LCGbank

A Corpus of Syntactic Analyses Based on Proof Nets

Aditya Bhargava, Timothy A. D. Fowler, and Gerald Penn

LREC-COLING 2024

Parsing with categorial grammars

- Many related members of the categorial grammar family
  - Ajdukiewicz–Bar-Hillel grammar (ABG)
  - Combinatory categorial grammar (CCG)
  - Lambek categorial grammar (LCG)
  - Type-logical grammar (TLG)
- CCG has received (by far) the most attention in CL research
  - Many statistical parsers
  - Corpora in multiple languages

LCG *versus* CCG

- Generative capacity
  - LCG is weakly context-free
  - CCG is context-sensitive
- Computational complexity
  - LCG parsing is NP-complete
  - CCG has known polynomial-time algorithms

LCG *versus* CCG

- Generative capacity
  - LCG is weakly context-free, but this is rarely a problem
  - CCG is context-sensitive
- Computational complexity
  - LCG parsing is NP-complete
  - CCG has known polynomial-time algorithms

LCG *versus* CCG

- Generative capacity
  - LCG is weakly context-free, but this is rarely a problem, especially for statistical parsers
  - CCG is context-sensitive
- Computational complexity
  - LCG parsing is NP-complete
  - CCG has known polynomial-time algorithms

LCG *versus* CCG

- Generative capacity
  - LCG is weakly context-free, but this is rarely a problem, especially for statistical parsers
  - CCG ~~is~~ can be context-sensitive
- Computational complexity
  - LCG parsing is NP-complete
  - CCG has known polynomial-time algorithms

LCG *versus* CCG

- Generative capacity
  - LCG is weakly context-free, but this is rarely a problem, especially for statistical parsers
  - CCG ~~is~~ can be context-sensitive, but in practice never is
- Computational complexity
  - LCG parsing is NP-complete
  - CCG has known polynomial-time algorithms

LCG *versus* CCG

- Generative capacity
  - LCG is weakly context-free, but this is rarely a problem, especially for statistical parsers
  - CCG ~~is~~ can be context-sensitive, but in practice never is
- Computational complexity
  - LCG parsing is NP-complete, but polynomial assuming reasonable bounds on categories
  - CCG has known polynomial-time algorithms

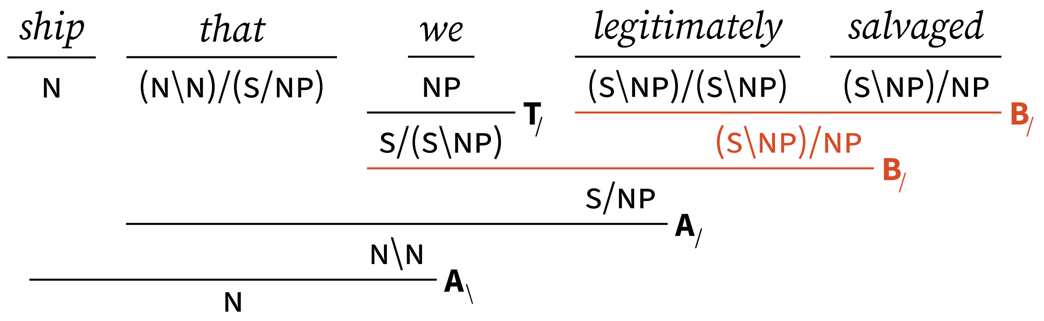
LCG *versus* CCG

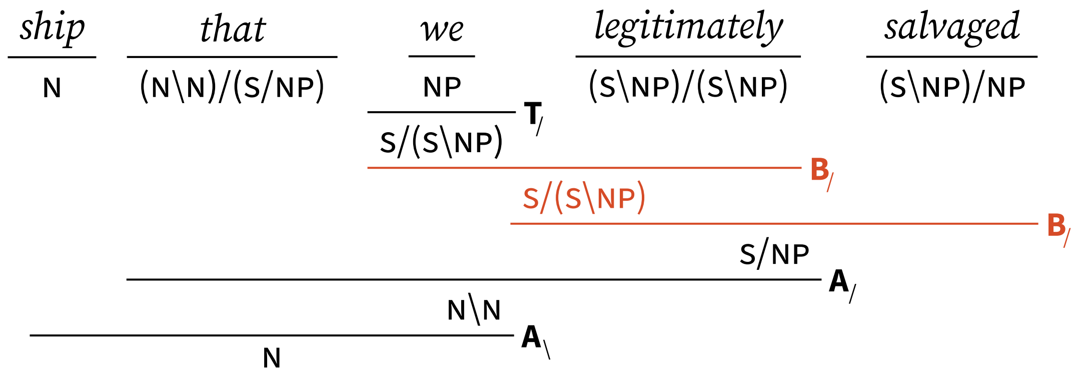
- Generative capacity
  - LCG is weakly context-free, but this is rarely a problem, especially for statistical parsers
  - CCG ~~is~~ can be context-sensitive, but in practice never is
- Computational complexity
  - LCG parsing is NP-complete, but polynomial assuming reasonable bounds on categories
  - CCG has known polynomial-time algorithms
  - Not so relevant to statistical parsers

Why LCG?

