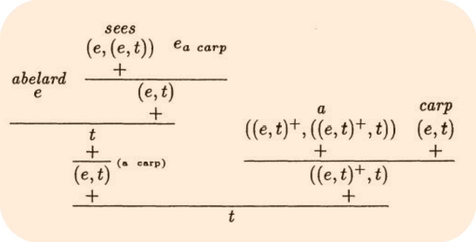
$$(every\ x)^{\#}\ is\ a\ y, F(y), X \bullet Y$$

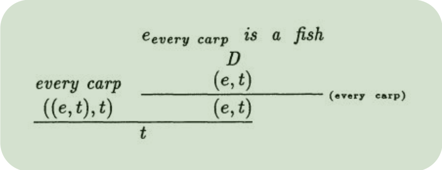
Inference Rules

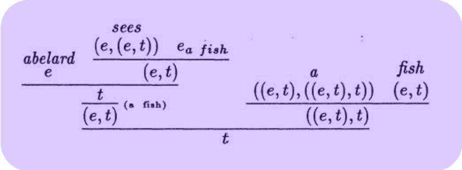
Inference 1 *abelard sees a carp*, *every carp is a fish* / *abelard sees a fish*

Monotonicity

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







abe see a carp, every carp is a fish	$\bullet$	*abe see a fish*	
abe sees (*a carp*)$^{\#}$, (*every carp*)$^{\#}$ is a fish	$\bullet$	*abe sees* (*a fish*)$^{\#}$	**marking**
abe sees (*a carp*$^{+}$)$^{\#}$, (*every carp*)$^{\#}$ is a fish	$\bullet$	*abe sees* (*a fish*)$^{\#}$	**marking**
abe sees (*a fish*)$^{\#}$, (*every carp*)$^{\#}$ is a fish	$\bullet$	*abe sees* (*a fish*)$^{\#}$	**monotonicity**

Inference Rules

Monotonicity

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

Monotonicity (UMI)

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

where δ is a determiner.

Inference Rules

Monotonicity (UMI)

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

where δ is a determiner.

1.	(|Abelard| (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)$^{+}$ (|Abelard| (see.v x)$^{+}$)$^{+}$)	SLF of 1. w/ polarity
4.	(|Abelard| (see.v (a.d carp.n)$^{+}$))	Pol marking 1.,3.
5.	(|Abelard| (see.v (a.d fish.n)))	UMI 2.,4.

Inference Rules

Conversion

$$\begin{array}{l} (some\ y)^{\#}\ is\ a\ x, X \bullet Y \\ (some\ x)^{\#}\ is\ a\ y, X \bullet Y \end{array}$$

Conversion (UCI)

$$\frac{((d1\ P)\ (\text{be.v}\ (=\ (d2\ Q))))}{((d1\ Q)\ (\text{be.v}\ (=\ (d2\ P))))} \quad \begin{array}{c} \text{where } d1 \in \{\text{some.d, a.d, no.d}\} \\ \text{and } d2 \in \{\text{some.d, a.d}\}. \end{array}$$

Inference Rules

Conversion

$$\begin{array}{ll} (some\ y)^{\#}\ is\ a\ x, X & Y \\ (some\ x)^{\#}\ is\ a\ y, X & Y \end{array}$$

Conversion (UCI)

$$\frac{((d1\ P)\ (\text{be.v}\ (=\ (d2\ Q))))}{((d1\ Q)\ (\text{be.v}\ (=\ (d2\ P))))}$$ where $d1 \in \{\text{some.d, a.d, no.d}\}$ and $d2 \in \{\text{some.d, a.d}\}$.

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

Generalized Inference

Rule Instantiation (EL)

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

"Abelard sees a carp" (a.d y: $(y \text{ carp.n})^{+}$
$(|\text{Abelard}| \text{ (see.v } y))^{+}$)

Generalized Inference

Rule Instantiation (EL)

1. Select logical fragments with opposing polarities

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

"Abelard sees a carp" (a.d y: $(y \text{ carp.n})^+$
($|$Abelard$|$ (see.v y))$^+$)

Generalized Inference

Rule Instantiation (EL)

1. Select logical fragments with opposing polarities

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

"Abelard sees a carp" (a.d y: $(y \text{ carp.n})^+$ $(|\text{Abelard}| (\text{see.v } y))^+$)

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" $(\text{every.d } x\text{: } (x \text{ carp.n})^{-} \; (x \text{ fish.n})^{+})$

"Abelard sees a carp" $(\text{a.d } y\text{: } (y \text{ carp.n})^{+} \; (|\text{Abelard}| \; (\text{see.v } y))^{+})$

$(x \rightarrow y)$

$(y \text{ carp.n}) \rightarrow \top$

$\top \rightarrow (y \text{ fish.n})^{+}$

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" $(\text{every.d } x\text{: } (x \text{ carp.n})^{-} \ (x \text{ fish.n})^{+})$

"Abelard sees a carp" $(\text{a.d } y\text{: } (y \text{ carp.n})^{+}$
$(|\text{Abelard}| \ (\text{see.v } y))^{+})$

$(x \rightarrow y)$

$(y \text{ carp.n}) \rightarrow \top$

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

Generalized Inference

Rule Instantiation (EL)

1. Select logical fragments with opposing polarities

$$(\text{a.d } y\text{: } (y \text{ carp.n})^{+} \; (|\text{Abelard}| \; (\text{see.v } y))^{+})$$

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

3. Convert the formula with the negative polarity fragment

$$(y \text{ fish.n})^{+}$$

Generalized Inference

Rule Instantiation (EL)

1. Select logical fragments with opposing polarities

$$(\text{a.d } y\text{: } (y \text{ carp.n})^{+} \; (|\text{Abelard}| \; (\text{see.v } y))^{+})$$

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

3. Convert the formula with the negative polarity fragment

$$\Big\downarrow (y \text{ fish.n})^{+}$$

4. Substitute converted formula for other match

"Abelard sees a fish" $(\text{a.d } y\text{: } (y \text{ fish.n})^{+} \; (|\text{Abelard}| \; (\text{see.v } y))^{+})$

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
4. Substitute converted formula for other match

MAJ: (every.d x: $(x \text{ carp.n})^{-}$ $(x \text{ fish.n})^{+}$)

MIN: (a.d y: $(y \text{ carp.n})^{+}$ (|Abelard| (see.v y))$^{+}$)

RI-2

$$\frac{MAJ(\phi^{-}), MIN(\phi'^{+})}{MIN_{\sigma}(MAJ_{\sigma}(\top^{-})^{+})}$$

"Abelard sees a fish" (a.d y: $(y \text{ fish.n})^{+}$ (|Abelard| (see.v y))$^{+}$)

Generalized Inference

Rule Instantiation (EL)

$$\frac{MAJ(\phi^{-}), MIN(\phi'^{+})}{MAJ_{\sigma}(\neg MIN_{\sigma}(\bot^{+})^{-})}$$

$$\frac{MAJ(\phi^{-}), MIN(\phi'^{+})}{MIN_{\sigma}(MAJ_{\sigma}(\top^{-})^{+})}$$

generalizes

ULF Monotonic Inference

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

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

where δ 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!