Proof net structure for neural Lambek categorial parsing

Aditya Bhargava and Gerald Penn
Department of Computer Science
University of Toronto

@rightaditya

Introduction

- Lambek categorial grammar (LCG): formalism related to CCG

$$\frac{\Delta \vdash X/Y \quad \Gamma \vdash Y}{\Delta, \Gamma \vdash X}\,/_{\mathrm{e}} \qquad \frac{\Gamma \vdash Y \quad \Delta \vdash X\backslash Y}{\Gamma, \Delta \vdash X}\,\backslash_{\mathrm{e}}$$

$$\frac{\Gamma, Y \vdash X}{\Gamma \vdash X/Y}\,/_{\mathrm{i}} \qquad \frac{Y, \Gamma \vdash X}{\Gamma \vdash X\backslash Y}\,\backslash_{\mathrm{i}}$$

$$\frac{}{X \vdash X}\,\text{axiom}$$

- No existing statistical LCG parsers
- LCG rules $\subset$ linear logic
- Proof nets: graphical representation of linear logic proofs
  - Abstract over irrelevant aspects
  - “Equivalent” proofs will have the same proof net
- We use term graphs, an enhanced type of proof net (Fowler, 2009, 2016)

LCG term graphs

- Input: lexical category list (antecedent), target category (consequent)

$S/(S\backslash NP)$	$(S\backslash NP)/PP$	PP/NP	NP/N	N	$\vdash S$
What	accounts	for	the	difference	?

LCG term graphs

- Add polarities
  - Lexical categories negative
  - Target category positive

$$
\begin{array}{cccccc}
(S/(S\backslash NP))^{-} & ((S\backslash NP)/PP)^{-} & (PP/NP)^{-} & (NP/N)^{-} & N^{-} & S^{+} \\
S/(S\backslash NP) & (S\backslash NP)/PP & PP/NP & NP/N & N & \vdash S \\
\text{What} & \text{accounts} & \text{for} & \text{the} & \text{difference} & ?
\end{array}
$$

LCG term graphs

- Decompose categories into polarized atoms

$$(X/Y)^- \;\Rightarrow\; X^- \rightarrow Y^+ \qquad\qquad (X\backslash Y)^- \;\Rightarrow\; Y^+ \leftarrow X^-$$

$$(X\backslash Y)^+ \;\Rightarrow\; X^+ \dashrightarrow Y^- \qquad\qquad (X/Y)^+ \;\Rightarrow\; Y^- \dashleftarrow X^+$$

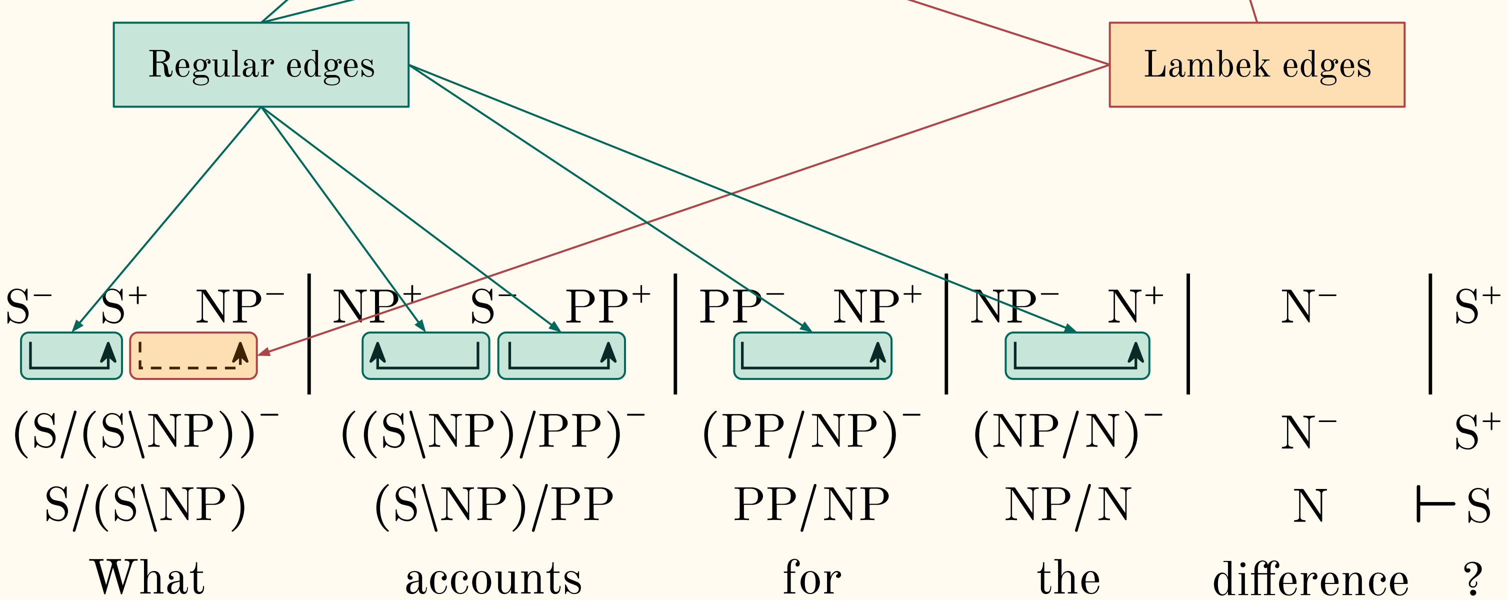
S^- S^+ NP^-	NP^+ S^- PP^+	PP^- NP^+	NP^- N^+	N^-	S^+
$(S/(S\backslash NP))^-$	$((S\backslash NP)/PP)^-$	$(PP/NP)^-$	$(NP/N)^-$	N^-	S^+
$S/(S\backslash NP)$	$(S\backslash NP)/PP$	PP/NP	NP/N	N	$\vdash S$
What	accounts	for	the	difference	?

