Natural Theorem Proving for Natural Language

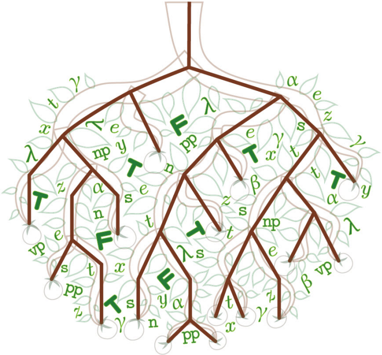
Theory & Application

Lasha Abzianidze

university of groningen

CLASP seminar, Göteborg
12.02.2020

- Semantic tableau method
- Simple type theory
- Natural Tableau
- Scaling up the Natural Tableau
- NL theorem prover
- Experiments
- Conclusion



Semantic tableau method

A semantic tableau method [Beth, 1955, Hintikka, 1955] is a proof procedure for formal logics:

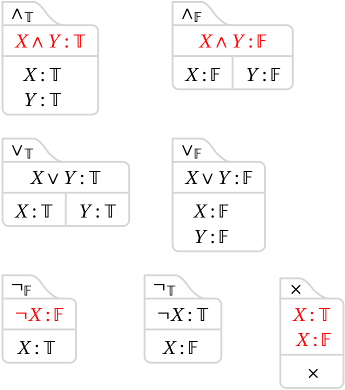
Input: a set of signed formulas
$P_1 : \mathbb{T}, \ldots, P_m : \mathbb{T},\ Q_1 : \mathbb{F}, \ldots, Q_n : \mathbb{F}$
Output: some or no model satisfying the input
☞ A model search problem

Prove φ by failing to refute φ:

Prove: $P \wedge Q \vDash Q \wedge \neg P$
Try: Justify the counterexample $P \wedge Q : \mathbb{T},\ Q \wedge \neg P : \mathbb{F}$

Propositional tableau (signed version)

Propositional tableau rules:



$\wedge_{\mathbb{T}}$

$X \wedge Y : \mathbb{T}$

$X : \mathbb{T}$

$Y : \mathbb{T}$

$\wedge_{\mathbb{F}}$

$X \wedge Y : \mathbb{F}$

$X : \mathbb{F}$ | $Y : \mathbb{F}$

$\vee_{\mathbb{T}}$

$X \vee Y : \mathbb{T}$

$X : \mathbb{T}$ | $Y : \mathbb{T}$

$\vee_{\mathbb{F}}$

$X \vee Y : \mathbb{F}$

$X : \mathbb{F}$

$Y : \mathbb{F}$

$\neg_{\mathbb{F}}$

$\neg X : \mathbb{F}$

$X : \mathbb{T}$

$\neg_{\mathbb{T}}$

$\neg X : \mathbb{T}$

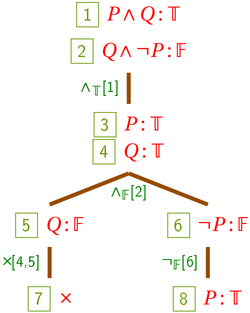
$X : \mathbb{F}$

×

$X : \mathbb{T}$

$X : \mathbb{F}$

×



A situation supporting

a counterexample: $P : \mathbb{T}, Q : \mathbb{T}$

Therefore, $P \wedge Q \models Q \wedge \neg P$ is refuted!

Closed tableau

Prove: $\neg(P \wedge Q) \vDash \neg P \vee \neg Q$ Proved!

Counterexample: $\neg(P \wedge Q) : \mathbb{T}$, $\neg P \vee \neg Q : \mathbb{F}$

Propositional tableau rules:

$\wedge_{\mathbb{T}}$

$X \wedge Y : \mathbb{T}$

$X : \mathbb{T}$
$Y : \mathbb{T}$

$\wedge_{\mathbb{F}}$

$X \wedge Y : \mathbb{F}$

$X : \mathbb{F}$	$Y : \mathbb{F}$

$\vee_{\mathbb{T}}$

$X \vee Y : \mathbb{T}$

$X : \mathbb{T}$	$Y : \mathbb{T}$

$\vee_{\mathbb{F}}$

$X \vee Y : \mathbb{F}$

$X : \mathbb{F}$
$Y : \mathbb{F}$

$\neg_{\mathbb{F}}$

$\neg X : \mathbb{F}$

$X : \mathbb{T}$

$\neg_{\mathbb{T}}$

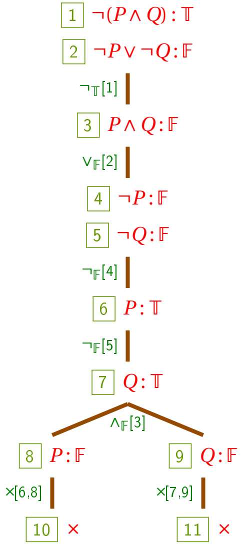
$\neg X : \mathbb{T}$

$X : \mathbb{F}$

×

$X : \mathbb{T}$
$X : \mathbb{F}$

×



Rules for quantifiers

Rules for ∃:

$\exists_{\mathbb{T}}$

$\exists x.\phi : \mathbb{T}$

$\phi[x/c] : \mathbb{T}$

c is fresh

$\exists^{c}_{\mathbb{F}}$

$\exists x.\phi : \mathbb{F}$

$\phi[x/c] : \mathbb{F}$

c is old

Rules for ∀:

$\forall_{\mathbb{F}}$

$\forall x.\phi : \mathbb{F}$

$\phi[x/c] : \mathbb{F}$

c is fresh

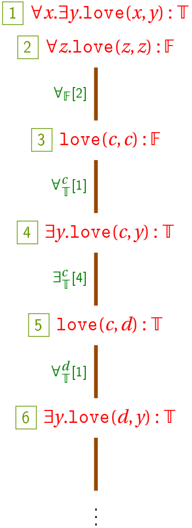
$\forall^{c}_{\mathbb{T}}$

$\forall x.\phi : \mathbb{T}$

$\phi[x/c] : \mathbb{T}$

c is old

⚠ Dangerous zone!



1 $\forall x.\exists y.\mathtt{love}(x,y) : \mathbb{T}$

2 $\forall z.\mathtt{love}(z,z) : \mathbb{F}$

$\forall_{\mathbb{F}}[2]$

3 $\mathtt{love}(c,c) : \mathbb{F}$

$\forall^{c}_{\mathbb{T}}[1]$

4 $\exists y.\mathtt{love}(c,y) : \mathbb{T}$

$\exists^{c}_{\mathbb{T}}[4]$

5 $\mathtt{love}(c,d) : \mathbb{T}$

$\forall^{d}_{\mathbb{T}}[1]$

6 $\exists y.\mathtt{love}(d,y) : \mathbb{T}$

⋮

Simple type theory

Use Simple Type Theory [Church, 1940] as a Higher-Order Logic.

A type system is built up from $\{e, t\} + \{\mathtt{np}, \mathtt{s}, \mathtt{n}, \mathtt{pp}\}$.

- $e, t, \mathtt{np}, \mathtt{s}, \mathtt{n}, \mathtt{pp}$ are types;
- if α and β are types, so are $(\alpha\beta)$

A subtyping as a syntax-semantic interface:

- $\mathtt{s} <: t;\quad e <: \mathtt{np};\quad \mathtt{n} <: et;\quad \mathtt{pp} <: et;$
- $(\alpha_1, \alpha_2) <: (\beta_1, \beta_2)$ iff $\beta_1 <: \alpha_1$ and $\alpha_2 <: \beta_2$

Typing terms:

- **love** of type $\mathtt{np}, (\mathtt{np}, \mathtt{s}) \equiv \mathtt{np}, \mathtt{np}, \mathtt{s} \equiv \mathtt{np}, \mathtt{vp}$
- $\mathbf{love}_{\mathtt{np,vp}}\, x_{\mathtt{np}}$ is of type $\mathtt{np}, \mathtt{s}$
- $\mathbf{love}_{\mathtt{np,vp}}\, x_{\mathtt{np}}\, \mathbf{john}_{\mathtt{np}}$ is of type $\mathtt{s}$
- $\lambda x.\mathbf{love}_{\mathtt{np,vp}}\, x_{\mathtt{np}}\, \mathbf{john}_{\mathtt{np}}$ is of type $\mathtt{vp}$

Interface of syntactic & semantic terms

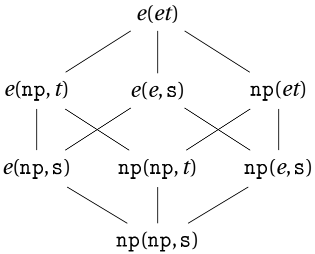
Terms of multiple types:

- **cat**$_{\mathrm{n}}$ is of type et
- **red**$_{\mathrm{n,n}}$ is of type (n, et) ~~and (et, et)~~
- **love**$_{\mathrm{np,np,s}}$ is of type $\mathrm{np}(\mathrm{np}, t)$, eet, . . .

Syntactic and semantic terms together:

- **cat**$_{\mathrm{n}}\, c_e$
- **love**$_{\mathrm{np,np,s}}$**john**$_{\mathrm{np}}\, c_e$
- **on**$_{\mathrm{pp}}\, d_e$

Types of **love**$_{\mathrm{np,np,s}}$



Simple type theory as Natural Logic

Natural logic is a hypothetical logic which is built in natural language and represents its integral part.

It is a theory about "the regularities governing the notion of a valid argument for reasoning in natural language" [Lakoff, 1970].

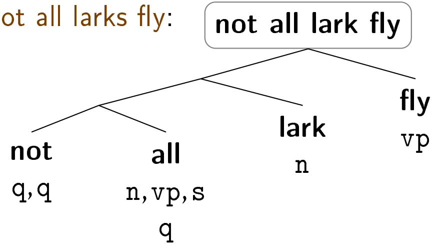
"Natural logic is a somewhat loose [...] term for [...] attempts [...] at describing basic patterns of human reasoning directly in natural language without the intermediate of some formal system" [van Benthem, 2008].

Natural logic is "the study of inference in natural language, done as close as possible to the surface forms" [Moss, 2010].