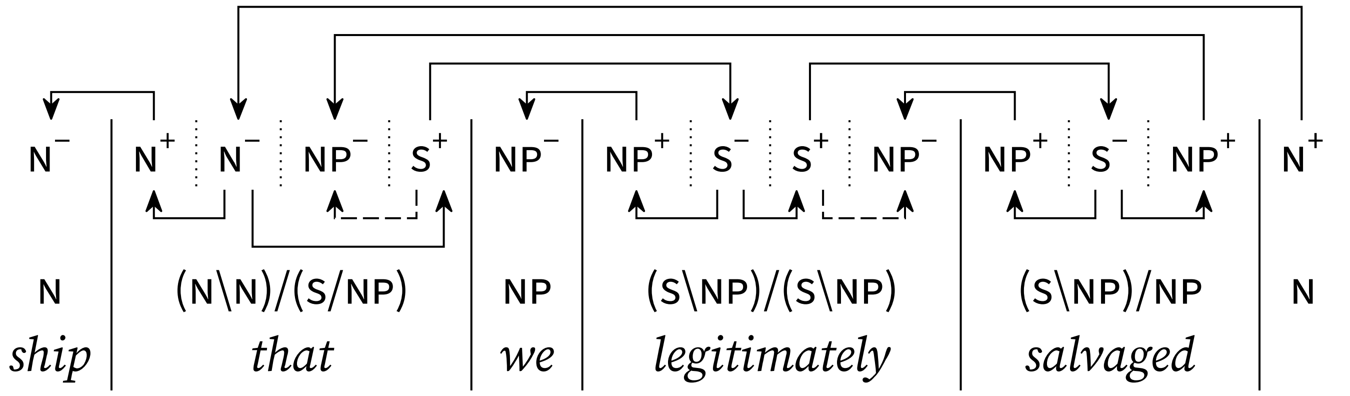
Why LCG?
Proof nets!

Spurious ambiguities





Spurious ambiguities



Spurious ambiguities

- CCG parsers use normal-form constraints during parsing
- Proof nets represent LCG derivations such that semantically-equivalent derivations correspond to the same proof net
- CCG is incompatible

Statistical parsing with proof nets

- Recent interest in statistical parsing with proof nets
  - Involves a more structured treatment of lexical categories
- Can even train *without* ground-truth derivations

Statistical parsing with proof nets

- Recent interest in statistical parsing with proof nets
  - Involves a more structured treatment of lexical categories
- Can even train *without* ground-truth derivations

…but very few corpora, and none in English!

LCGbank

LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

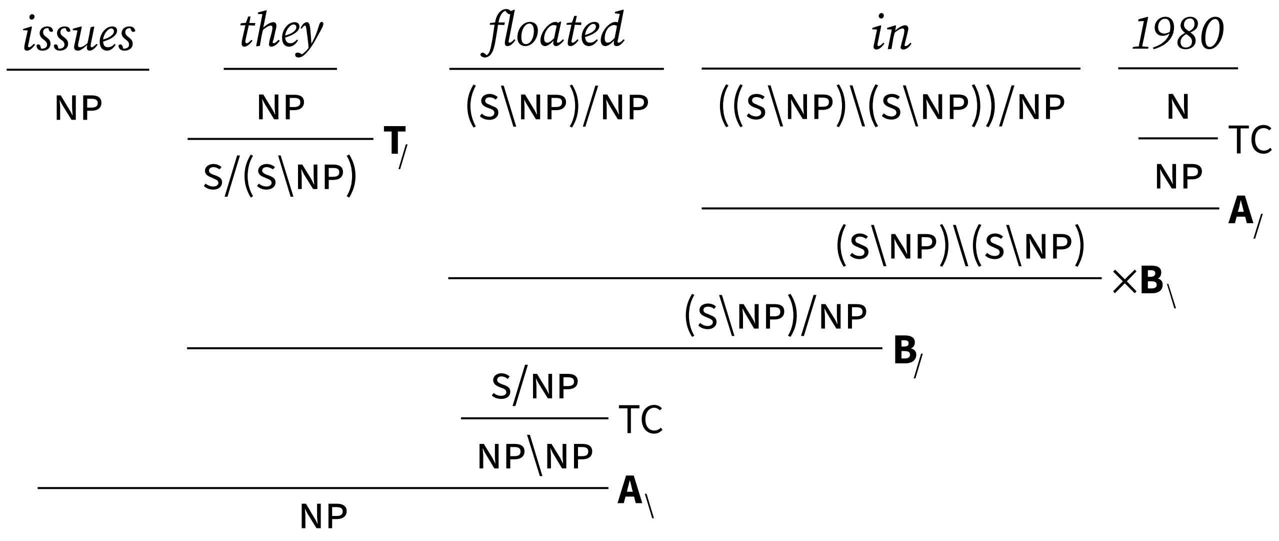
LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