LCG term graphs

- Decompose categories into polarized atoms

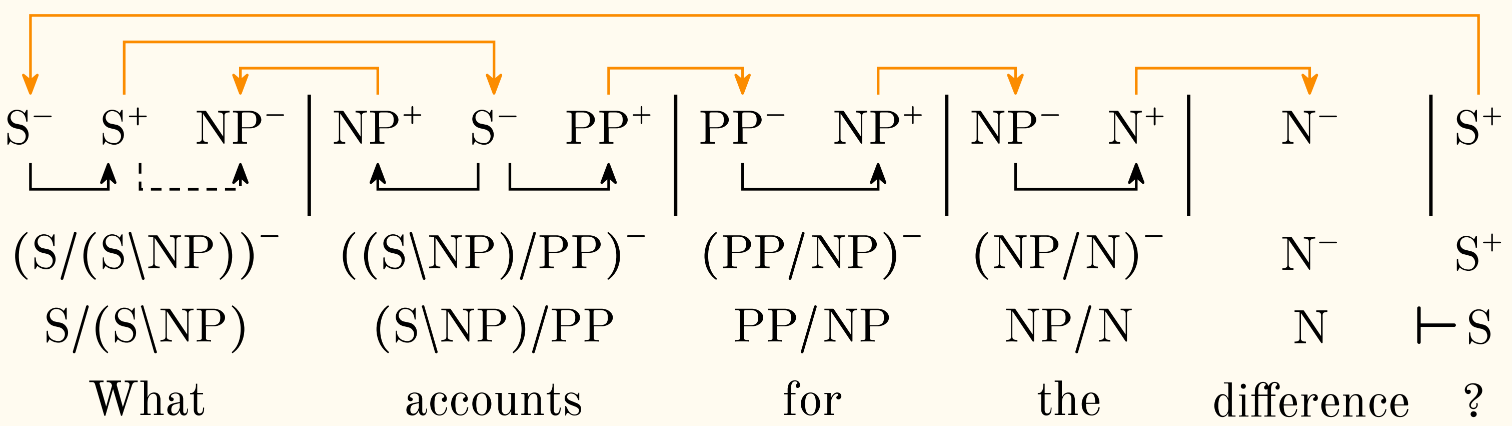
$$(X/Y)^- \Rightarrow X^- \boxed{\rightarrow} Y^+ \qquad (X\backslash Y)^- \Rightarrow Y^+ \boxed{\leftarrow} X^-$$

$$(X\backslash Y)^+ \Rightarrow X^+ \boxed{\dashrightarrow} Y^- \qquad (X/Y)^+ \Rightarrow Y^- \boxed{\dashleftarrow} X^+$$