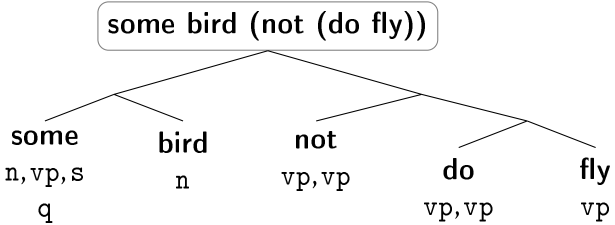
Let's use teh simple type theory as natural logic and call its terms Lambda Logical Forms.

Examples of LLFs

Not all larks fly:



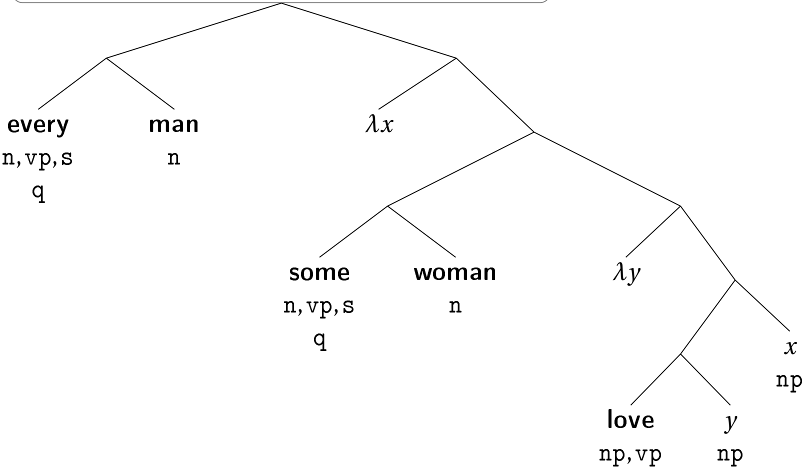
Some bird does not fly:



Zooming in on LLFs (scope ambiguity)

Every man loves some woman:

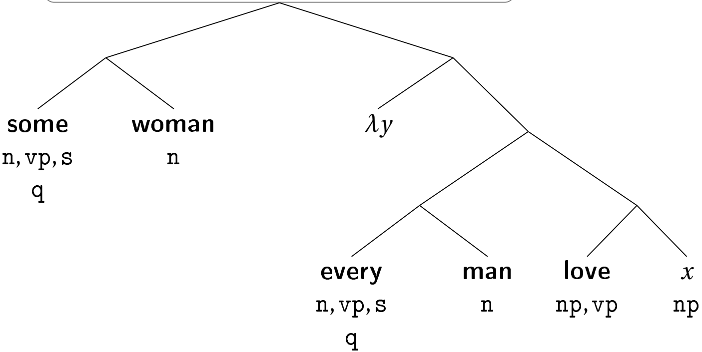
every man $(\lambda x.$ **some woman** $(\lambda y.$ **love** $y\ x))$



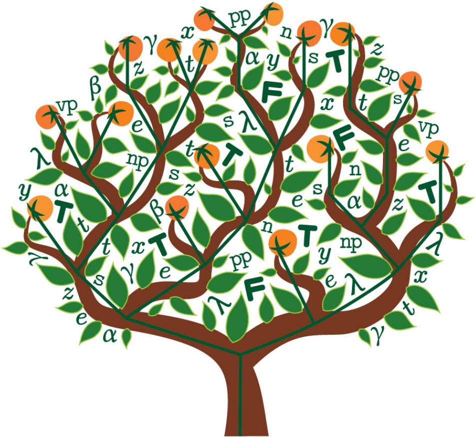
Zooming in on LLFs (scope ambiguity)

Every man loves some woman:

some woman $(\lambda y.$ **every man** (**love** y))



Natural Tableau



An Analytic Tableau System for Natural Logic [Muskens, 2010]

LLFs & tableau entries

LLFs are represented in tableau entries as:

$$\underbrace{\texttt{LLF}:\texttt{argumentList}}_{\text{Binary format of a term}}:\texttt{truthSign}$$

Different binary representations of the same signed term:

$$\texttt{love john}:[\texttt{mary}]:\mathbb{F} \qquad \texttt{love}:[\texttt{john,mary}]:\mathbb{F}$$

Two advantages of `argumentList`:

- Traverse through a recursive structure of a term
- Align the shared arguments and contrast different terms:

$$\texttt{no}:[\texttt{little bird, fly}]:\mathbb{T}$$
$$\texttt{some (littlebird) fly}:[]:\mathbb{T}$$
$$\texttt{some (little bird)}:[\texttt{fly}]:\mathbb{T}$$
$$\texttt{some}:[\texttt{little bird, fly}]:\mathbb{T}$$

Ordering over LLFs

Remember we use types built up from e and t.

We have two truth values 1 (*true*) and 0 (*false*), where $0 < 1$.

With the help of $0 < 1$, we can have (partial) ordering over the terms of type $\vec{\alpha}\,t$:

- **dog**$_{\mathrm{n}}$ is more specific than **animal**$_{\mathrm{n}}$, because for any x, **dog** x is less than or equal to **animal** x
- **kiss**$_{\mathrm{np,vp}}$ is more specific than `touch`$_{\mathrm{np,vp}}$, because for any x, y, **kiss** $x\ y$ is less than or equal to **touch** $x\ y$
- For $A_{\vec{\alpha}\,t}$ and $B_{\vec{\alpha}\,t}$, we define $A \sqsubseteq B \overset{\text{def}}{=} \forall \vec{X}.\ A\vec{X} \leq B\vec{X}$

Sequence/vector notation conventions:

- $\alpha_1 \ldots \alpha_n\, t \equiv \vec{\alpha}\, t$
- $A B_1 B_2 \ldots B_n \equiv A \vec{B}$

Tableau rules in action

A>

$$\frac{A\ B : [\vec{C}] : \mathbb{X}}{A : [B, \vec{C}] : \mathbb{X}}$$

A<

$$\frac{A : [B, \vec{C}] : \mathbb{X}}{A\ B : [\vec{C}] : \mathbb{X}}$$

¬

$$\frac{\text{not } A : [\vec{C}] : \mathbb{X}}{A : [\vec{C}] : \overline{\mathbb{X}}}$$

$\exists^{c}_{\mathbb{F}}$

$$\frac{\text{some } A\ B : [\,] : \mathbb{F}}{A : [c_e] : \mathbb{F} \quad \mid \quad B : [c_e] : \mathbb{F}}$$

c is old

$\forall_{\mathbb{F}}$

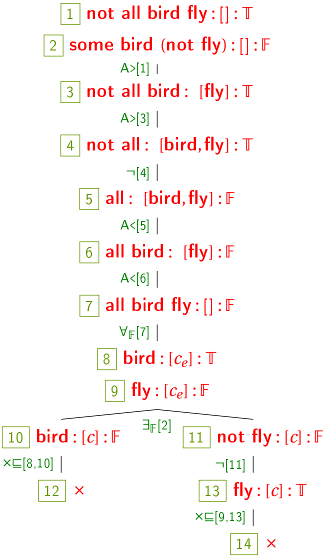
$$\frac{\text{every } A\ B : [\,] : \mathbb{F}}{\begin{array}{c} A : [c_e] : \mathbb{T} \\ B : [c_e] : \mathbb{F} \end{array}}$$

c is fresh

$\times\sqsubseteq$

$$\frac{\begin{array}{c} A : [\vec{C}] : \mathbb{T} \\ B : [\vec{C}] : \mathbb{F} \end{array}}{\times}$$

$A \sqsubseteq B$



1 not all bird fly : [] : $\mathbb{T}$

2 some bird (not fly) : [] : $\mathbb{F}$

A>[1]

3 not all bird : [fly] : $\mathbb{T}$

A>[3]

4 not all : [bird, fly] : $\mathbb{T}$

¬[4]

5 all : [bird, fly] : $\mathbb{F}$

A<[5]

6 all bird : [fly] : $\mathbb{F}$

A<[6]

7 all bird fly : [] : $\mathbb{F}$

$\forall_{\mathbb{F}}$[7]

8 bird : $[c_e]$: $\mathbb{T}$

9 fly : $[c_e]$: $\mathbb{F}$

$\exists_{\mathbb{F}}$[2]

10 bird : $[c]$: $\mathbb{F}$

×⊑[8,10]

12 ×

11 not fly : $[c]$: $\mathbb{F}$

¬[11]

13 fly : $[c]$: $\mathbb{T}$

×⊑[9,13]

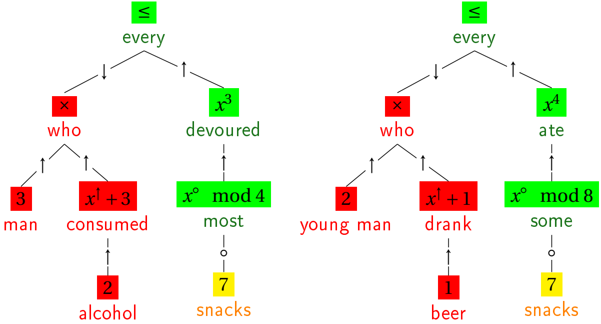
14 ×

Monotonicity reasoning

GOLD: `entailment`

P: $3 \times [x+3](2) \;\leq\; [x^3]\big([x \mod 4](7)\big)$

H: $2 \times [x+1](1) \;\leq\; [x^4]\big([x \mod 8](7)\big)$



GOLD: `entailment`

P: Every man who consumed alcohol devoured most snacks

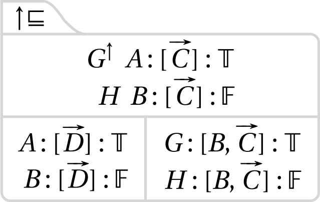
H: Every young man who drank beer ate some snacks

Monotonicity rules (Upward)

Definition (Upward monotonicity)

A function term F of type $(\vec{\alpha}\,t)\vec{\gamma}\,t$ is upward monotone (↑), denoted as $F^{\uparrow}$, if it satisfies one of the following equivalent properties:

$$\forall XY\big((X \sqsubseteq Y) \rightarrow (FX \sqsubseteq FY)\big)$$



$\uparrow\sqsubseteq$

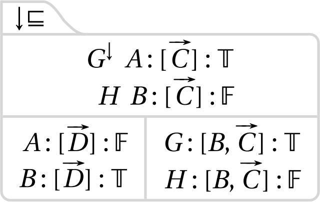
$$\begin{array}{c|c} \multicolumn{2}{c}{G^{\uparrow}\ A : [\vec{C}] : \mathbb{T}} \\ \multicolumn{2}{c}{H\ B : [\vec{C}] : \mathbb{F}} \\ \hline A : [\vec{D}] : \mathbb{T} & G : [B, \vec{C}] : \mathbb{T} \\ B : [\vec{D}] : \mathbb{F} & H : [B, \vec{C}] : \mathbb{F} \end{array}$$