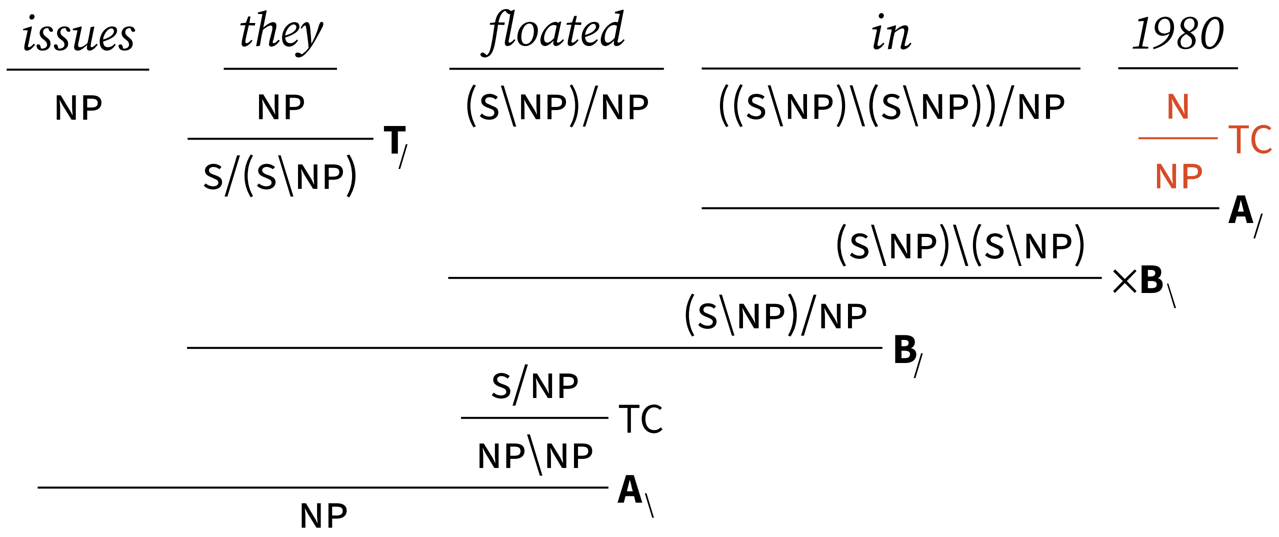
LCGbank development process

- Start from CCGbank
- Discard category features
- Lexicalize incompatible CCG rules for LCG compatibility
- Lexicalize incompatible CCG*bank* rules (e.g., type-changing)
- Converted (LCG) derivations then specify the proof net
- New derivations for left-out sentences
- Result: fully compositional semantics