LCG term graphs

- Link positive atoms to negative atoms of same atomic category



Term graph validity conditions

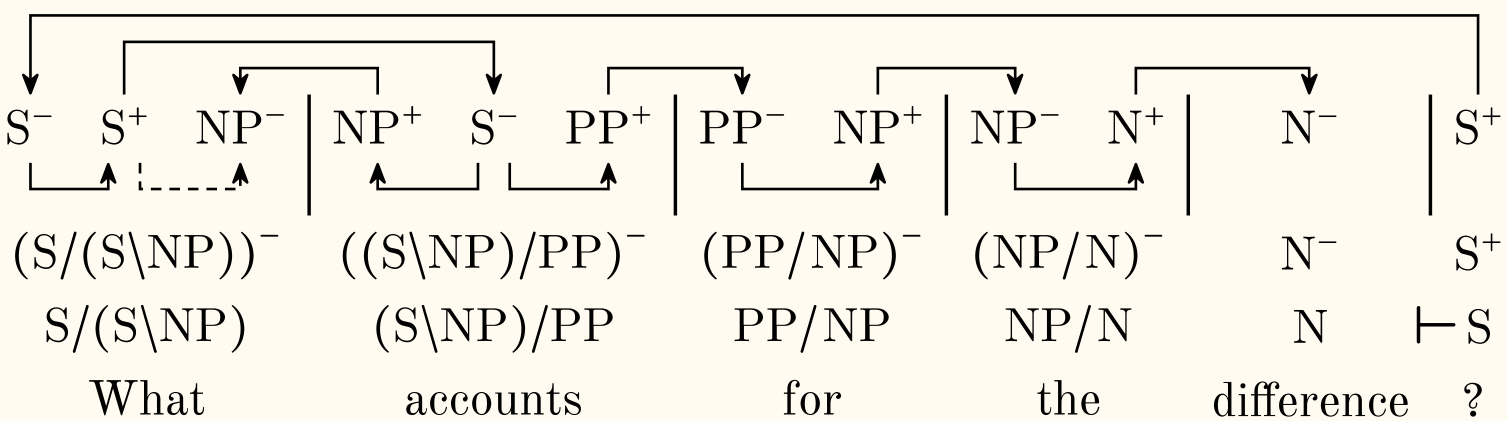
T1. Linkage must be half-planar

- No crossing edges in half-plane above vertices

T2. No regular cycles

- Links included as regular edges

T3. Each Lambek edge must have regular path between its vertices



Term graph validity conditions

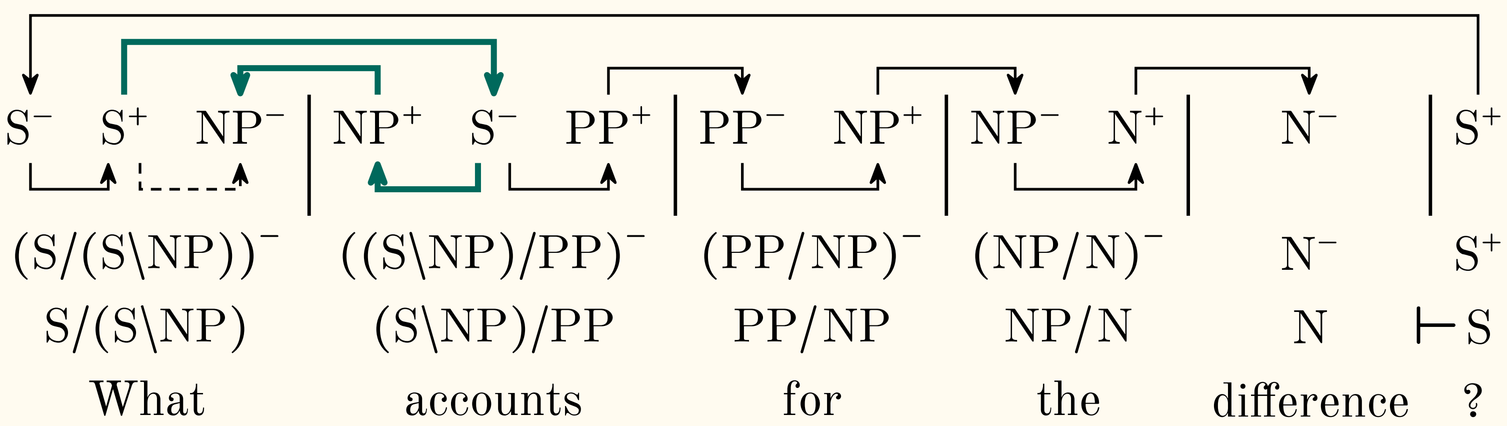
T1. Linkage must be half-planar

- No crossing edges in half-plane above vertices

T2. No regular cycles

- Links included as regular edges

T3. Each Lambek edge must have regular path between its vertices



Term graph validity conditions

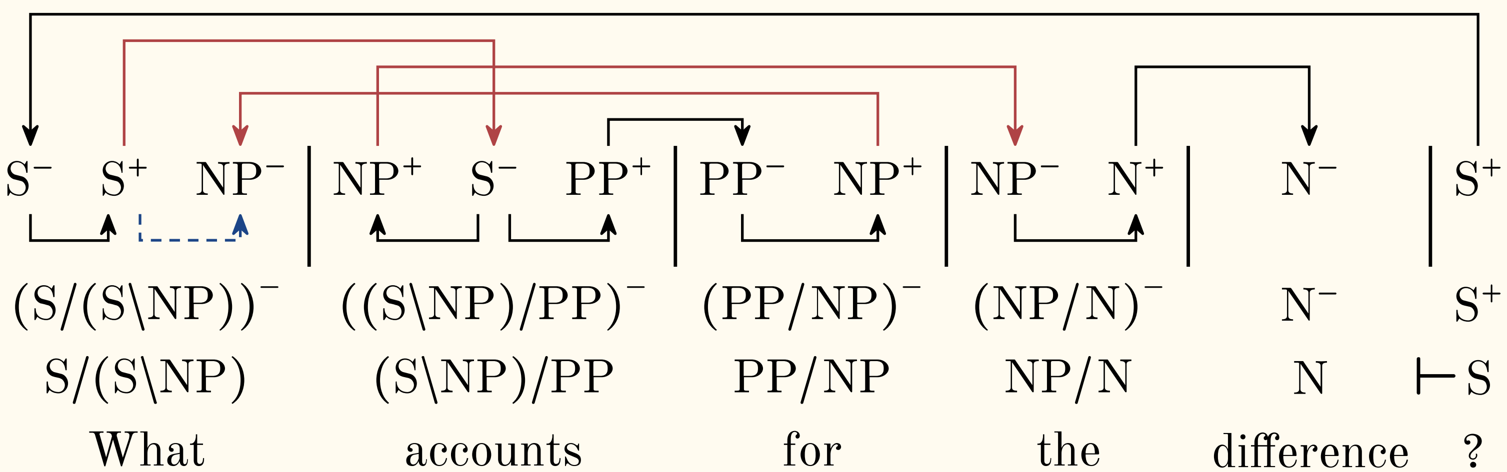
T1. Linkage must be half-planar