Monotonicity rules (Downward)

Definition (Downward monotonicity)

A function term F of type $(\vec{\alpha}\,t)\vec{\gamma}\,t$ is downward monotone (↓), denoted as $F^{\downarrow}$, if it satisfies one of the following equivalent properties:

$$\forall XY\big((X \sqsubseteq Y) \rightarrow (FY \sqsubseteq FX)\big)$$

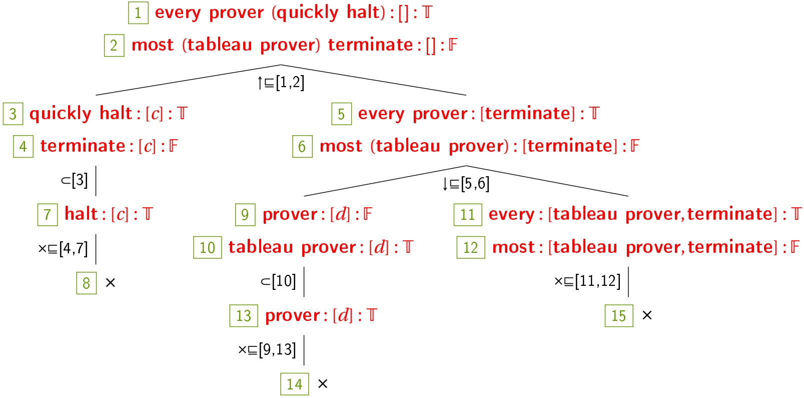


$\downarrow\sqsubseteq$

$$G^{\downarrow}\ A : [\vec{C}] : \mathbb{T}$$
$$H\ B : [\vec{C}] : \mathbb{F}$$

$$\begin{array}{c|c} A : [\vec{D}] : \mathbb{F} & G : [B, \vec{C}] : \mathbb{T} \\ B : [\vec{D}] : \mathbb{T} & H : [B, \vec{C}] : \mathbb{F} \end{array}$$

Monotonicity rules in action



$\boxed{1}$ **every prover (quickly halt)** $: [\,] : \mathbb{T}$

$\boxed{2}$ **most (tableau prover) terminate** $: [\,] : \mathbb{F}$

$\uparrow\sqsubseteq[1,2]$

$\boxed{3}$ **quickly halt** $: [c] : \mathbb{T}$

$\boxed{4}$ **terminate** $: [c] : \mathbb{F}$

$\subset[3]$

$\boxed{7}$ **halt** $: [c] : \mathbb{T}$

$\times\sqsubseteq[4,7]$

$\boxed{8}$ ×

$\boxed{5}$ **every prover** $:$ [**terminate**] $: \mathbb{T}$

$\boxed{6}$ **most (tableau prover)** $:$ [**terminate**] $: \mathbb{F}$

$\downarrow\sqsubseteq[5,6]$

$\boxed{9}$ **prover** $: [d] : \mathbb{F}$

$\boxed{10}$ **tableau prover** $: [d] : \mathbb{T}$

$\subset[10]$

$\boxed{13}$ **prover** $: [d] : \mathbb{T}$

$\times\sqsubseteq[9,13]$

$\boxed{14}$ ×

$\boxed{11}$ **every** $:$ [**tableau prover**, **terminate**] $: \mathbb{T}$

$\boxed{12}$ **most** $:$ [**tableau prover**, **terminate**] $: \mathbb{F}$

$\times\sqsubseteq[11,12]$

$\boxed{15}$ ×

$\uparrow\sqsubseteq$

$$\frac{\begin{array}{c} G^{\uparrow}\ A:[\vec{C}]:\mathbb{T} \\ H\ B:[\vec{C}]:\mathbb{F} \end{array}}{\begin{array}{c} A:[\vec{d}\,]:\mathbb{T} \\ B:[\vec{d}\,]:\mathbb{F} \end{array} \;\Big|\; \begin{array}{c} G:[B,\vec{C}]:\mathbb{T} \\ H:[B,\vec{C}]:\mathbb{F} \end{array}}$$

$\downarrow\sqsubseteq$

$$\frac{\begin{array}{c} G^{\downarrow}\ A:[\vec{C}]:\mathbb{T} \\ H\ B:[\vec{C}]:\mathbb{F} \end{array}}{\begin{array}{c} A:[\vec{d}\,]:\mathbb{F} \\ B:[\vec{d}\,]:\mathbb{T} \end{array} \;\Big|\; \begin{array}{c} G:[B,\vec{C}]:\mathbb{T} \\ H:[B,\vec{C}]:\mathbb{F} \end{array}}$$

$\subset$

$$\frac{A^{\subset}\ N:[\vec{C}]:\mathbb{T}}{N:[\vec{C}]:\mathbb{T}}$$

$$\forall X(A^{\subset} X \sqsubseteq X)$$

$\times\sqsubseteq$

$$\frac{\begin{array}{c} A:[\vec{C}]:\mathbb{T} \\ B:[\vec{C}]:\mathbb{F} \end{array}}{\times}$$

$A \sqsubseteq B$

LLFs and Categorial Grammar

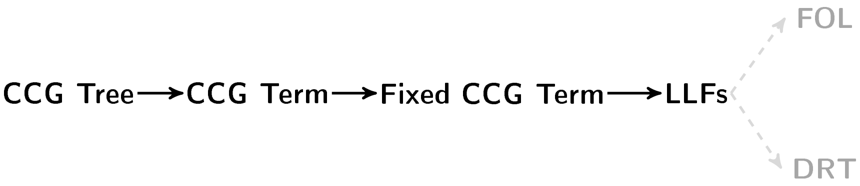
LLFs are similar to formal derivations studied in Categorial Grammars (CGs) [Ajdukiewicz, 1935, Hillel, 1953].

Combinatory Categorial Grammar (CCG) [Steedman, 2000] is the only CG that is scaled up for wide-coverage text processing:

- CCG is well-studied from linguistic perspectives;
- There exists robust and accurate wide-coverage parsers for CCG, e.g., C&C parser [Clark and Curran, 2007] and EasyCCG [Lewis and Steedman, 2014].

LLF generation

Producing an LLF from a CCG derivation consists of several steps:



CCG Tree → **CCG Term** → **Fixed CCG Term** → **LLFs**

FOL

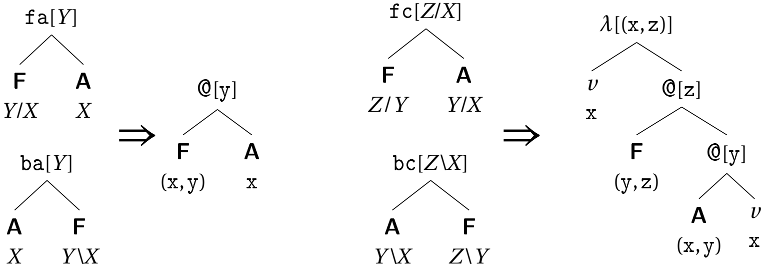
DRT

CCG Tree → CCG term

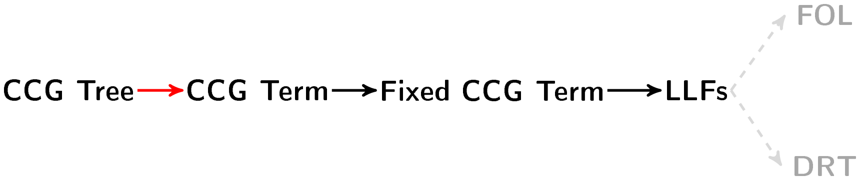
CCG Tree → CCG Term → Fixed CCG Term → LLFs

FOL

DRT



CCG Tree → CCG term



Removing directionality:

$$Y\backslash X \text{ and } Y/X \rightsquigarrow (\mathtt{x},\mathtt{y})$$

$$\mathtt{ba}(A_X, F_{Y\backslash X}) \rightsquigarrow F\,A$$

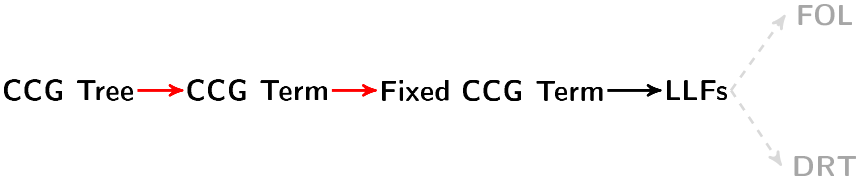
$$\mathtt{fxc}(F_{Z/Y}, A_{Y\backslash X}) \rightsquigarrow \lambda x.\,F(Ax)$$

$$\mathtt{tr}(T/(T\backslash X), A_X) \rightsquigarrow \lambda x.\,xA$$

$$\mathtt{lx}(Y, A_X) \rightsquigarrow [A_X]_Y$$

$$\mathtt{conj}(C_{conj}, A_X) \rightsquigarrow C_{X,X,X}A$$

CCG term → fixed CCG term



Correcting CCG terms:

$$[\mathtt{Dow}^{\mathrm{PER}}_{\mathrm{n,n}}\ \mathtt{Jones}^{\mathrm{PER}}_{\mathrm{n}}]_{\mathrm{np}} \rightsquigarrow \mathtt{Dow\_Jones}_{\mathrm{np}}$$

$$\mathtt{nobody}_{\mathrm{np}} \rightsquigarrow \mathtt{no}_{\mathrm{n,np}}\ \mathtt{person}_{\mathrm{n}}$$

$$[\mathtt{ice}_{\mathrm{n}}]_{\mathrm{np}} \rightsquigarrow \mathtt{a}_{\mathrm{n,np}}\ \mathtt{ice}_{\mathrm{n}}$$

$$[\mathtt{two}_{\mathrm{n,n}}\ \mathtt{dogs}_{\mathrm{n}}]_{\mathrm{np}} \rightsquigarrow \mathtt{two}_{\mathrm{n,np}}\ \mathtt{dogs}_{\mathrm{n}}$$

$$[\mathtt{working}_{\mathrm{np,s}}]_{\mathrm{n,n}} \rightsquigarrow \mathtt{who}_{\mathrm{(np,s),n,n}}\ \mathtt{working}$$

$$\mathtt{who}\ V(Q_{\mathrm{n,np}}N) \rightsquigarrow Q_{\mathrm{n,np}}(\mathtt{who}'\ VN)$$

