The Limitations of Limited Context for Constituency Parsing

(The presenter you see on Zoom)

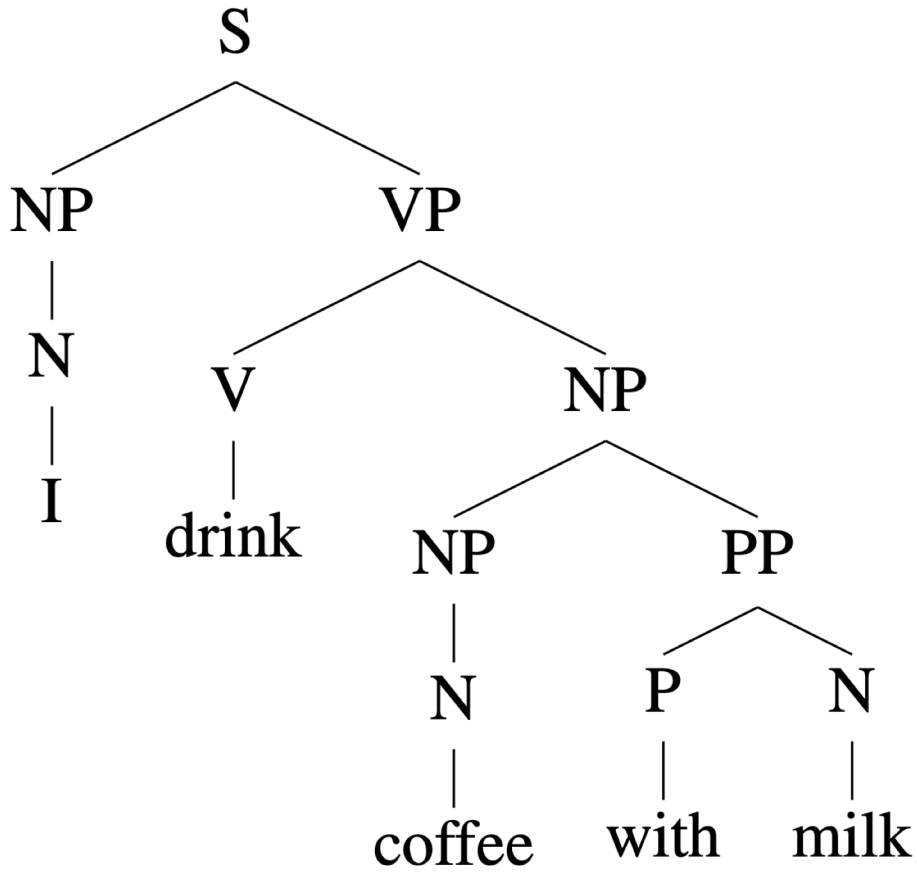
Yuchen Li
(CMU)

Andrej Risteski
(CMU)

https://arxiv.org/abs/2106.01580

Background: constituency parsing

- Given a sentence, return the parse tree
- E.g., consider the sentence
  - *"I drink coffee with milk"*
- *Side note*: there are other types of parsing, e.g. dependency parsing



(Unsupervised) parsing is hard

- Some recent advances (PRPN, ON-LSTM, Compound PCFG)

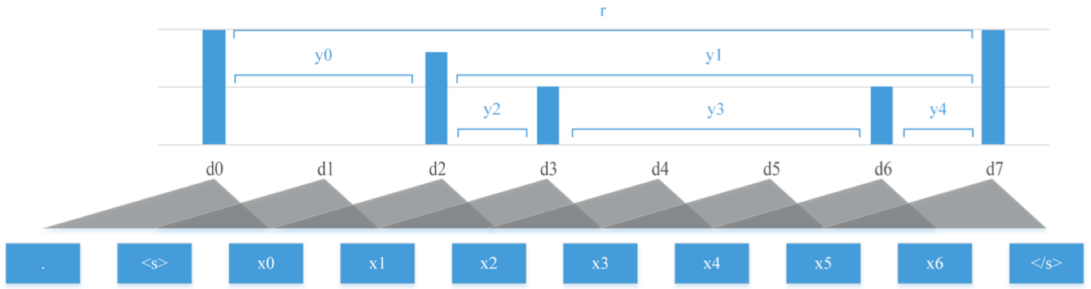
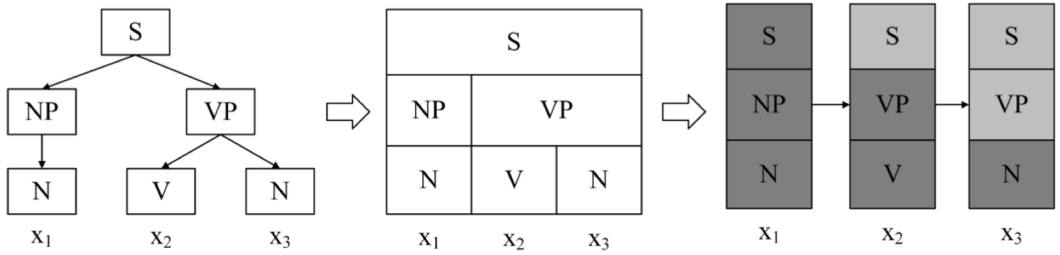
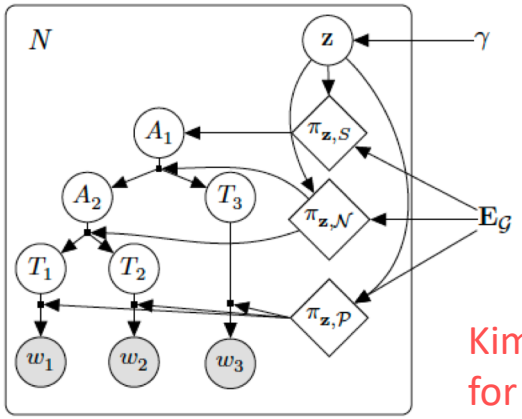