- No crossing edges in half-plane above vertices

T2. No regular cycles

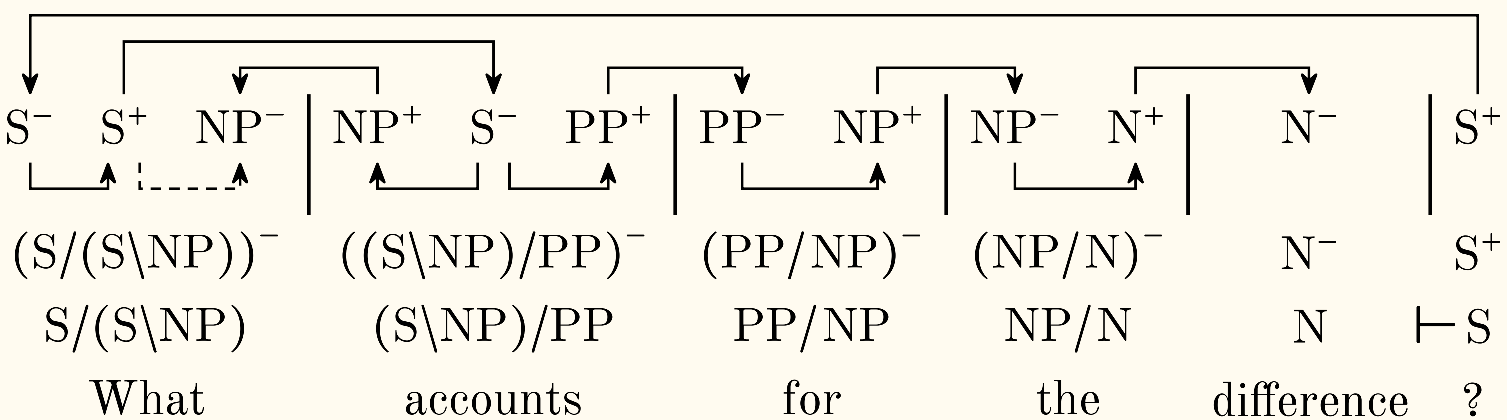
- Links included as regular edges

T3. Each Lambek edge must have regular path between its vertices



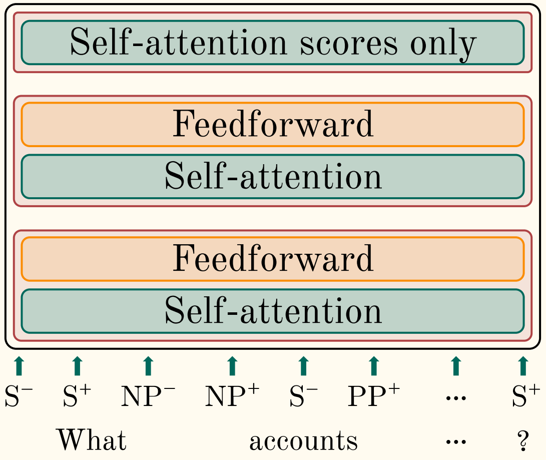
LCG term graphs

- Compact representation
  - Dependency-like structure
  - No spurious ambiguity
- Here, we assume lexical and target categories are given
  - Task is then to predict correct linkage
    - Failing that, linkage should still yield valid term graph



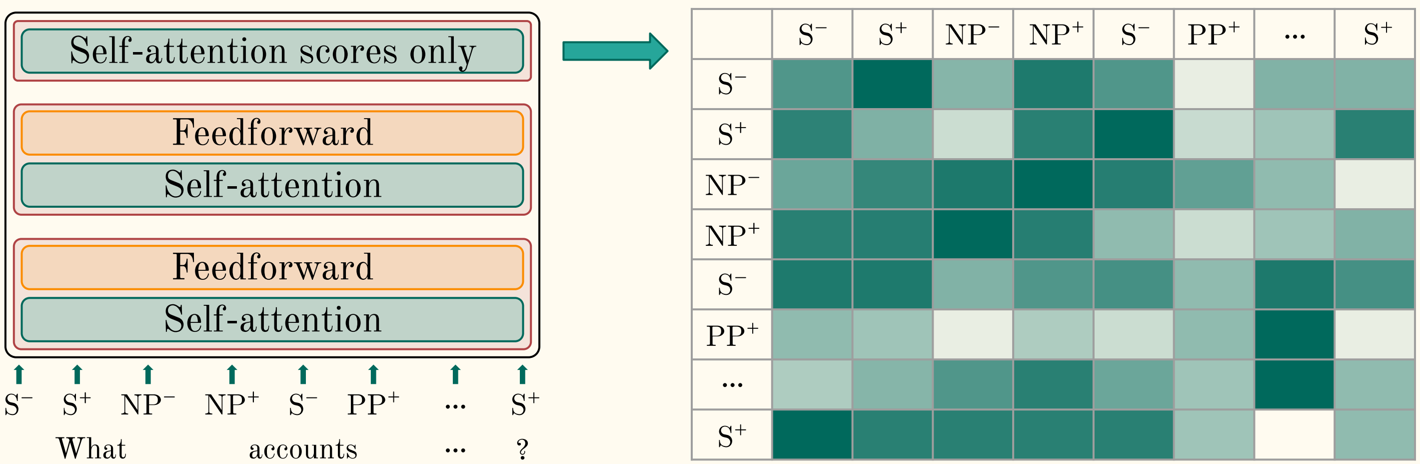
Predicting linkages with self-attention networks

- Input: words and corresponding lexical categories (decomposed to atoms)
- Output: valid linkage
- Base model similar to Transformer encoder