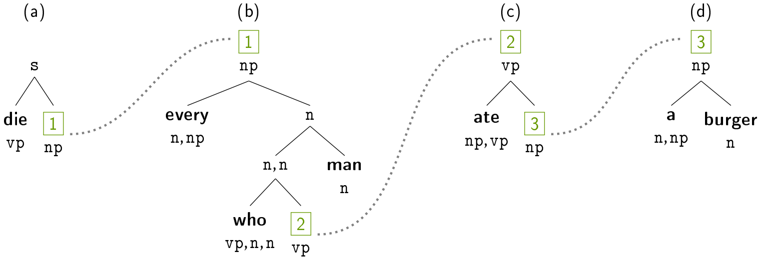
fixed CCG term → LLFs

Every man who ate a burger died

$$\mathbf{die}_{\mathtt{vp}}\ (\mathbf{every}_{\mathtt{n,np}}(\mathbf{who}_{\mathtt{vp,n,n}}\ (\mathbf{eat}_{\mathtt{np,vp}}\ (\mathbf{a}_{\mathtt{n,np}}\ \mathbf{burger}_{\mathtt{n}}))\ \mathbf{man}_{\mathtt{n}}))\ \rightsquigarrow$$

$$\mathbf{EVERY}_{\mathtt{n,vp,s}}(\mathbf{who}\ (\lambda x.\mathbf{A}_{\mathtt{n,vp,s}}\ \mathbf{burger}\ (\lambda y.\mathbf{eat}\ y_{\mathtt{np}}\ x_{\mathtt{np}}))\mathbf{man})\mathbf{die}$$

$$\mathbf{A}_{\mathtt{n,vp,s}}\ \mathbf{burger}(\lambda y.\mathbf{EVERY}_{\mathtt{n,vp,s}}(\mathbf{who}\ (\lambda x.\mathbf{eat}\ y_{\mathtt{np}}\ x_{\mathtt{np}})\mathbf{man})\mathbf{die})$$



Sentence coordination

For sentence coordination, the quantified NP raising algorithm avoids scope interaction across the sentences.

Every man sleeps and some woman worries

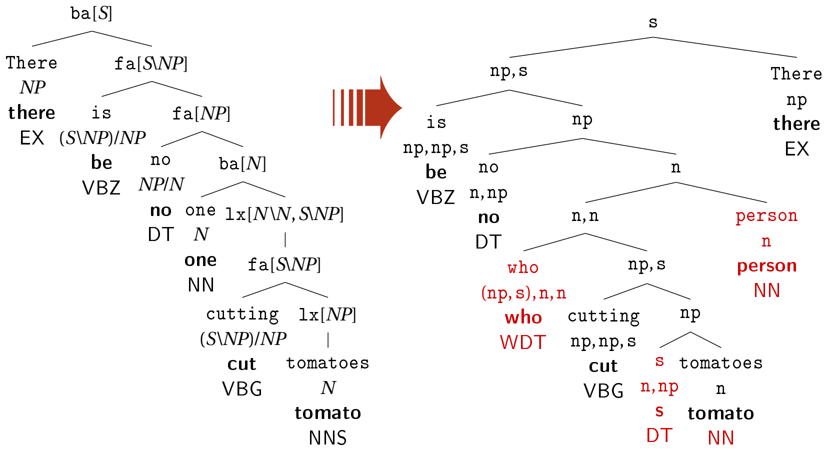
$\mathbf{and}_{\mathrm{s,s,s}}(\mathbf{sleep}_{\mathrm{vp}}(\mathbf{every}_{\mathrm{n,np}}\ \mathbf{man}_{\mathrm{n}}))(\mathbf{worry}_{\mathrm{vp}}(\mathbf{some}_{\mathrm{n,np}}\ \mathbf{woman}_{\mathrm{n}}))$

$\mathbf{and}\ (\mathbf{EVERY}_{\mathrm{n,vp,s}}\ \mathbf{man}\ \mathbf{sleep})(\mathbf{SOME}_{\mathrm{n,vp,s}}\ \mathbf{woman}\ \mathbf{worry})$

$\mathbf{SOME}_{\mathrm{n,vp,s}}\mathbf{woman}(\lambda y.\mathbf{EVERY}_{\mathrm{n,vp,s}}\mathbf{man}(\lambda x.\mathbf{and}\,(\mathbf{sleep}x)(\mathbf{worry}\ y)))$

$\mathbf{EVERY}_{\mathrm{n,vp,s}}\mathbf{man}(\lambda x.\mathbf{SOME}_{\mathrm{n,vp,s}}\mathbf{woman}(\lambda y.\mathbf{and}\,(\mathbf{sleep}x)(\mathbf{worry}\ y)))$

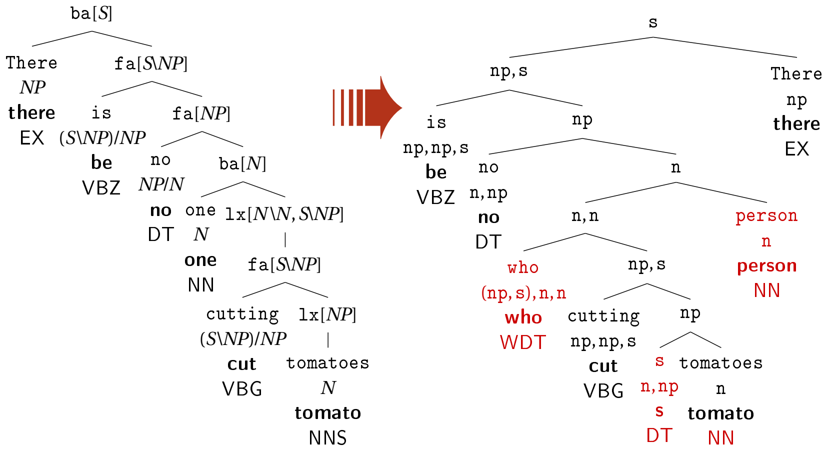
LLFgen: CCG tree → fixed CCG term → LLFs



`There is no one cutting tomatoes` ⇝

be(**no**(**who**(**cut**(**s** **tomato**))**person**))**there**

LLFgen: CCG tree → fixed CCG term → LLFs



be(no(who(cut(s tomato))person))there $\rightsquigarrow$

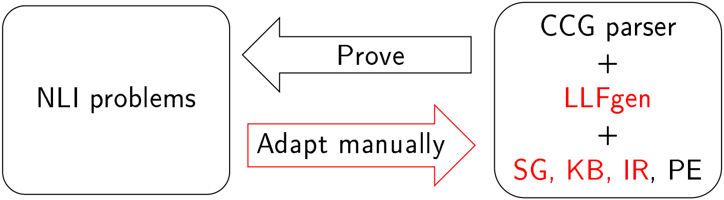
no (**who** $(\lambda x.$ **s tomato** $(\lambda y.$ **cut** $y\,x))$ **person**) $(\lambda z.$**be** z **there**)

s tomato $(\lambda y.$**no** (**who** (**cut** y) **person**) $(\lambda z.$**be** z **there**))

Linguistic Rules

Linguistic rules, in contrast to the algebraic rules, account for a certain syntactic constructions:

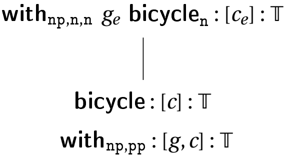
- Collected in a data-drive fashion using real LLFs;
- Remedy errors coming from syntactic parsers.



Rules for prepositions

PP@N$_{\mathbb{T}}$

$$\begin{array}{c} p^{\texttt{IN}}_{\mathrm{np,n,n}}\, d\, N : [c] : \mathbb{T} \\ \hline N : [c] : \mathbb{T} \\ p_{\mathrm{np,pp}} : [d, c] : \mathbb{T} \end{array}$$



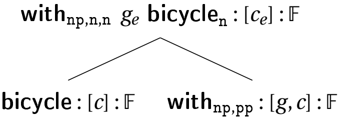
$\mathsf{with}_{\mathrm{np,n,n}}\ g_e\ \mathsf{bicycle}_{\mathrm{n}} : [c_e] : \mathbb{T}$

$\mathsf{bicycle} : [c] : \mathbb{T}$

$\mathsf{with}_{\mathrm{np,pp}} : [g, c] : \mathbb{T}$

PP@N$_{\mathbb{F}}$

$$\begin{array}{c|c} \multicolumn{2}{c}{p^{\texttt{IN}}_{\mathrm{np,n,n}}\, d\, N : [c] : \mathbb{F}} \\ \hline N : [c] : \mathbb{F} & p_{\mathrm{np,pp}} : [g, c] : \mathbb{F} \end{array}$$



$\mathsf{with}_{\mathrm{np,n,n}}\ g_e\ \mathsf{bicycle}_{\mathrm{n}} : [c_e] : \mathbb{F}$

$\mathsf{bicycle} : [c] : \mathbb{F}$ $\mathsf{with}_{\mathrm{np,pp}} : [g, c] : \mathbb{F}$

Problem of PP attachment

PP attachment is characterised with an attachment site and a nature of the attachment:

$$\text{John } [[\text{ate a roll}]_{VP/PP}\ [\text{with his hands}]_{PP}]_{VP} \qquad (1)$$

$$\text{John } [[\text{ate a roll}]_{VP}\ [\text{with Sam}]_{VP\backslash VP}]_{VP} \qquad (2)$$

$$\text{John ate a } [\text{roll}_{N}\ [\text{with eel}]_{N\backslash N}]_{N} \qquad (3)$$

$$\text{John ate a } [\text{roll}_{N/PP}\ [\text{of sushi}]_{PP}]_{N} \qquad (4)$$