Figure 3: Convolutional network for computing syntactic distance. Gray triangles represent 2 layers

Shen et al., Neural Language Modeling by Jointly Learning Syntax and Lexicon, ICLR 2018



Shen et al., Ordered Neurons, ICLR 2019



Kim et al., Compound Probabilistic Context-Free Grammars for Grammar Induction, ACL 2019

(Unsupervised) parsing is hard

- However, some of these models (e.g. PRPN) tend to be[1]
  - Complex
  - Greedy
  - Left to right
- It is unclear what kinds of structure they can (or cannot) capture

1. Intuitions also discussed by:
Phu Mon Htut, Kyunghyun Cho, Samuel Bowman. Grammar induction with neural language models: An unusual replication. BlackboxNLP 2018

Goal of this paper

- What is the representational power of empirically successful models for parsing?
  - (i.e. what kinds of syntactic structure can they recover?)

Overview of results

i.e. when each parsing decision must be made, how *myopic* is the model?

i.e. context = everything to the left + bounded look-ahead to the right

mount of context is a key factor for representational power

Which model relevant characteristics of natural language

Flavor of main result:

For left-to-right parsing approaches, there exists a family of PCFGs, such that their parsing accuracy is arbitrarily bad.

But with full context, they can (in theory) possibly work perfectly, i.e. full representational power.

Goal of this paper

- What is the representational power of empirically successful models for parsing?
  - i.e. what kinds of structure can they recover?
  - **Our result**: representational power crucially depends on the amount of *context*
- How do we reason about a "ground truth" syntactic structure the model should recover?
  - In this paper: probabilistic context-free grammar (PCFG)

Background: context-free grammar

- $G = (\Sigma, N, S, R)$
- Σ: the set of terminals
- N: the set of non-terminals
- $S \in N$: the start symbol
- R: the set of production rules of the form

$$r_L \rightarrow r_R$$

where $r_L \in N$, r_R is a sequence of Σ and N

Background: probabilistic context-free grammar (PCFG)

- $G = (\Sigma, N, S, R, \Pi)$
- Σ: the set of terminals
- N: the set of non-terminals
- $S \in N$: the start symbol
- R: the set of production rules of the form

$$r_L \rightarrow r_R$$

where $r_L \in N$, r_R is a sequence of Σ and N

Additionally,
each rule is associated with a
conditional probability

P(right | left)

Background: Chomsky normal form (CNF)

- Each rule is of one of the following forms[1]:

$$A \rightarrow B_1 B_2$$
$$A \rightarrow a$$

- Every CFG G can be converted into a CFG G' in CNF such that[2]

$$L(G) = L(G')$$

- In our paper, we will assume that the ground-truth parse is in CNF
  - The empirically successful parsing algorithms output binary trees
  - Among binary tree representations, CNF is a convenience

1. Plus some requirements related to empty strings
2. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. 2006. Introduction to Automata Theory, Languages, and Computation (3rd Edition).

Goal of this paper

- What is the representational power of empirically successful models for parsing?
  - i.e. what kinds of structure can they recover?
  - Our claim: it depends on the "context"
- What is a groundtruth for syntax that we can quantify?
  - In this paper: probabilistic context-free grammar (PCFG)
- When is a "framework for parsing" able to represent the max likelihood parse of a sentence generated from a PCFG?
  - In this paper: frameworks based on syntactic distance and transition-based parsers.

Syntactic distance-based parsing

Syntactic distance

c_t is the context that d_t takes into consideration.
(Note that this is **not** a conditional distribution.)

Step 1: Learn/train a *syntactic distance* $d_t := d(w_{t-1}, w_t | c_t)$ between adjacent words.

(Parametrized by a shallow convolutional neural network, trained using language modeling objective.)

Step 2: Produce a parse tree, based on greedily branching at the point of largest syntactic distance.