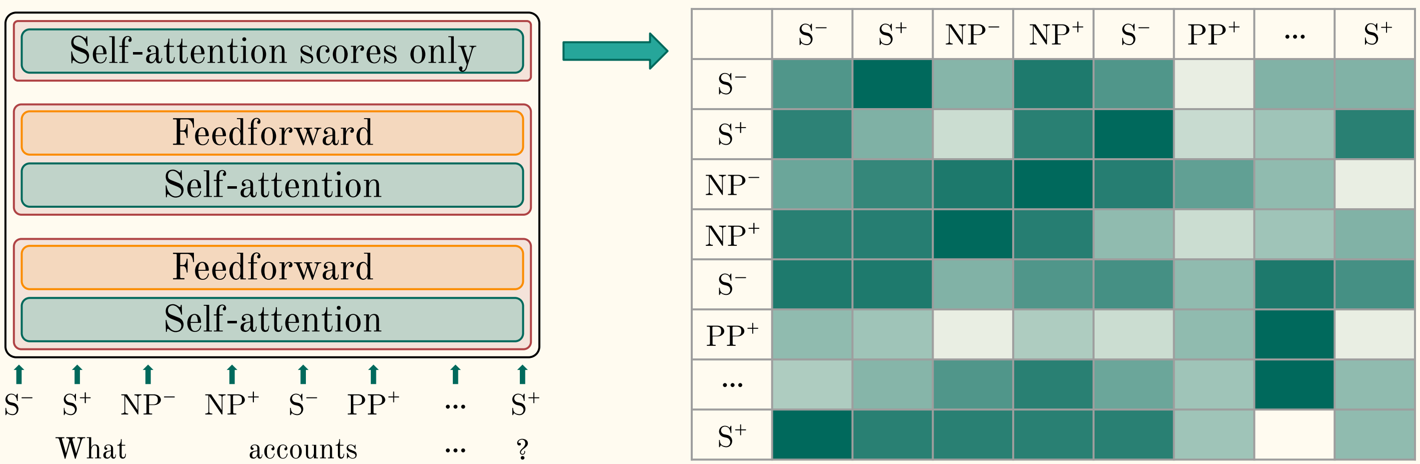
Predicting linkages with self-attention networks

- Input: words and corresponding lexical categories (decomposed to atoms)
- Output: valid linkage
- Base model similar to Transformer encoder
  - Top layer omits softmax onwards, leaving raw attention scores



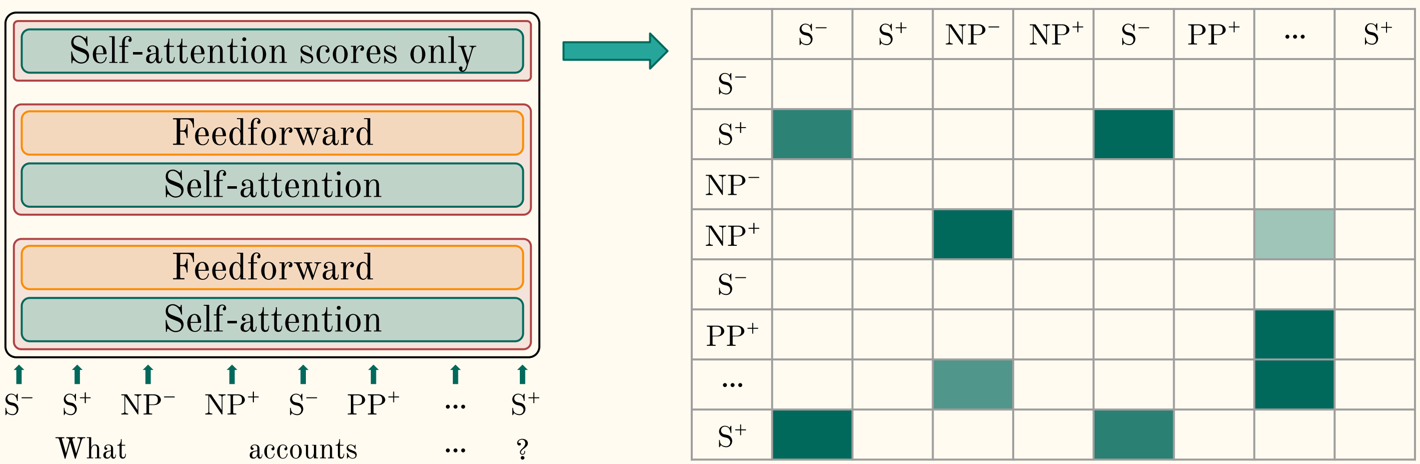
Predicting linkages with self-attention networks

- Input: words and corresponding lexical categories (decomposed to atoms)
- Output: valid linkage
- Base model similar to Transformer encoder
    - Top layer omits softmax onwards, leaving raw attention scores
    - Keep only links from positive to negative atoms of same type



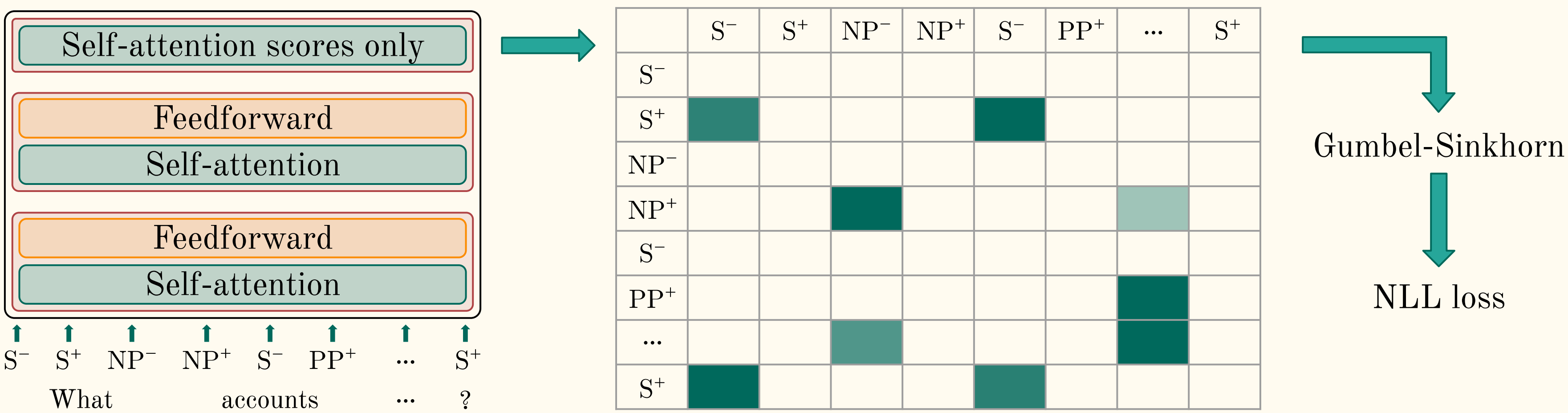
Predicting linkages with self-attention networks