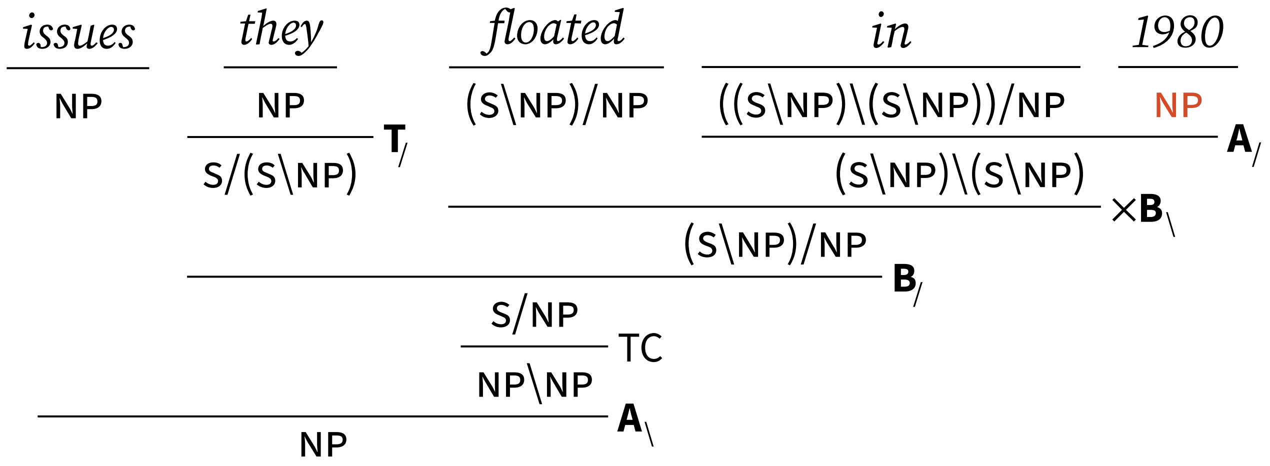
CCGbank's type-changing



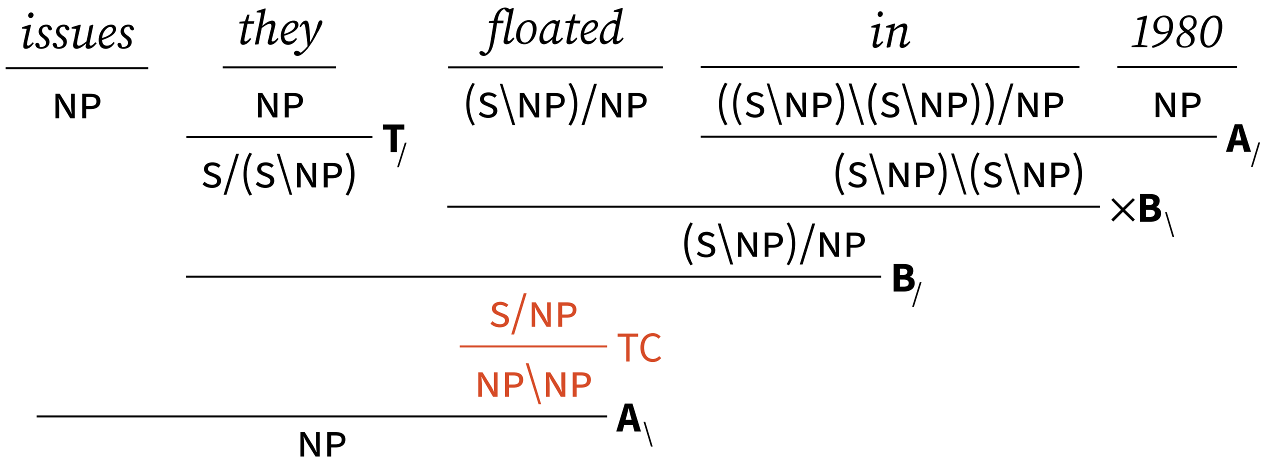
CCGbank's type-changing



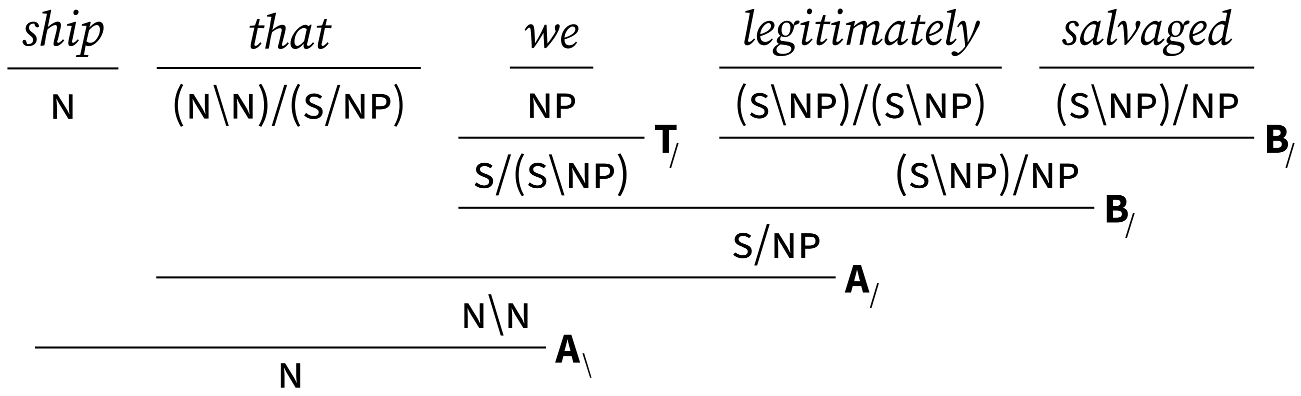
CCGbank's type-changing



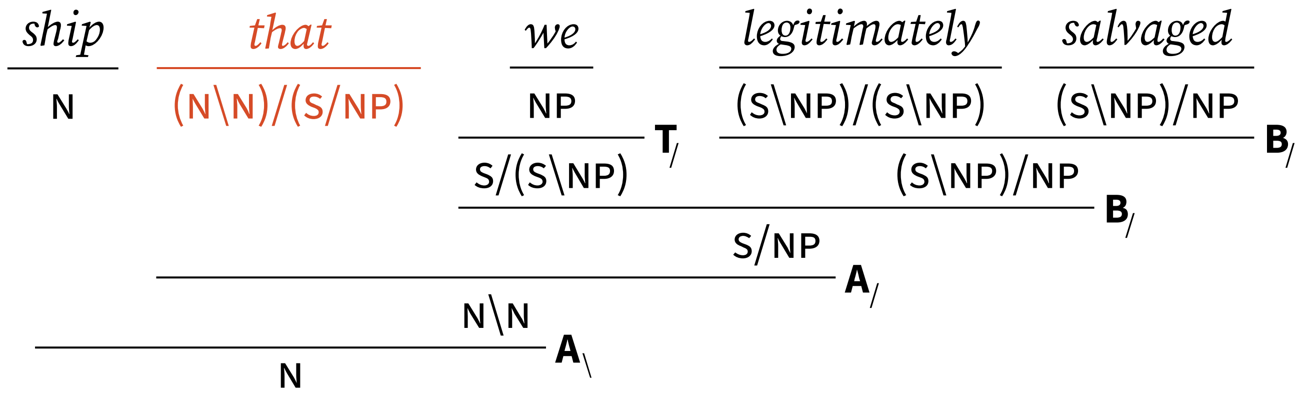