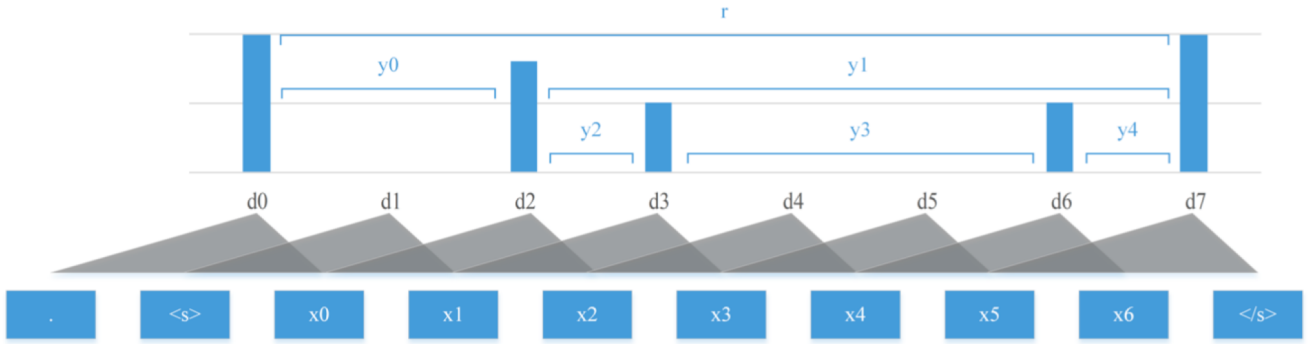


Figure 3: Convolutional network for computing syntactic distance. Gray triangles represent 2 layers

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, Aaron Courville. Neural Language Modeling by Jointly Learning Syntax and Lexicon. ICLR 2018

Tree induction based on syntactic distance

- Initialize the root to contain the whole sentence
- When there exists a single segment consisting of 2 or more words
- Split at the point with the largest syntactic distance
- Return the binary tree whose leaves each contains 1 single word

Yikang Shen, Zhouhan Lin, Chin-wei Huang, Aaron Courville. Neural language modeling by jointly learning syntax and lexicon. ICLR 2018

Representing PCFG with syntactic distance

Definition: Let G be any PCFG in Chomsky Normal Form.
A syntactic distance function d is said to be able to $\boldsymbol{p}$-represent G,
if for a set of sentences in $L(G)$ whose total probability is at least p,
d can correctly induce the max likelihood parse of these sentences unambiguously.

Ambiguities: e.g. if $d_t = d_{t+1}$,
how to parse $w_{t-1}w_t w_{t+1}$?

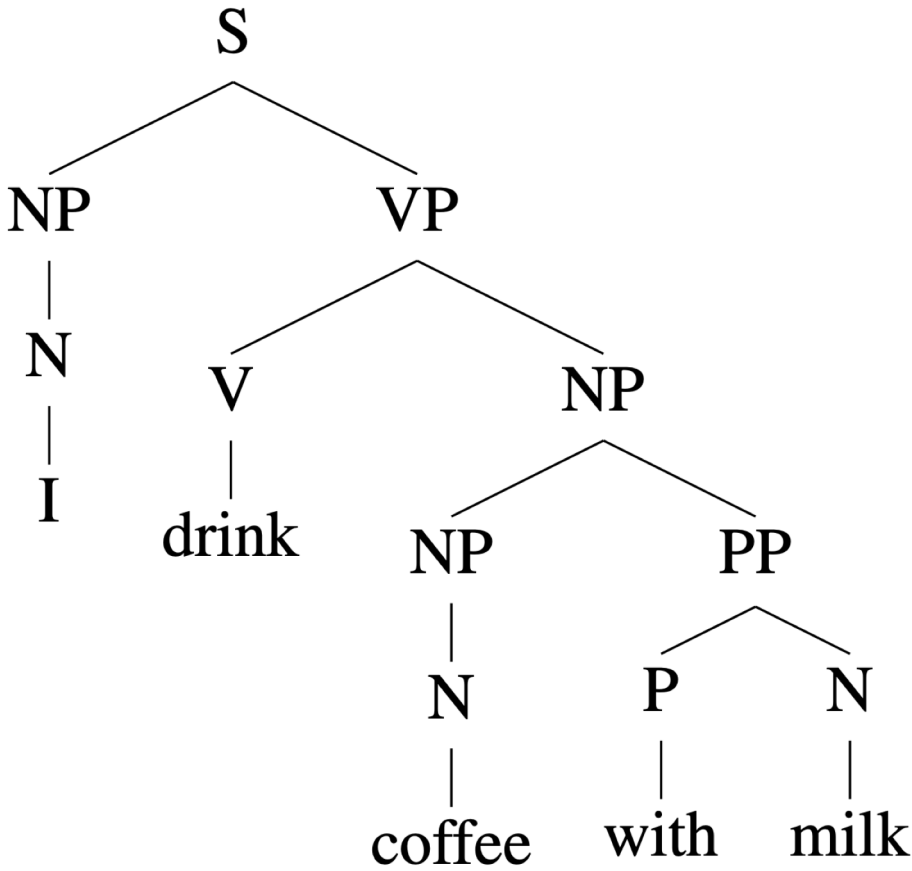
Syntactic distance with full context

Recall: $d_t = d(w_{t-1}, w_t | c_t)$, for all $2 \leq t \leq n$

Theorem: Let $c_t = (W, t)$.
For each PCFG G in Chomsky normal form,
there exists a syntactic distance measure
$d_t = d(w_{t-1}, w_t | c_t)$,
that can 1-represent G.