- Input: words and corresponding lexical categories (decomposed to atoms)
- Output: valid linkage
- Base model similar to Transformer encoder
  - Top layer omits softmax onwards, leaving raw attention scores
  - Keep only links from positive to negative atoms of same type



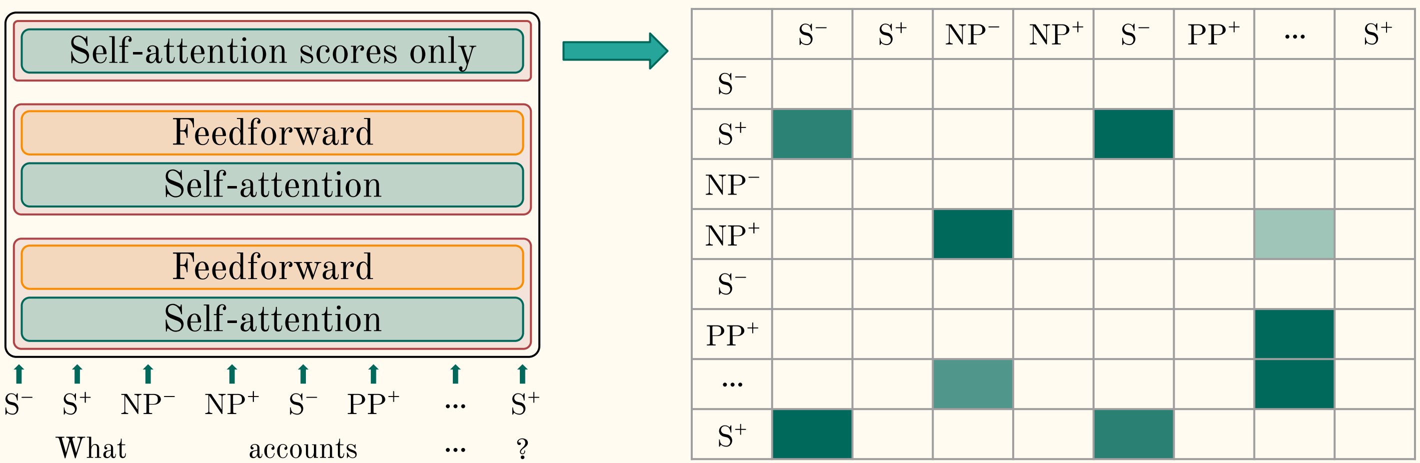
Predicting linkages with self-attention networks

- Input: words and corresponding lexical categories (decomposed to atoms)
- Output: valid linkage
- Base model similar to Transformer encoder
  - Top layer omits softmax onwards, leaving raw attention scores
  - Keep only links from positive to negative atoms of same category
  - Scores run through Gumbel-Sinkhorn yield doubly-stochastic matrix
  - Negative log likelihood loss