CCGbank's type-changing



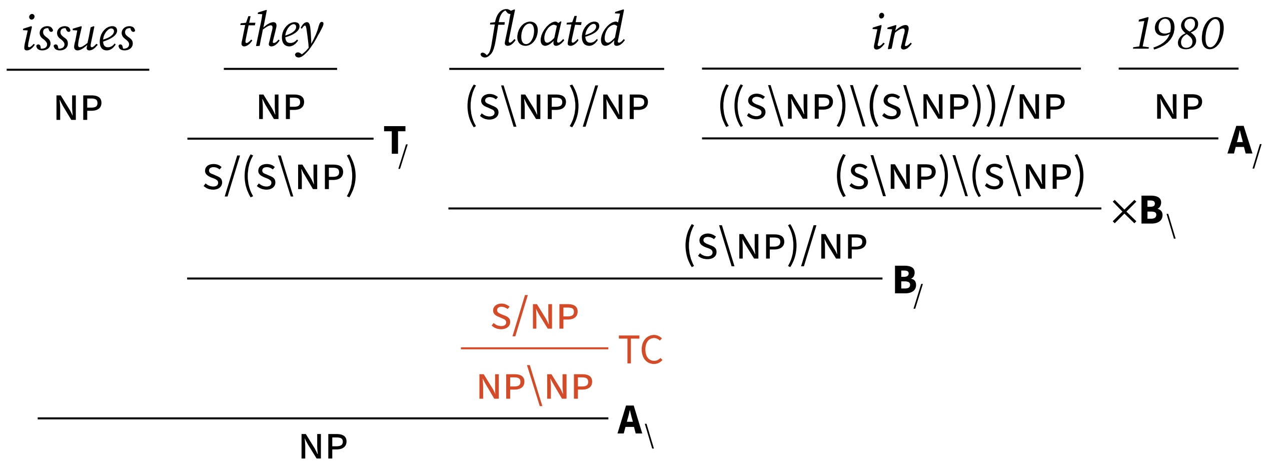
CCGbank's type-changing



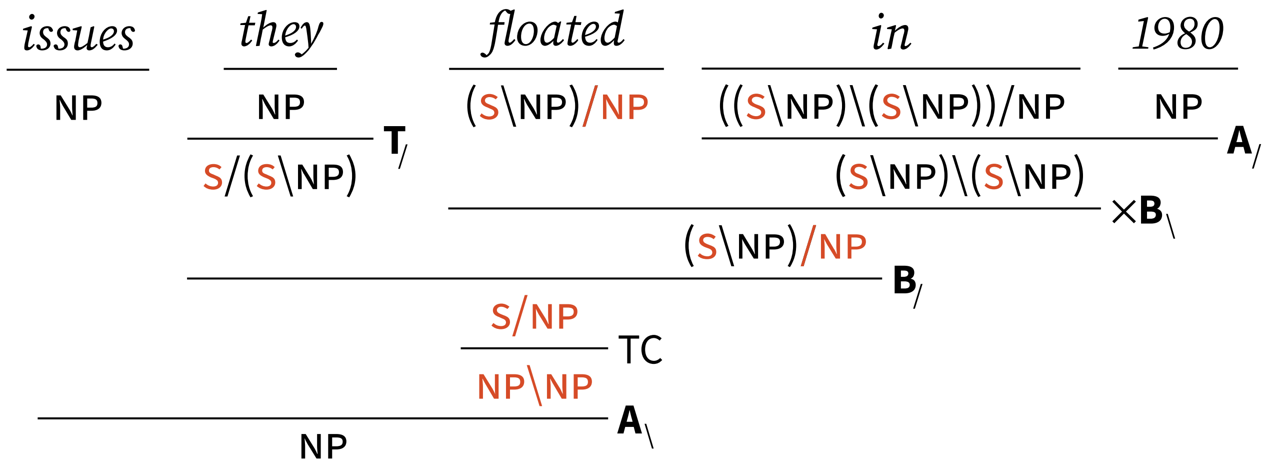
CCGbank’s type-changing



CCGbank's type-changing

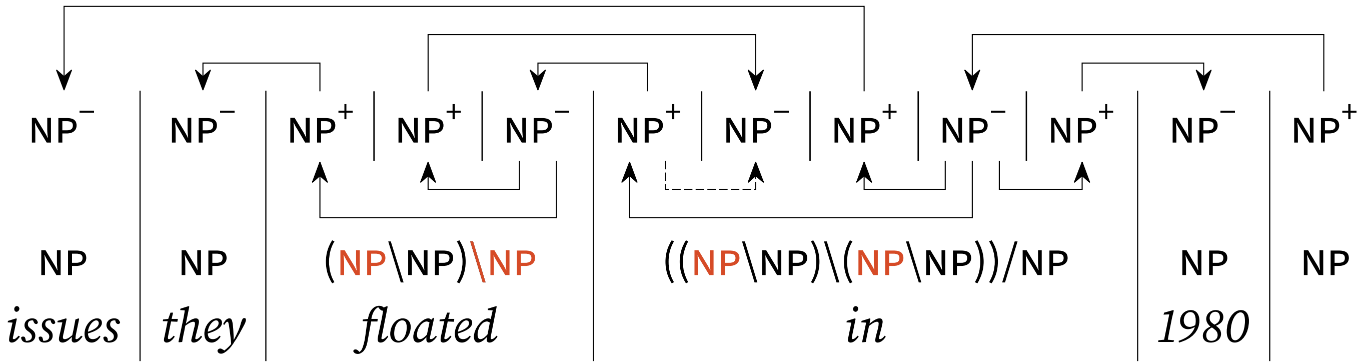