Proof ideas (full context)

- Given sentence W generated by PCFG G in Chomsky normal form
- Let binary tree T be the max likelihood parse tree of W
- *Recall*: $d_t = d(w_{t-1}, w_t | c_t)$, for all $2 \leq t \leq n$
- d_t depends on the entire ground truth T
- Find an assignment of d_t (t = 2..n)
- such that their order matches the level at which the branches split in T



$d_2 = 5$ $d_3 = 4$ $d_4 = 3$ $d_5 = 2$

Set: $d_t = n - tree\_dist(root,\ lowest\ common\ ancestor\ of\ w_{t-1}\ \&\ w_t)$

Syntactic distance with limited context

Recall: $\mathrm{W} = w_1 w_2 \ldots w_n$

Additionally: w_0 is the sentence start symbol.

$d_\mathrm{t} = d(w_{t-1}, w_t | c_t)$, for all $2 \leq t \leq n$

The context includes everything to the left. We additionally allow a look-ahead window of size L'.

Theorem: Let $\mathrm{c_t} = (w_0 w_1 w_2 \ldots w_{t+L'})$.

$\forall \epsilon > 0, \exists$ PCFG G in Chomsky normal form,

such that any syntactic distance measure

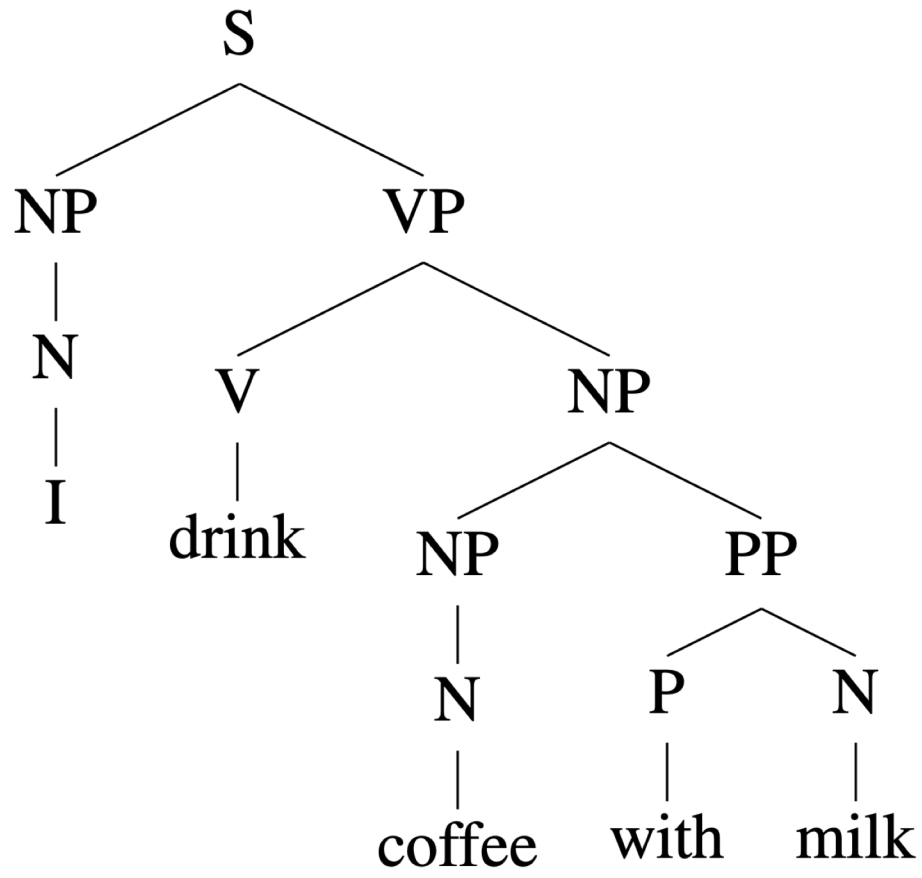
$d_\mathrm{t} = d(w_{t-1}, w_t | c_t)$,

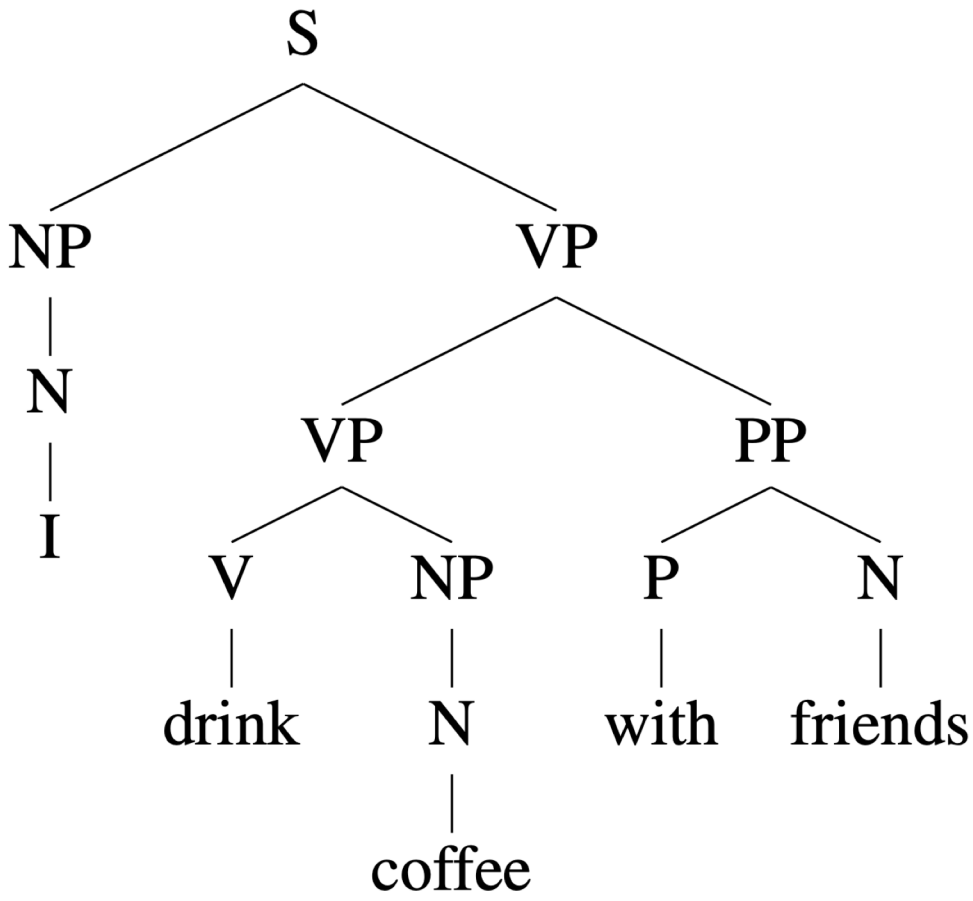
cannot ϵ-represent G.

This statement analyzes the left-to-right parsing. A symmetric result can be proven for right-to-left parsing.

Proof ideas - motivation

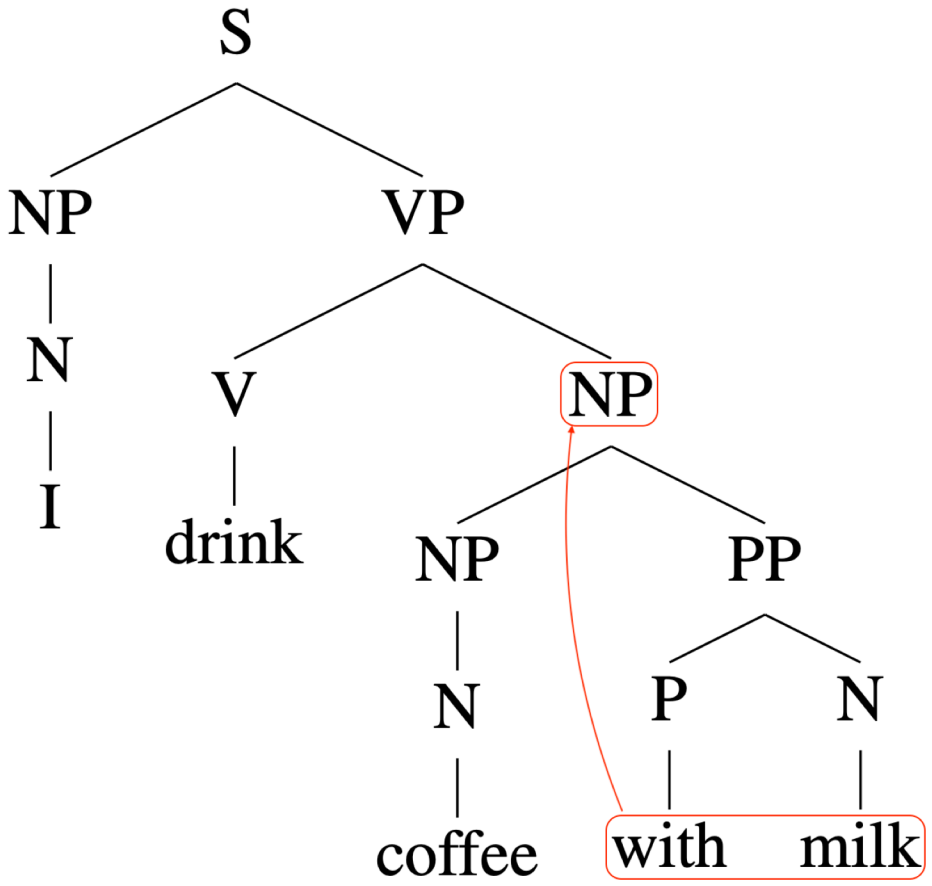
- The earlier parses can depend on later words