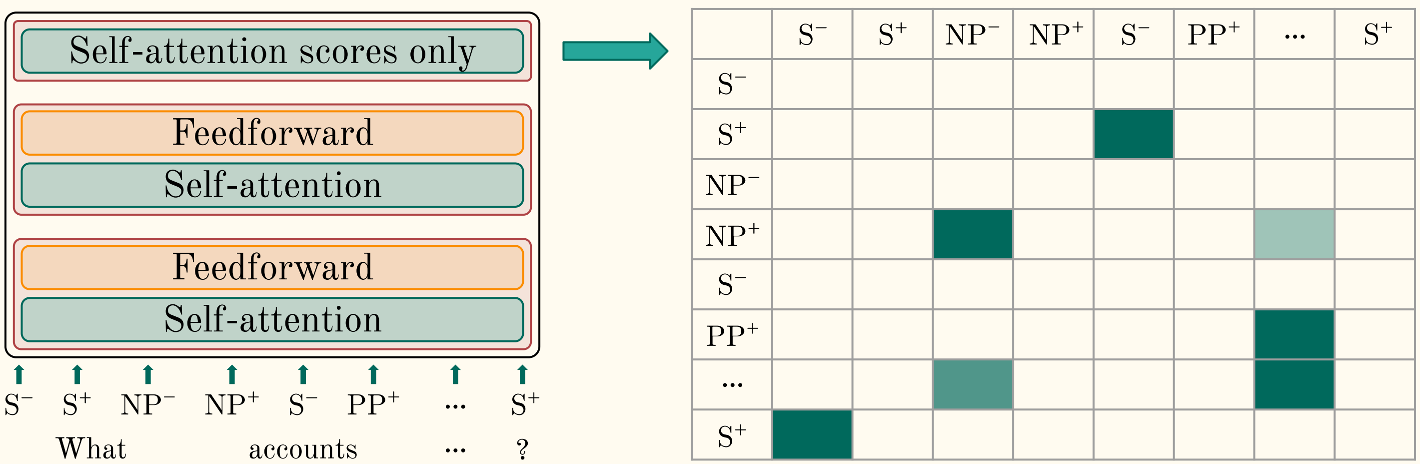
Term graph–based model enhancements

- Disallow intra-word links



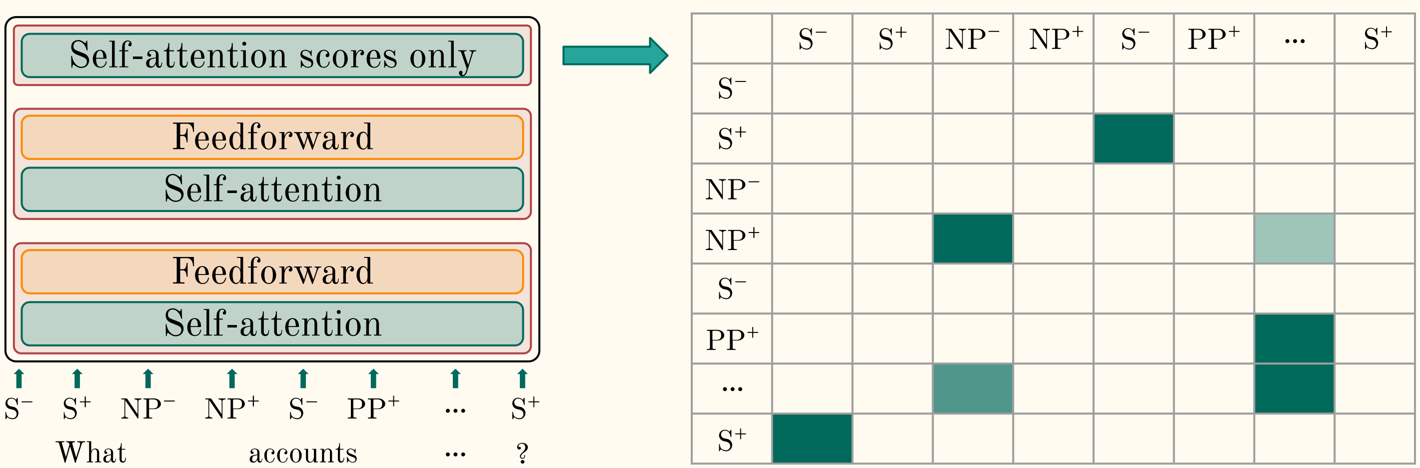
Term graph–based model enhancements

- Disallow intra-word links
  - In this example, S links are settled



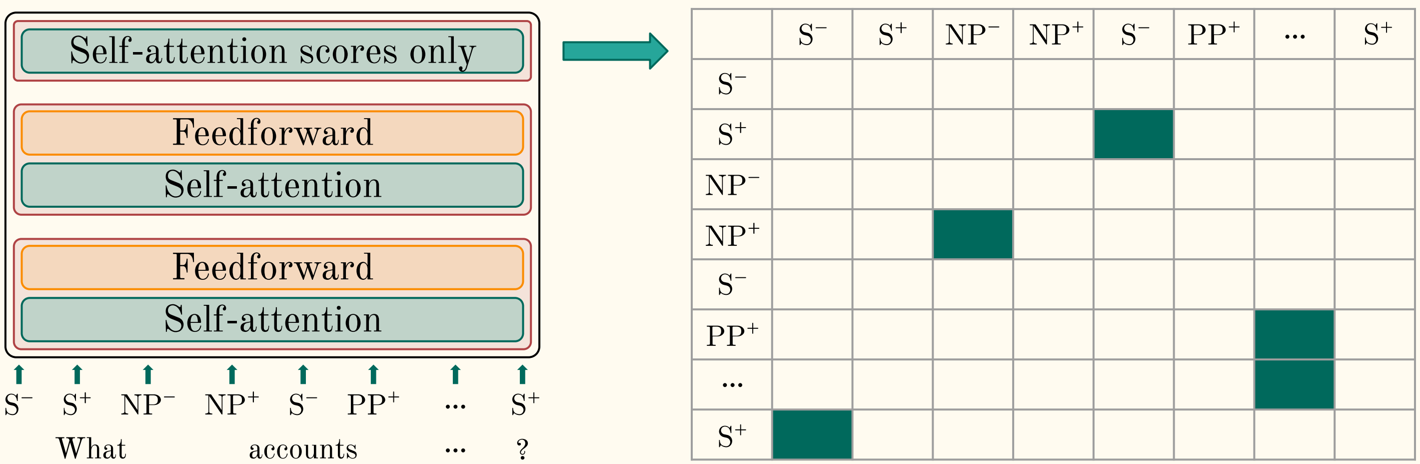
Term graph–based model enhancements

- Disallow intra-word links
    - In this example, S links are settled
- Disallow necessarily non-planar links



Term graph–based model enhancements

- Disallow intra-word links
  - In this example, S links are settled
- Disallow necessarily non-planar links
  - In this example, settles everything else