Parsers can introduce errors in the PP attachments:

$$\text{John ate a } [[\text{roll}]_{N}\ [\text{with his hands}]_{N\backslash N}]_{N} \qquad (1a)$$

$$\text{John } [[\text{ate a roll}]_{VP/PP}\ [\text{with Sam}]_{PP}]_{VP} \qquad (2a)$$

$$\text{John } [[\text{ate a roll}]_{VP}\ [\text{with eel}]_{VP\backslash VP}]_{VP} \qquad (3a)$$

Treating a nature of PP attachment

SICK-9069 GOLD: `entailment` BY: C&C

Two boys are [[laying$_{VP}$ [in the ocean]$_{VP\backslash VP}$] [close to the beach]$_{VP\backslash VP}$]

Two boys are [[laying$_{VP/PP}$ [in the water]$_{PP}$] [close to the beach]$_{VP\backslash VP}$]

V@PP

$$V_{\mathrm{pp},\alpha}\,(p^{\mathtt{IN}}_{\mathrm{np,pp}}\,D) : [\vec{C}] : \mathbb{X}$$

$$p^{\mathtt{IN}}_{\mathrm{np},\alpha,\alpha}\,D\,V_{\alpha} : [\vec{C}] : \mathbb{X}$$

$$\alpha = (\mathrm{np}^{*}, \mathrm{vp})$$

$$\mathbf{lie}_{\mathtt{pp,vp}}\ (\mathbf{in}_{\mathtt{np,pp}}\ o_e) : [c] : \mathbb{F}$$

$$\mathbf{in}_{\mathtt{np,vp,vp}}\ o_e\ \mathbf{lie}_{\mathtt{vp}} : [c] : \mathbb{F}$$

SICK-340 GOLD: `entailment` BY: C&C

[schoolgirl$_{N/PP}$ [with a black bag]$_{PP}$] is on a crowded train

---

[girl$_{N}$ [with a black bag]$_{N\backslash N}$] is on a crowded train

N@PP$_{\mathbb{T}}$

$$N_{\mathrm{pp,n}}\ P_{\mathrm{pp}} : [c_e] : \mathbb{T}$$

$$N_{\mathrm{n}} : [c] : \mathbb{T}$$

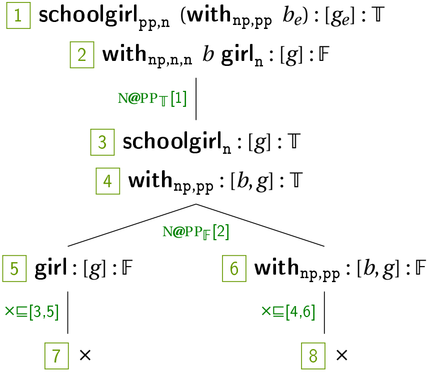
$$P : [c] : \mathbb{T}$$

N@PP$_{\mathbb{F}}$

$$N_{\mathrm{pp,n}}\ P_{\mathrm{pp}} : [c_e] : \mathbb{F}$$

$$N_{\mathrm{n}} : [c] : \mathbb{F} \qquad P : [c] : \mathbb{F}$$

Treatment case



$\boxed{1}$ $\textsf{schoolgirl}_{\mathrm{pp,n}}\ (\textsf{with}_{\mathrm{np,pp}}\ b_e) : [g_e] : \mathbb{T}$

$\boxed{2}$ $\textsf{with}_{\mathrm{np,n,n}}\ b\ \textsf{girl}_{\mathrm{n}} : [g] : \mathbb{F}$

$\mathrm{N@PP}_{\mathbb{T}}[1]$

$\boxed{3}$ $\textsf{schoolgirl}_{\mathrm{n}} : [g] : \mathbb{T}$

$\boxed{4}$ $\textsf{with}_{\mathrm{np,pp}} : [b, g] : \mathbb{T}$

$\mathrm{N@PP}_{\mathbb{F}}[2]$

$\boxed{5}$ $\textsf{girl} : [g] : \mathbb{F}$

$\times\sqsubseteq[3,5]$

$\boxed{7}$ $\times$

$\boxed{6}$ $\textsf{with}_{\mathrm{np,pp}} : [b, g] : \mathbb{F}$

$\times\sqsubseteq[4,6]$

$\boxed{8}$ $\times$

$\mathrm{N@PP}_{\mathbb{T}}$

$$\begin{array}{c} N_{\mathrm{pp,n}}\ P_{\mathrm{pp}} : [c_e] : \mathbb{T} \\ \hline N_{\mathrm{n}} : [c] : \mathbb{T} \\ P : [c] : \mathbb{T} \end{array}$$

$\mathrm{N@PP}_{\mathbb{F}}$

$$\begin{array}{c|c} \multicolumn{2}{c}{N_{\mathrm{pp,n}}\ P_{\mathrm{pp}} : [c_e] : \mathbb{F}} \\ \hline N_{\mathrm{n}} : [c] : \mathbb{F} & P : [c] : \mathbb{F} \end{array}$$

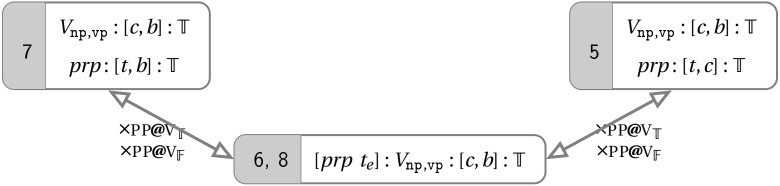
Treating a site of PP attachment

A boy saw a [criminal$_N$ [with a telescope]$_{N\backslash N}$]$_N$ (5)

A boy [[saw a criminal]$_{VP}$ [with a telescope]$_{VP\backslash VP}$]$_{VP}$ (6)

A criminal was seen by a [boy$_N$ [with a telescope]$_{N\backslash N}$]$_N$ (7)

A criminal [[was seen by a boy]$_{VP}$ [with a telescope]$_{VP\backslash VP}$]$_{VP}$ (8)



7

$V_{\mathrm{np,vp}} : [c, b] : \mathbb{T}$

$prp : [t, b] : \mathbb{T}$

5

$V_{\mathrm{np,vp}} : [c, b] : \mathbb{T}$

$prp : [t, c] : \mathbb{T}$

6, 8

$[prp\ t_e] : V_{\mathrm{np,vp}} : [c, b] : \mathbb{T}$

×PP@V$_{\mathbb{T}}$
×PP@V$_{\mathbb{F}}$

×PP@V$_{\mathbb{T}}$
×PP@V$_{\mathbb{F}}$

×PP@V$_{\mathbb{T}}$

$[prp\ t] : V : [\vec{A}] : \mathbb{T}$

$prp : [t, c] : \mathbb{F}$

×

$c \in \vec{A}$

×PP@V$_{\mathbb{F}}$

$[prp\ t] : V : [\vec{A}] : \mathbb{F}$

$V : [\vec{A}] : \mathbb{T}$

$prp : [t, c] : \mathbb{T}$

×

$c \in \vec{A}$

Lexical closure rules

Open compound nouns:

×CPN

$$\frac{\begin{array}{r} N_{\mathrm{n}} : [d] : \mathbb{T} \\ H_{\mathrm{pp,n}}(prp\ d) : [c] : \overline{\mathbb{X}} \\ A_{\mathrm{n,n}}\ H_{\mathrm{n}} : [c] : \mathbb{X} \end{array}}{\times}$$

$N \approx A$ or $N \approx_d A$

$$\frac{\begin{array}{r} \textbf{protection}_{\mathrm{n}} : [d_e] : \mathbb{T} \\ \textbf{gear}_{\mathrm{pp,n}}(\textbf{for}_{\mathrm{np,pp}}\ d_e) : [c_e] : \mathbb{F} \\ \textbf{protective}_{\mathrm{n,n}}\ \textbf{gear}_{\mathrm{n}} : [c_e] : \mathbb{T} \end{array}}{\times}\,(\times\mathrm{CPN}^*)$$

Light verb constructions:

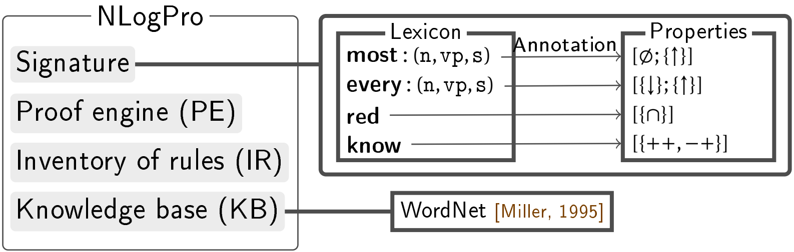
×LVC

$$\frac{\begin{array}{r} l_{\vec{\alpha},\mathrm{vp}} : [c, \vec{D}] : \mathbb{X} \\ u_{\mathrm{n}} : [c] : \mathbb{T} \\ v_{\vec{\alpha},\mathrm{s}} : [D] : \overline{\mathbb{X}} \end{array}}{\times}$$

$l \in \{$**do**, **get**, **give**, **have**, **make**, **take**$\}$,
$\vec{\alpha}$ is formed by `np` and `pp`, and $u \approx_d v$

$$\frac{\begin{array}{r} \textbf{do}_{\mathrm{np,vp}} : [d_e, h_e] : \mathbb{T} \\ \textbf{dance}_{\mathrm{n}} : [d_e] : \mathbb{T} \\ \textbf{dance}_{\mathrm{vp}} : [h_e] : \mathbb{F} \end{array}}{\times}\,(\times\mathrm{LVC}^*)$$

Natural logic theorem prover (NLogPro)



NLogPro

Signature

Proof engine (PE)

Inventory of rules (IR)

Knowledge base (KB)

Lexicon

most : $(\mathtt{n}, \mathtt{vp}, \mathtt{s})$

every : $(\mathtt{n}, \mathtt{vp}, \mathtt{s})$

red

know

Annotation

Properties

$[\emptyset; \{\uparrow\}]$

$[\{\downarrow\}; \{\uparrow\}]$

$[\{\cap\}]$

$[\{++, -+\}]$

WordNet [Miller, 1995]

KB uses 4 relations from WordNet 3.0:

- derivation
- similarity
- hyponymy/hypernymy
- antonymy

⚠ No word sense disambiguation system is used.

Some derivable rules

Derivable rules are shortcuts for several rule applications.