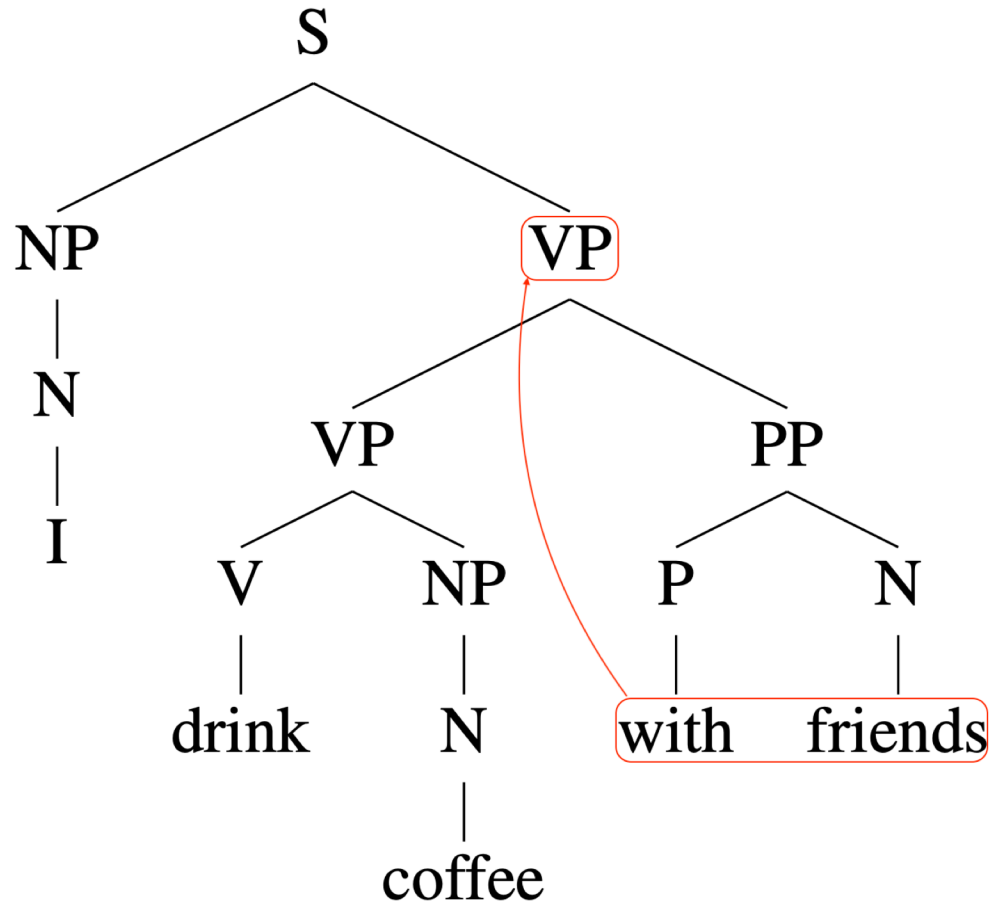




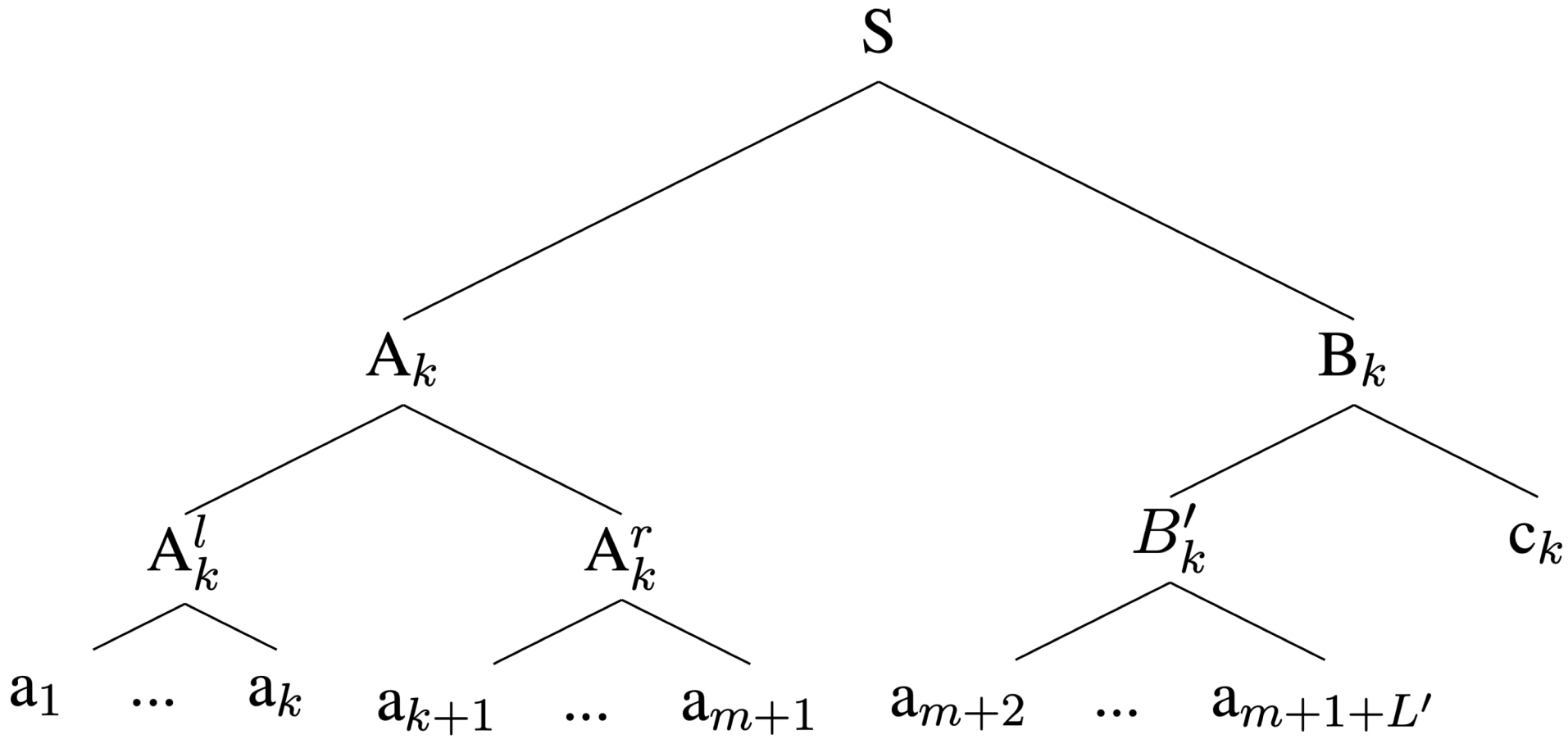
Proof ideas - motivation

- The earlier parses can depend on later words



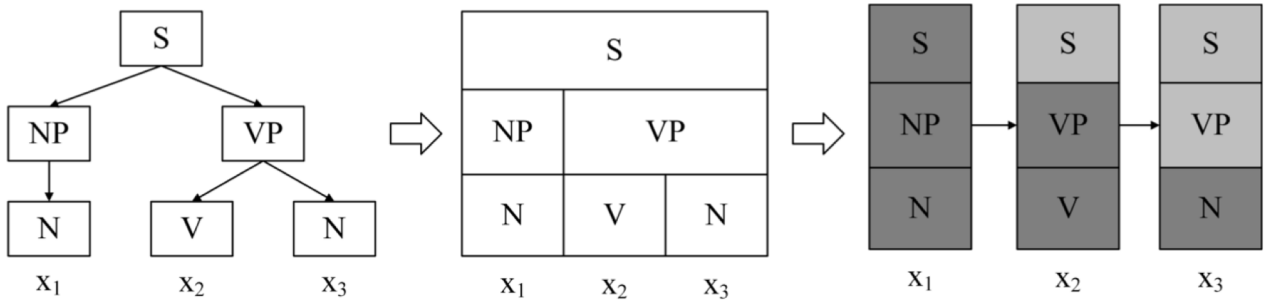


Proof ideas - counterexample grammar



Ordered Neuron LSTM

- Using language modeling objective, train some carefully structured gate vectors at each position
- Reduce the learned gates to distances
- Induce the parse tree by greedily branching
- Similar results & proofs apply



Yikang Shen, Shawn Tan, Alessandro Sordoni, Aaron Courville. Ordered Neurons. ICLR 2019

Remarks on frameworks considered

- RNN, in its most general form, can be proved to be Turing-complete[1]
  - Unbounded time and memory
- On the other hand, the one-pass RNN, context = everything to the left
  - Has similar power to finite state automaton[2]
  - Insufficient power for parsing PCFGs
- We consider an intermediate setting
  - More general than the simple one-pass RNN
  - When the parsing decision at each word is calculated
  - The context can include a finite look-ahead (to the right) but not the whole sentence

1. Hava T. Siegelmann and Eduardo D. Sontag. On the computational power of neural nets. COLT 1992

2. William Merrill. Sequential Neural Networks as Automata. ACL 2019

Summary: syntactic distance & ON-LSTM

Recall: $\mathrm{W} = w_1 w_2 \dots w_n$

Additionally: w_0 is the sentence start symbol.

$d_\mathrm{t} = d(w_{t-1}, w_t | c_t)$, for all $2 \leq t \leq n$

Theorem (full context):

Let $\mathrm{c_t} = (\mathrm{W}, \mathrm{t})$.