CCGbank's type-changing

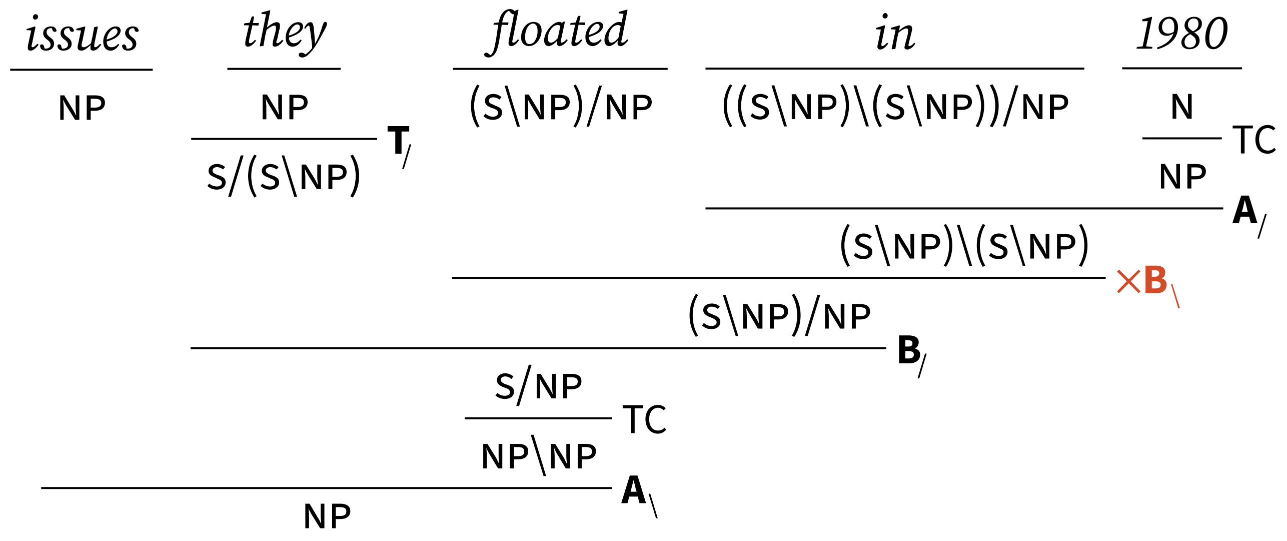


CCGbank's type-changing

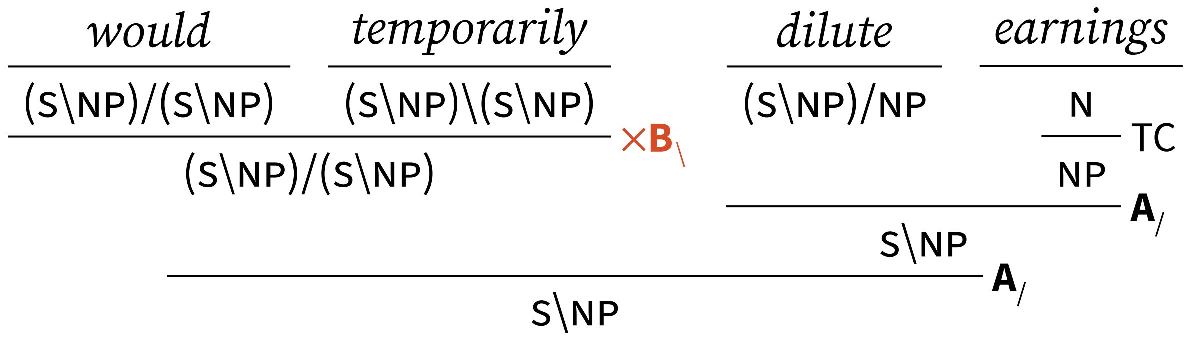
issues	*they*	*floated*	*in*	*1980*
NP	NP	(S\NP)/NP	((S\NP)\(S\NP))/NP	NP