$\exists^{n}_{\mathbb{F}}$

$$\frac{\begin{array}{r} q_{\mathrm{n,vp,s}}\, N\, V : [\,] : \mathbb{F} \\ N : [c_e] : \mathbb{T} \end{array}}{V : [c] : \mathbb{F}}$$

$q \in \{\mathsf{a}, \mathsf{some}, \mathsf{the}, \mathsf{s}\}$

$\forall^{n}_{\mathbb{T}}$

$$\frac{\begin{array}{r} q_{\mathrm{n,vp,s}}\, N\, V : [\,] : \mathbb{T} \\ N : [c_e] : \mathbb{T} \end{array}}{V : [c] : \mathbb{T}}$$

$q \in \{\mathsf{every}, \mathsf{the}\}$

$\mathrm{NO}^{n}_{\mathbb{T}}$

$$\frac{\begin{array}{r} \mathbf{no}_{\mathrm{n,vp,s}}\, N\, V : [\,] : \mathbb{T} \\ N : [c_e] : \mathbb{T} \end{array}}{V : [c] : \mathbb{F}}$$

$\exists^{v}_{\mathbb{F}}$

$$\frac{\begin{array}{r} q_{\mathrm{n,vp,s}}\, N\, V : [\,] : \mathbb{F} \\ V : [c_e] : \mathbb{T} \end{array}}{N : [c] : \mathbb{F}}$$

$q \in \{\mathsf{a}, \mathsf{some}, \mathsf{the}, \mathsf{s}\}$

$\forall^{v}_{\mathbb{T}}$

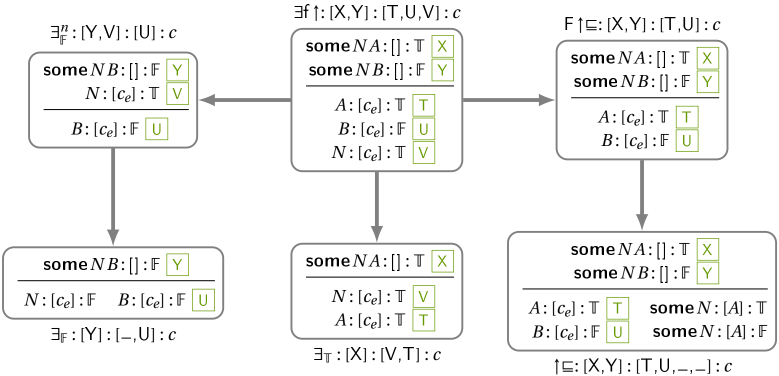
$$\frac{\begin{array}{r} q_{\mathrm{n,vp,s}}\, N\, V : [\,] : \mathbb{T} \\ V : [c_e] : \mathbb{F} \end{array}}{N : [c] : \mathbb{F}}$$

$q \in \{\mathsf{every}, \mathsf{the}\}$

$\mathrm{NO}^{v}_{\mathbb{T}}$

$$\frac{\begin{array}{r} \mathbf{no}_{\mathrm{n,vp,s}}\, N\, V : [\,] : \mathbb{T} \\ V : [c_e] : \mathbb{T} \end{array}}{N : [c] : \mathbb{F}}$$

Rule application subsumption



$\exists\mathsf{f}\!\uparrow : [\mathsf{X},\mathsf{Y}] : [\mathsf{T},\mathsf{U},\mathsf{V}] : c$

$$\begin{array}{rl} \textbf{some}\,N\,A : [\,] : \mathbb{T} & \boxed{\mathsf{X}} \\ \textbf{some}\,N\,B : [\,] : \mathbb{F} & \boxed{\mathsf{Y}} \\ \hline A : [c_e] : \mathbb{T} & \boxed{\mathsf{T}} \\ B : [c_e] : \mathbb{F} & \boxed{\mathsf{U}} \\ N : [c_e] : \mathbb{T} & \boxed{\mathsf{V}} \end{array}$$

$\exists^n_{\mathbb{F}} : [\mathsf{Y},\mathsf{V}] : [\mathsf{U}] : c$

$$\begin{array}{rl} \textbf{some}\,N\,B : [\,] : \mathbb{F} & \boxed{\mathsf{Y}} \\ N : [c_e] : \mathbb{T} & \boxed{\mathsf{V}} \\ \hline B : [c_e] : \mathbb{F} & \boxed{\mathsf{U}} \end{array}$$

$$\begin{array}{c} \textbf{some}\,N\,B : [\,] : \mathbb{F} \; \boxed{\mathsf{Y}} \\ \hline N : [c_e] : \mathbb{F} \quad B : [c_e] : \mathbb{F} \; \boxed{\mathsf{U}} \end{array}$$

$\exists_{\mathbb{F}} : [\mathsf{Y}] : [\_,\mathsf{U}] : c$

$$\begin{array}{rl} \textbf{some}\,N\,A : [\,] : \mathbb{T} & \boxed{\mathsf{X}} \\ \hline N : [c_e] : \mathbb{T} & \boxed{\mathsf{V}} \\ A : [c_e] : \mathbb{T} & \boxed{\mathsf{T}} \end{array}$$

$\exists_{\mathbb{T}} : [\mathsf{X}] : [\mathsf{V},\mathsf{T}] : c$

$\mathsf{F}\!\uparrow\!\sqsubseteq : [\mathsf{X},\mathsf{Y}] : [\mathsf{T},\mathsf{U}] : c$

$$\begin{array}{rl} \textbf{some}\,N\,A : [\,] : \mathbb{T} & \boxed{\mathsf{X}} \\ \textbf{some}\,N\,B : [\,] : \mathbb{F} & \boxed{\mathsf{Y}} \\ \hline A : [c_e] : \mathbb{T} & \boxed{\mathsf{T}} \\ B : [c_e] : \mathbb{F} & \boxed{\mathsf{U}} \end{array}$$

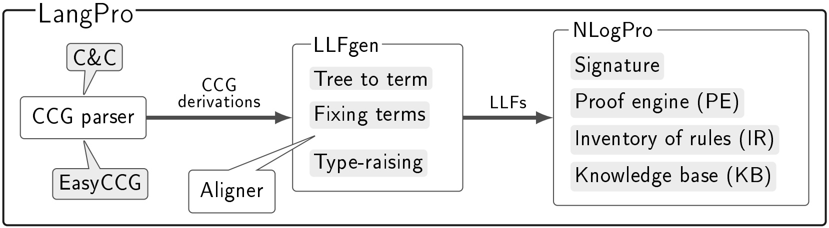
$$\begin{array}{rlr} \multicolumn{3}{c}{\textbf{some}\,N\,A : [\,] : \mathbb{T} \; \boxed{\mathsf{X}}} \\ \multicolumn{3}{c}{\textbf{some}\,N\,B : [\,] : \mathbb{F} \; \boxed{\mathsf{Y}}} \\ \hline A : [c_e] : \mathbb{T} & \boxed{\mathsf{T}} & \textbf{some}\,N : [A] : \mathbb{T} \\ B : [c_e] : \mathbb{F} & \boxed{\mathsf{U}} & \textbf{some}\,N : [A] : \mathbb{F} \end{array}$$

$\uparrow\!\sqsubseteq : [\mathsf{X},\mathsf{Y}] : [\mathsf{T},\mathsf{U},\_,\_] : c$

$$\exists^n_{\mathbb{F}} : [\mathsf{Y},\mathsf{V}] : [\mathsf{U}] : c \Rightarrow \exists_{\mathbb{F}} : [\mathsf{Y}] : [\_,\mathsf{U}] : c$$

Natural language theorem prover (LangPro)

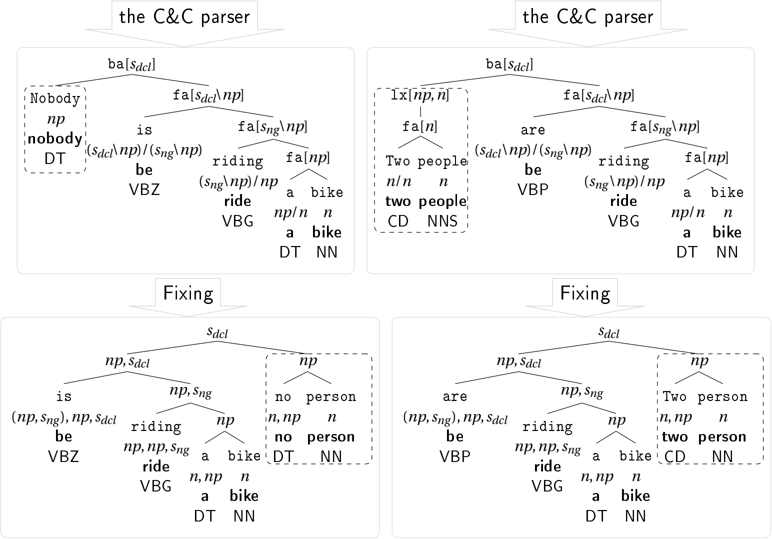
Chaining a CCG parser, the LLF generator and NLogPro results in a theorem prover for natural language.