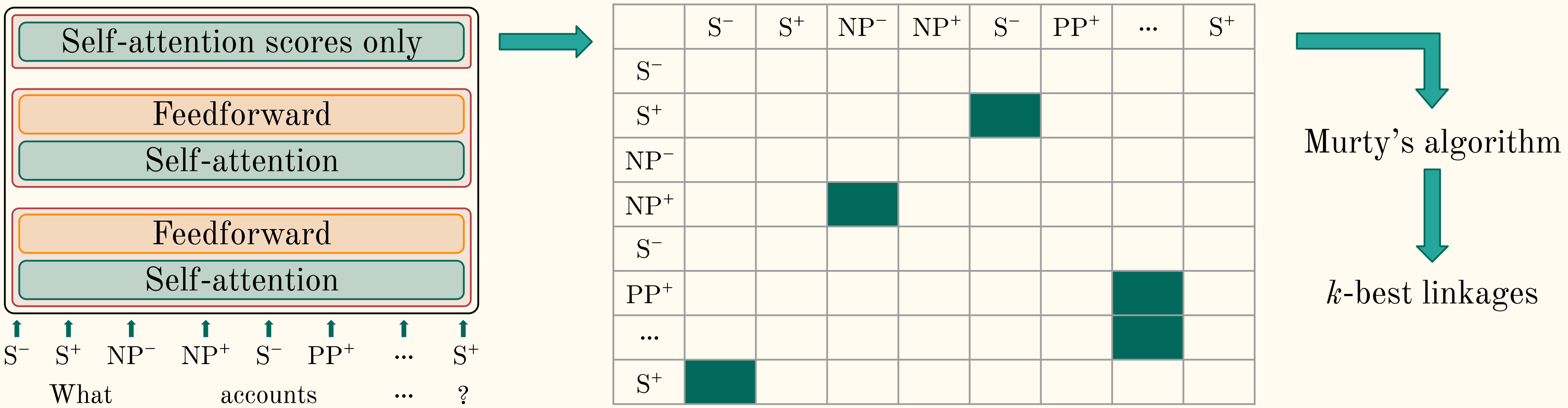


Term graph–based model enhancements

- Disallow intra-word links
  - In this example, S links are settled
- Disallow necessarily non-planar links
  - In this example, settles everything else
- Regular and Lambek edges into attention queries and keys
  - Similar to message passing along regular and Lambek edges
- Penalize attention scores to encourage planarity
  - Imagine each attention score is an edge score
  - Penalize edge score according to scores assigned to crossing edges

Enhancements to inference

- No need for Gumbel-Sinkhorn during inference
  - Use maximum bipartite matching algorithms to get highest-scoring linkage
  - Murty's algorithm for k-best linkages



Novel loss functions

- Term graph validity conditions as loss functions
- T1 (half-planarity): penalize crossing links in proportion to their model scores
- T2 & T3: penalize edges that contribute to condition violations
    - Key: transitive closure of candidate graph (linkage scores, regular & Lambek edges)
    - Computable differentiably
    - Select source and destination vertices of interest to penalize violations
- Loss terms are functions of model output only
    - Enables training without ground-truth derivations

Ground-truth experiments*

- Three conditions
  1. Base model with NLL loss
  2. Enhanced model with NLL loss
  3. Enhanced model with NLL loss + losses derived from term graph conditions
- Three measures
  - Link accuracy
  - Sentence accuracy
  - Coverage
- $k = 1$ and $k = 512$
- Corpus: LCGbank

*See paper for training details such as hyperparameters, etc.

Ground-truth experiments*

- Three conditions
  1. Base model with NLL loss
  2. Enhanced model with NLL loss
  3. Enhanced model with NLL loss + losses derived from term graph conditions
- Three measures
  - Link accuracy
  - Sentence accuracy
  - Coverage
- $k = 1$ and $k = 512$
- Corpus: LCGbank

	$k=1$			$k=512$		
Condition	Link Acc	Sent Acc	Coverage	Link Acc	Sent Acc	Coverage
Base	97.7	86.2	97.3	97.9	87.7	99.8
Enhanced model	97.9	87.4	98.4	98.0	88.2	99.9
Enhanced model + losses	97.9	87.2	98.7	98.0	87.8	99.9

*See paper for training details such as hyperparameters, etc.

Ground-truth–free experiments*

- Enhanced model with losses derived from term graph conditions only
    - (No NLL loss)
- Ablation on various pieces of model/loss
- Coverage is reported measure
    - No way to distinguish correct derivation

*See paper for training details such as hyperparameters, etc.

Ground-truth–free experiments*

- Enhanced model with losses derived from term graph conditions only
  - (No NLL loss)
- Ablation on various pieces of model/loss
- Coverage is reported measure
  - No way to distinguish correct derivation

Condition	$k = 1$	$k = 512$
Enhanced model + losses	91.2	96.2
—T1 loss	84.5	95.1
—T2 loss	72.9	92.9
—T3 loss	70.6	93.8
—Regular/Lambek edges	89.0	95.9
—Intraword link filter	81.1	91.0
—Nonplanar link filter	73.9	85.6
—R/L edges — planar attention	74.9	90.7
—planar attention — T1 loss	19.2	44.7

*See paper for training details such as hyperparameters, etc.

Ground-truth–free experiments*

- Enhanced model with losses derived from term graph conditions only
  - (No NLL loss)
- Ablation on various pieces of model/loss
- Coverage is reported measure
  - No way to distinguish correct derivation

Condition	$k = 1$	$k = 512$
Enhanced model + losses	91.2	96.2
—T1 loss	84.5	95.1
—T2 loss	72.9	92.9
—T3 loss	70.6	93.8
—Regular/Lambek edges	89.0	95.9
—Intraword link filter	81.1	91.0
—Nonplanar link filter	73.9	85.6
—R/L edges — planar attention	74.9	90.7
—planar attention — T1 loss	19.2	44.7

All planarity information removed

*See paper for training details such as hyperparameters, etc.

Summary & future work

- Incorporating term graph structure can increase parser accuracy and coverage
- Term graph conditions allow specification of novel loss terms
  - Enable training high-coverage without ground-truth derivations
  - Potential applications to unsupervised & semi-supervised parsing
- Parser is differentiable function of inputs, i.e., supertags
  - Potential for improving joint supertagger/parser

Proof net structure for neural Lambek categorial parsing

Aditya Bhargava and Gerald Penn
Department of Computer Science
University of Toronto

@rightaditya