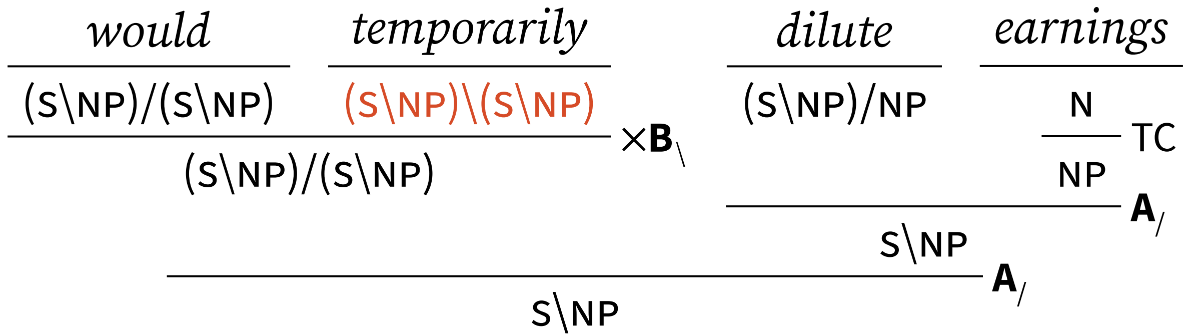
Crossed composition



Crossed composition

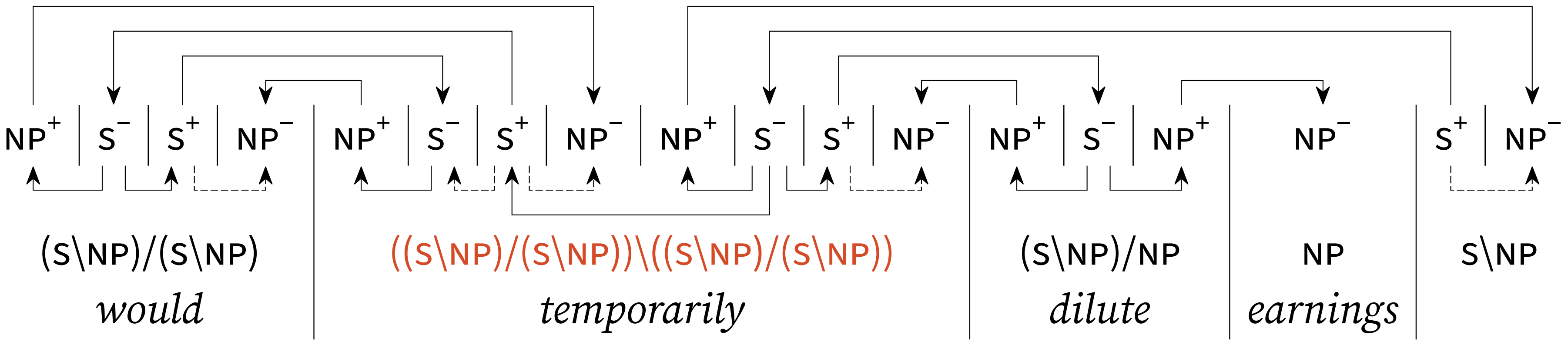


Crossed composition



Crossed composition

would	*temporarily*	*dilute*	*earnings*
(S\NP)/(S\NP)	(S\NP)\\(S\NP)	(S\NP)/NP	N



NP⁺ S⁻ S⁺ NP⁻ NP⁺ S⁻ S⁺ NP⁻ NP⁺ S⁻ S⁺ NP⁻ NP⁺ S⁻ NP⁺ NP⁻ S⁺ NP⁻
(S\NP)/(S\NP)
would
((S\NP)/(S\NP))\((S\NP)/(S\NP))
temporarily
(S\NP)/NP
dilute
NP
earnings
S\NP

Coordination

- CCGbank assigns a special conj category to coordinators (and, or, etc.)
- Usual interpretation is as polymorphic $(x\backslash x)/x$
  - But only for *like* coordination

Coordination

- CCGbank assigns a special conj category to coordinators (*and*, *or*, etc.)
- Usual interpretation is as polymorphic $(x\backslash x)/x$
  - But only for *like* coordination

$$\dfrac{\dfrac{\textit{doors}}{\mathrm{NP}} \qquad \dfrac{\textit{and}}{\mathrm{conj}} \qquad \dfrac{\textit{corners}}{\mathrm{NP}}}{\mathrm{NP}}\,\Phi$$

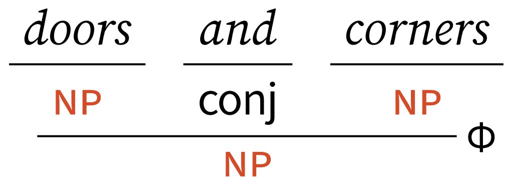
Coordination

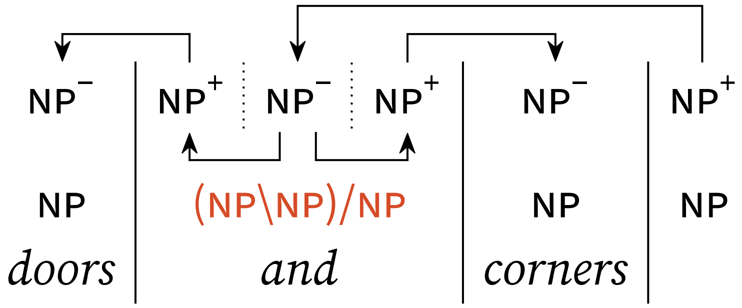
- CCGbank assigns a special conj category to coordinators (*and*, *or*, etc.)
- Usual interpretation is as polymorphic $(x\backslash x)/x$
  - But only for *like* coordination

$$\begin{array}{ccc} \underline{\textit{doors}} & \underline{\textit{and}} & \underline{\textit{corners}} \\ \mathrm{NP} & \mathrm{conj} & \mathrm{NP} \\ \hline \multicolumn{3}{c}{\mathrm{NP}} \end{array}\ \Phi$$

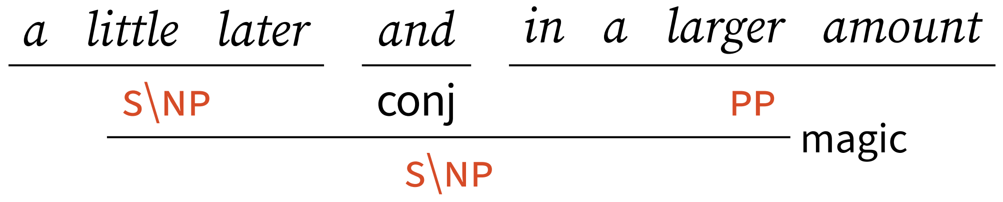
Coordination

- CCGbank assigns a special conj category to coordinators (*and*, *or*, etc.)
- Usual interpretation is as polymorphic $(x\backslash x)/x$
  - But only for *like* coordination