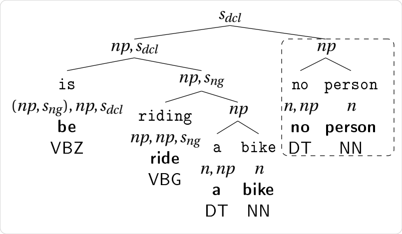
Online demo at: http://naturallogic.pro/LangPro

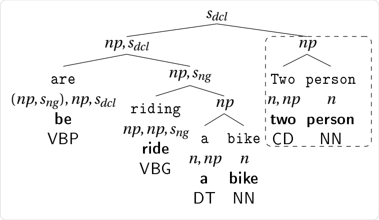
LangPro in action

SICK-2865: `Nobody is riding a bike` $\overset{?}{\Rightarrow}$ `Two people are riding a bike`



LangPro in action (2)





Type-raising

Type-raising

no person (**be** ($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** y x)))
a bike ($\lambda x.$ **no person** (**be** (**ride** x)))

two person (**be** ($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** y x)))
a bike ($\lambda x.$ **two person** (**be** (**ride** x)))

Proving by PE using IR & KB

`intial nodes for entailment checking:`
no person (**be** ($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** y x))) $: [\,] : \mathbb{T}$
two person (**be** ($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** y x))) $: [\,] : \mathbb{F}$

`intial nodes for contradiction checking:`
no person (**be** ($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** y x))) $: [\,] : \mathbb{T}$
two person (**be** ($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** y x))) $: [\,] : \mathbb{T}$

LangPro in action (3)

$\boxed{1}$ **no person** (**be**($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** $y\ x$))): $[\,]:\mathbb{T}$

$\boxed{2}$ **two person** (**be** ($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** $y\ x$))): $[\,]:\mathbb{T}$

$\exists_{\mathbb{T}}[2]$

$\boxed{3}$ **person**: $[c]:\mathbb{T}$

$\boxed{4}$ **be**($\lambda x.$ (**a bike**) ($\lambda y.$ **ride** $y\ x$)): $[c]:\mathbb{T}$

$\text{no}^{n}_{\mathbb{T}}[1{,}4]$

$\boxed{5}$ **person**: $[c]:\mathbb{F}$

$\boxed{6}$ ×

$$\frac{\textbf{no}\ A\ B:[\,]:\mathbb{T} \qquad A:[c]:\mathbb{T}}{B:[c]:\mathbb{F}}\ \text{no}^{n}_{\mathbb{T}}$$

$$\frac{N^{\mathtt{CD}}\ A\ B:[\,]:\mathbb{T}}{A:[c]:\mathbb{T} \qquad B:[c]:\mathbb{T}}\ \exists_{\mathbb{T}}$$

The SICK dataset

SICK [Marelli et al., 2014b] contains Sentences Involving Compositional Knowledge:

- 10K Text-Hypothesis pairs with E, C, & N labels.
- Contains no encyclopedic knowledge, no named entities, relatively small vocabulary, less MWE and no lengthy sentences (9 words per sentence).
- Contradictions (86%) rely too much on negative words and antonyms [Lai and Hockenmaier, 2014].
- A benchmark for the SemEval-14 RTE task [Marelli et al., 2014a]: Trial (5%), Train (45%), and test (50%).
- 84% of crowd workers' labels match the gold labels.

The FraCaS dataset

The FraCaS test suite [Cooper et al., 1996] was an early attempt to creating a semantic benchmark for NLP systems.

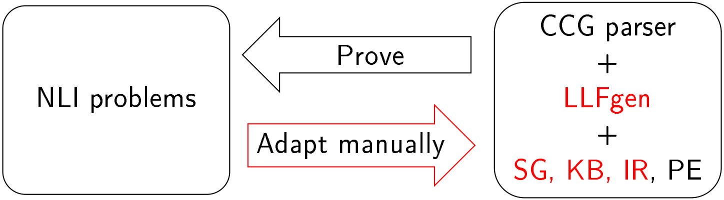
- 346 problems (45% multi-premised).
- Covers GQs, plurals, anaphora, ellipsis, adjectives, comparatives, temporal reference, verbs and attitudes.
- Three-way annotated by the authors of the dataset.
- Has some ambiguous sentences and erroneous problems.
- Requires almost no lexical or world knowledge

Later, the FraCaS question-answer pairs where converted into an NLI format [MacCartney and Manning, 2007].

Learning phase

The prover LangPro is (semi-automatically) trained on the NLI datasets [Abzianidze, 2016a].

- **Adaptation**:



NLI problems

Prove

Adapt manually

CCG parser
+
LLFgen
+
SG, KB, IR, PE

  Used datasets: SICK-trial and FraCaS

- **Development**:
  Finding optimal values for certain parameters of the prover.

NB: Only C&C parser is used in the learning phase in order to test LangPro for an unseen parser, EasyCCG, later.

Adaptation: negative cases

Avoid fitting the data by adopting unsound/non-general solutions.

The problems that were not solved during the adaptation:

- Sentence with non-S or no categories;
- The error is analysis is too specific to fix:

At	most	ten	commissioners	spend	time	at	home
$(S/S)/NP$	N/N	N/N	N	$(VP/PP)/NP$	N	PP/NP	N

- Lexical relation is context dependent:

SICK-4505 GOLD: `entailment`

The doctors are healing a man

The doctor is helping the patient

SICK-384 GOLD: `entailment`

A white and tan dog is running through the tall and green grass

A white and tan dog is running through a field

Adaptation: positive cases

The problems that were solved by upgrading one of the components of the prover:

- Treat **few** as ↓ in its 1st arg (*absolute* reading):

  FraCaS-76 GOLD: `entailment`

  Few committee members are from southern Europe

  Few female committee members are from southern Europe

- Introduce **fit** $\sqsubseteq$ **apply** and **food** $\sqsubseteq$ **meal**:

  SICK-4734 GOLD: `entailment`

  A man is fitting a silencer to a pistol

  A man is applying a silencer to a gun

  SICK-5110 GOLD: `entailment`

  A chef is preparing some food

  A chef is preparing a meal

Development phase

Optimal values of the following parameters are searched:

- The number of word senses to consider;
- The upper bound for the number of rule applications;
- Use a term aligner:
    - Weak aligner aligns everything but terms of type `np`:

      SICK-727 GOLD: `contradiction`

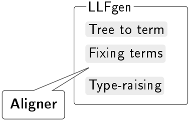
      The man in a grey t-shirt is sitting on a rock in front of the waterfall

      There is no man in a grey t-shirt sitting on a rock in front of the waterfall
    - Strong aligner aligns everything but terms of type `np` with ↓arg.

      SICK-423 GOLD: `contradiction`

      Two men are not holding fishing poles

      Two men are holding fishing poles
- Efficiency criterion of tableau rules.



Efficiency criterion

Tableau rules have the following properties:

- Non-branching or branching (so called, α or β rules);
- Semantic equivalence vs proper entailment;
- Consuming (so called, γ rule) vs non-consuming;
- Producing (so called, δ rule) vs non-producing.

An example of an efficiency criterion:

$$EC = \langle \texttt{nonBr}, \ \texttt{semEqui}, \ \texttt{nonConsum}, \ \texttt{nonProd} \rangle$$

An efficiency vectors based on the EC efficiency criterion:

- $V_{EC}(\wedge_{\mathbb{T}}) = 1111$
- $V_{EC}(\vee_{\mathbb{T}}) = 0111$
- $V_{EC}(\exists_{\mathbb{T}}) = 1110$
- $V_{EC}(\exists_{\mathbb{F}}) = 0001$

What is the optimal efficiency criterion?

Greedy search for optimal parameters

Acc%	Prec%	Rec%	Sense	Efficiency criterion	Aligner	RAL	Parser
75.09	98.5	43.6	1	`[nonP,nonB,equi,nonC]`	No	200	C&C
76.42	98.3	46.8	1-5	-	-	-	-
76.89	97.8	48.1	All	-	-	-	-
78.44	97.9	51.7	-	`[equi,nonB,nonP,nonC]`	-	-	-
79.33	97.9	53.8	-	-	Weak	-	-
81.5	97.7	59.0	-	-	Strong	-	-
81.53	97.7	59.1	-	-	Strong	400	-
81.38	98.0	58.5	-	-	Strong	400	EasyCCG
82.6	97.7	61.6	-	-	Strong	400	Both

The results are given on the SICK-train problems.

FraCaS-21 GOLD: `entailment`

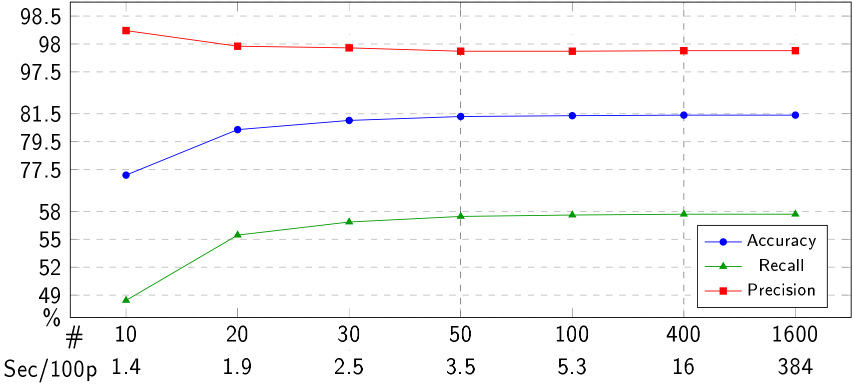
The residents of member states have the right to live in Europe

All residents of member states are individuals

Every individual who has the right to live in Europe can travel freely within Europe

The residents of member states can travel freely within Europe

Efficient and optimal rule application numbers



The results are given on the SICK-train problems.

Solving FraCaS [Abzianidze, 2016b]

LangPro with C&C

Gold**ccLP**	yes	no	unk
yes	**51**	0	19 + 4
no	1	**14**	2
unk	1	0	**44 + 6**

P = .97, R = .71, Acc = .81

+

LangPro with EasyCCG

Gold**easyLP**	yes	no	unk
yes	**52**	0	22
no	1	**12**	4
unk	2	0	**49**

P = .96, R = .70, Acc = .80

=

LangPro

Gold**LP**	yes	no	unk
yes	**60**	0	14
no	1	**14**	2
unk	2	0	**49**

P = **.96**, R = **.81**, Acc = **.87**

FraCaS-109 GOLD: `contradiction` LP: `entailment`

Just one accountant attended the meeting

Some accountants attended the meeting

Related work (FraCaS)

[MacCartney and Manning, 2008] and [Angeli and Manning, 2014] employ a natural logic that is driven by sentence edits.

[Lewis and Steedman, 2013] employ Boxer-style [Bos et al., 2004] translation into FOL, Prover9 and distributional relation clustering.

[Mineshima et al., 2015] also uses the Boxer-style translation but some HOGQs are treated as higher-order terms. Their inference system is implemented in the proof assistant Coq.

[Tian et al., 2014] and [Dong et al., 2014] uses abstract denotations obtained from DCS trees [Liang et al., 2011].

[Bernardy and Chatzikyriakidis, 2017] uses Grammatical Framework and Coq. They use gold standard GF trees.

Comparison on FraCaS

Sec (Sing/All)	Single-premised (Acc %)							Overall (Acc %)				
	BL	NL07,08	LS P/G	NLI	T14a,b	M15	**LP**	BL	LS P/G	T14a,b	M15	**LP**
1 GQs (44/74)	45	84 **98**	70 89	95	80 93	82	93	50	62 85	80 **95**	78	**95**
2 Plur (24/33)	58	~~42~~ **75**	-	~~38~~	-	67	**75**	61	-	-	67	**73**
5 Adj (15/22)	40	60 80	-	**87**	-	**87**	**87**	41	-	-	68	**77**
9 Att (9/13)	67	~~56~~ 89	-	~~22~~	-	78	**100**	62	-	-	77	**92**
1,2,5,9 (92/142)	50	- **88**	-	-	-	78	**88**	52	-	-	74	**87**

NL07 [MacCartney and Manning, 2007], **NL08** [MacCartney and Manning, 2008], **NLI** [Angeli and Manning, 2014], **LS** [Lewis and Steedman, 2013],
M15 [Mineshima et al., 2015], **T14a** [Tian et al., 2014] and **T14b** [Dong et al., 2014]

Advantages of our approach over the related ones include:

- Reasoning (with the semantic tableau) over multiple-premises;
- Logical forms close to surface forms;
- Underlying expressive high-order logic.

Curing SICK [Abzianidze, 2015]

Gold SICK-test \ LangPro	Ent	Cont	Neut
Entailment	**805**	0	609
Contradiction	2	**482**	236
Neutral	26	7	**2760**

P=**97.4%**, R=**60.3%**, Acc=**82.14%**

Mainly the usage of WordNet and noisy gold labels are blamed for false proofs.

ID G/LP	Premise	Conclusion
1405 N/E	A **prawn** is being cut by a woman	A woman is cutting **shrimps**
4443 N/E	A man is singing to a **girl**	A man is singing to a **woman**
2870 N/C	Two people are riding a **motorcycle**	Nobody is riding a **bike**
8913 N/C	A couple is not looking at a map	A couple is looking at a map
363 C/C	P: A soccer ball is not rolling into a goal net C: A soccer ball is rolling into a goal net	

False neutrals

Reason for false neutrals are knowledge sparsity (ca 50%), a lack of rules (ca 25%), wrong labels and parsing mistakes.

ID	G/LP	Premise	Conclusion
4974	E/N	Someone is holding a hedgehog	Someone is holding a small animal
6258	E/N	P: A policeman is sitting on a **motorcycle**	C: The cop is sitting on a police **bike**
4553	E/N	P: A man is emptying a container made of plastic	C: A man is emptying a plastic container
4720	E/N	A monkey is practicing martial arts	A chimp is practicing martial arts
6447	C/N	P: [A small boy [in a yellow shirt]] is laughing on the beach	C: There is no small boy [in a yellow shirt [laughing on the beach]]

Comparison on SICK

SemEval-14 systems	Prec%	Rec%	Acc%	(+LP)	NWS%
Baseline (majority)	-	-	56.69		39.7
Illinois-LH	81.56	**81.87**	**84.57**	(+0.65)	72.8
ECNU	84.37	74.37	83.64	(+1.77)	72.7
UNAL-NLP	81.99	76.80	83.05	(+1.48)	71.2
SemantiKLUE	85.40	69.63	82.32	(+2.84)	71.5
The Meaning Factory	93.63	60.64	81.59	(+2.78)	73.0
UTexas (Prob-FOL)	**97.87**	38.71	73.23	(+9.44)	62.5
LangPro	**97.35**	60.31	82.14		**74.8**

RTE systems	Acc%
Prob-FOL	76.52
Prob-FOL*+Rules	**85.10**
Nutcracker+PPDB	79.60
LSTM RNN+SNLI	80.80
ABCNN-3	**86.20**
BERT/RoBERTa	? 90

"Hard" problems

The problems from SICK-test that were proved correctly by both ccLangPro and easyLangPro but failed by all the top five systems at the SemEval-14 task.

ID	G	Text	Hypothesis
247	C	T: The woman is not wearing glasses or a headdress H: A woman is wearing an Egyptian headdress	
406	E	T: A group of scouts are hiking through the grass H: People are walking	
2895	C	The man isn't lifting weights	The man is lifting barbells
3527	E	T: A person is jotting something with a pencil H: A person is writing	
3570	C	The piece of paper is not being cut	Paper is being cut with scissors
3608	N	T: A monkey is riding a bike H: A bike is being ridden over a monkey	
3806	E	A man in a hat is playing a harp	A man is playing an instrument
4479	E	The boy is playing the piano	The boy is playing a musical instrument

Conclusion

Natural Tableau is a wide-coverage but still logic-based reasoning system inspired by Natural Logic.

It represents a proof-theoretic approach to NLI.

Pros and cons of Natural Tableau:

- ✓ Employs higher-order logic to model linguistic semantics;
- ✓ Allows deep logical and shallow (e.g. monotonicity) reasoning;
- ✓ Getting logical form is similar to syntactic parsing;
- ✗ Heavily hinges on CCG parsing;
- ✗ Heavily hinges on knowledge;
- ✓ Proofs are highly reliable ($\leq 3\%$ false proofs);
- ✗ Suffers from multi-sense words;
- ✗ Suffers from multi-sense words and out0of-context knowledge;
- ✓ Possible to train on the data;
- ✓ Its decision procedure is transparent and explanatory;

Future work

There are really many directions for future work:

- Explore different types of RTE data, e.g., the newswire or human generated data [Bowman et al., 2015];
- Incorporate more knowledge in KB, e.g., paraphrase database [Ganitkevitch et al., 2013].
- Model different phenomena: comparatives, anaphora, cardinals, etc.
- Pairing with distributional semantics: $R(w_1, w_2, r)$ and weighted closure branches;
- Improve knowledge induction with abductive reasoning;
- Generate LLFs from Universal Dependency trees

  \+ the Universal Semantic Tagging [Abzianidze and Bos, 2017]

  $\rightarrow^{?}$ Multilingual Natural Tableau

References I

Abzianidze, L. (2015). A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502, Lisbon, Portugal. Association for Computational Linguistics.

Abzianidze, L. (2016a). *A natural proof system for natural language*. PhD thesis, Tilburg University.

Abzianidze, L. (2016b). Natural solution to fracas entailment problems. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 64–74, Berlin, Germany. Association for Computational Linguistics.

Abzianidze, L. and Bos, J. (2017). Towards universal semantic tagging. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Ajdukiewicz, K. (1935). Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27. English translation "Syntactic Connexion" by H. Weber in McCall, S. (Ed.) *Polish Logic*, pp. 207–231, Oxford University Press, Oxford, 1967.

Angeli, G. and Manning, C. D. (2014). Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bernardy, J.-P. and Chatzikyriakidis, S. (2017). A type-theoretical system for the FraCaS test suite: Grammatical framework meets coq. In *IWCS 2017 - 12th International Conference on Computational Semantics - Long papers*.

References II

Beth, E. W. (1955). Semantic Entailment and Formal Derivability. *Koninklijke Nederlandse Akademie van Wentenschappen, Proceedings of the Section of Sciences*, 18:309–342.

Bos, J., Clark, S., Steedman, M., Curran, J. R., and Hockenmaier, J. (2004). Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, pages 1240–1246, Geneva, Switzerland.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Church, A. (1940). A formulation of the simple theory of types. *Jurnal of Symbolic Logic*, 5(2):56–68.

Clark, S. and Curran, J. R. (2007). Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistics*, 33.

Cooper, R., Crouch, D., Eijck, J. V., Fox, C., Genabith, J. V., Jaspars, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M., Pulman, S., Briscoe, T., Maier, H., and Konrad, K. (1996). *FraCaS: A Framework for Computational Semantics*. Deliverable D16.

Dong, Y., Tian, R., and Miyao, Y. (2014). Encoding generalized quantifiers in dependency-based compositional semantics. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 585–594, Phuket,Thailand. Department of Linguistics, Chulalongkorn University.

References III

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Hillel, Y. B. (1953). A Quasi-Arithmetical Notation for Syntactic Description. *Language*, 29(1):47–58.

Hintikka, J. (1955). *Two Papers on Symbolic Logic: Form and Content in Quantification Theory and Reductions in the Theory of Types*. Number 8 in Acta philosophica Fennica. Societas Philosophica.

Lai, A. and Hockenmaier, J. (2014). Illinois-lh: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 329–334, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Lakoff, G. (1970). Linguistics and natural logic. In Davidson, D. and Harman, G., editors, *Semantics of Natural Language*, volume 40 of *Synthese Library*, pages 545–665. Springer Netherlands.

Lewis, M. and Steedman, M. (2013). Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics (TACL)*, 1:179–192.

Lewis, M. and Steedman, M. (2014). A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar. Association for Computational Linguistics.

References IV

Liang, P., Jordan, M. I., and Klein, D. (2011). Learning dependency-based compositional semantics. In *Association for Computational Linguistics (ACL)*, pages 590–599.

MacCartney, B. and Manning, C. D. (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 193–200, Stroudsburg, PA, USA. Association for Computational Linguistics.

MacCartney, B. and Manning, C. D. (2008). Modeling semantic containment and exclusion in natural language inference. In Scott, D. and Uszkoreit, H., editors, *COLING*, pages 521–528.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014a). Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of SemEval 2014 (International Workshop on Semantic Evaluation)*, pages 1–8, East Stroudsburg PA. ACL.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014b). A sick cure for the evaluation of compositional distributional semantic models. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

References V

Mineshima, K., Martínez-Gómez, P., Miyao, Y., and Bekki, D. (2015). Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal. Association for Computational Linguistics.

Moss, L. S. (2010). Natural logic and semantics. In Aloni, M., Bastiaanse, H., de Jager, T., and Schulz, K., editors, *Logic, Language and Meaning: 17th Amsterdam Colloquium, Amsterdam, The Netherlands, December 16-18, 2009, Revised Selected Papers*, pages 84–93. Springer Berlin Heidelberg, Berlin, Heidelberg.

Muskens, R. (2010). An analytic tableau system for natural logic. In Aloni, M., Bastiaanse, H., de Jager, T., and Schulz, K., editors, *Logic, Language and Meaning*, volume 6042 of *Lecture Notes in Computer Science*, pages 104–113. Springer Berlin Heidelberg.

Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Tian, R., Miyao, Y., and Matsuzaki, T. (2014). Logical inference on dependency-based compositional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 79–89, Baltimore, Maryland. Association for Computational Linguistics.

van Benthem, J. (2008). A brief history of natural logic. In *Technical Report PP-2008-05*. Institute for Logic, Language & Computation.