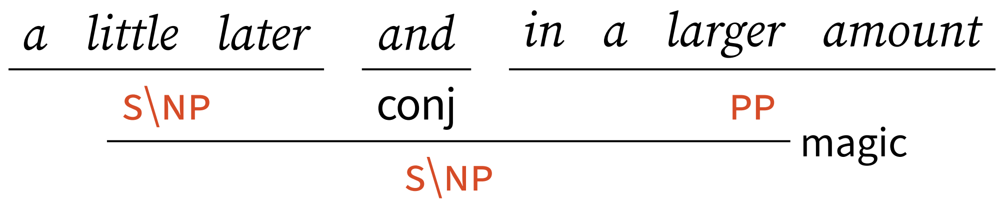


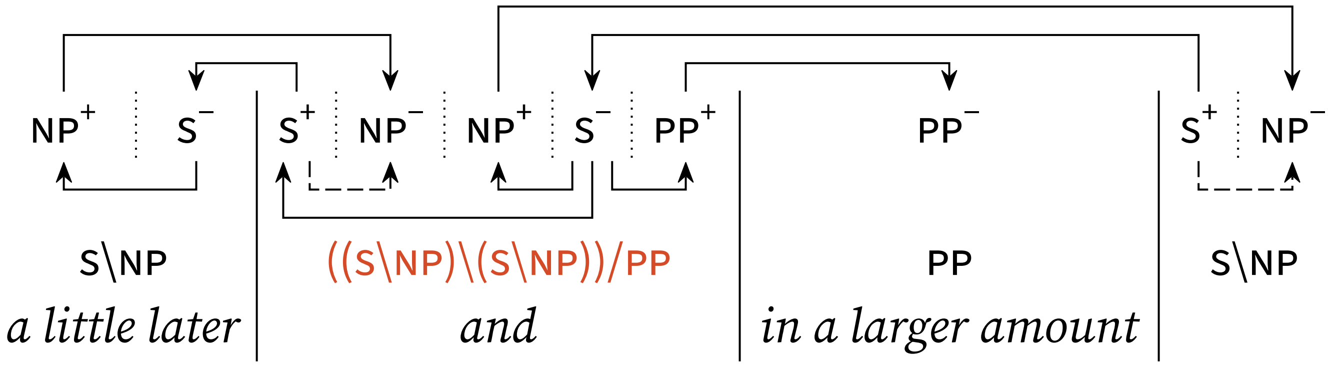


Coordination: *unlike*



Coordination: *unlike*





Manual annotations

- 274 sentences from PTB omitted from CCGbank (e.g., sentential gapping)
  - We parsed these with a CCG parser and adjusted manually

Manual annotations

- 274 sentences from PTB omitted from CCGbank (e.g., sentential gapping)
  - We parsed these with a CCG parser and adjusted manually
- ~500 rules not accounted for by our conversion rules
  - We annotated these manually; most were annotation errors
- 40 sentences with links *within* lexical categories
  - We provide *additional* analyses without using these links

LCGbank: the corpus

- ~49k sentences with analyses
- Use all sections for data splits
- Released as a set of conversion scripts & data
  - Apache 2.0 license
  - CCGbank is required

Training set statistics

	CCGbank	CCGbank w/o feats	LCGbank
Sentences	44,614	44,614	44,870
Atomic categories	34	11	5
Lexical categories	1,327	487	1,071
Avg. cat. order	1.748	1.916	2.317
Avg. cats/word	1.701	1.577	1.947
Exp. cats/word	20.083	14.958	29.731

Training set statistics

	CCGbank	CCGbank w/o feats	LCGbank
Sentences	44,614	44,614	44,870
Atomic categories	34	11	5
Lexical categories	1,327	487	1,071
Avg. cat. order	1.748	1.916	2.317
Avg. cats/word	1.701	1.577	1.947
Exp. cats/word	20.083	14.958	29.731

Training set statistics

	CCGbank	CCGbank w/o feats	LCGbank
Sentences	44,614	44,614	44,870
Atomic categories	34	11	5
Lexical categories	1,327	487	1,071
Avg. cat. order	1.748	1.916	2.317
Avg. cats/word	1.701	1.577	1.947
Exp. cats/word	20.083	14.958	29.731

Training set statistics

	CCGbank	CCGbank w/o feats	LCGbank
Sentences	44,614	44,614	44,870
Atomic categories	34	11	5
Lexical categories	1,327	487	1,071
Avg. cat. order	1.748	1.916	2.317
Avg. cats/word	1.701	1.577	1.947
Exp. cats/word	20.083	14.958	29.731

Training set statistics

	CCGbank	CCGbank w/o feats	LCGbank
Sentences	44,614	44,614	44,870
Atomic categories	34	11	5
Lexical categories	1,327	487	1,071
Avg. cat. order	1.748	1.916	2.317
Avg. cats/word	1.701	1.577	1.947
Exp. cats/word	20.083	14.958	29.731

Training set statistics

	CCGbank	CCGbank w/o feats	LCGbank
Sentences	44,614	44,614	44,870
Atomic categories	34	11	5
Lexical categories	1,327	487	1,071
Avg. cat. order	1.748	1.916	2.317
Avg. cats/word	1.701	1.577	1.947
Exp. cats/word	20.083	14.958	29.731

Training set statistics

	CCGbank	CCGbank w/o feats	LCGbank
Sentences	44,614	44,614	44,870
Atomic categories	34	11	5
Lexical categories	1,327	487	1,071
Avg. cat. order	1.748	1.916	2.317
Avg. cats/word	1.701	1.577	1.947
Exp. cats/word	20.083	14.958	29.731

What we didn’t get to do

- Porting CCGbank dependencies
    - Motivation partially taken care of by proof nets
    - But still useful for evaluation
    - No good evaluation in place for statistical LCG parsing

What we didn’t get to do

- CCGrebank
  - Numerous improvements over CCGbank
  - Not readily available 😢

LCGbank

A Corpus of Syntactic Analyses Based on Proof Nets

Aditya Bhargava, Timothy A. D. Fowler, and Gerald Penn

LREC-COLING 2024