For each PCFG G in Chomsky normal form,

there exists a syntactic distance measure

$d_\mathrm{t} = d(w_{t-1}, w_t | c_t)$,

that can 1-represent G.

Theorem (limited context):

Let $\mathrm{c_t} = (w_0 w_1 w_2 \dots w_{t+L'})$.

$\forall \epsilon > 0, \exists$ PCFG G in Chomsky normal form,

such that any syntactic distance measure

$d_\mathrm{t} = d(w_{t-1}, w_t | c_t)$,

cannot ϵ-represent G.

Transition-based parsing

More results: shift-reduce based parsing

- At each step, instead of a number (syntactic distance)
- We can output parsing transitions
- A stack (initialized to empty)
- An input buffer (initialized with the input sentence)
- At each position t, based on the context at t, the parser outputs a sequence of parsing transitions
- Similar results hold for shift reduce parsers

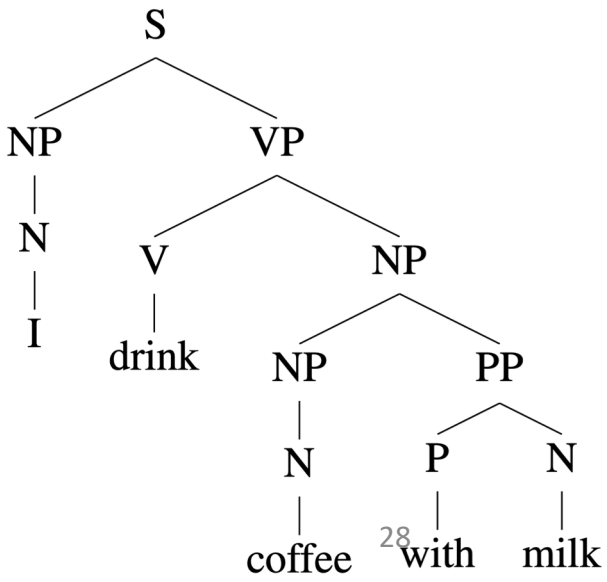
Parsing transitions

- NT(X) pushes a non-terminal X onto the stack.
- SHIFT: removes the first terminal from the input buffer and pushes onto the stack.
- REDUCE: form a new constituent based on the items on the stack

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars.

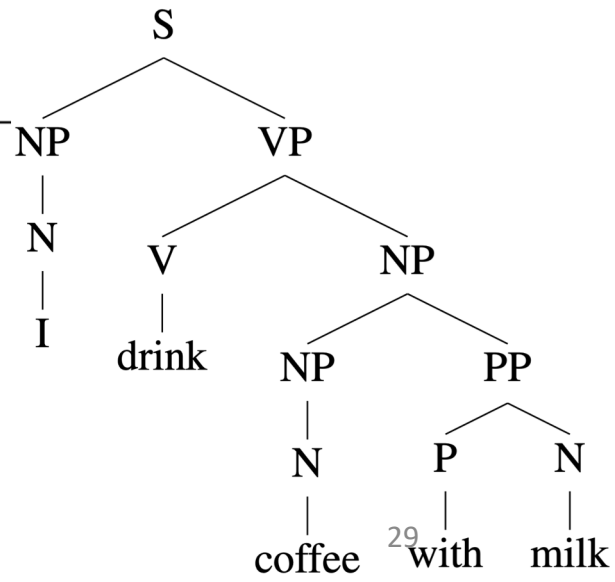
Transition-based parsing: example

Stack	Buffer	Action
	I drink coffee with milk	NT(S)
(S	*I drink coffee with milk*	NT(NP)
(S | (NP	*I drink coffee with milk*	NT(N)
(S | (NP | (N	*I drink coffee with milk*	SHIFT
(S | (NP | (N | *I*	*drink coffee with milk*	REDUCE
(S | (NP (N *I*))	*drink coffee with milk*	NT(VP)
(S | (NP (N *I*)) | (VP	*drink coffee with milk*	NT(V)
(S | (NP (N *I*)) | (VP | (V	*drink coffee with milk*	SHIFT
(S | (NP (N *I*)) | (VP | (V | *drink*	*coffee with milk*	REDUCE
(S | (NP (N *I*)) | (VP | (V *drink*)	*coffee with milk*	NT(NP)



Transition-based parsing: example

(S | (NP (N *I*)) | (VP | (V *drink*) | (NP | (NP (N *coffee*)) | (PP | (P *with*)	*milk*	NT (N)
(S | (NP (N *I*)) | (VP | (V *drink*) | (NP | (NP (N *coffee*)) | (PP | (P *with*) | (N	*milk*	SHIFT
(S | (NP (N *I*)) | (VP | (V *drink*) | (NP | (NP (N *coffee*)) | (PP | (P *with*) | (N | *milk*		REDUCE
(S (NP (N *I*)) (VP (V *drink*) (NP (NP (N *coffee*)) (PP (P *with*) (N *milk*)))))		

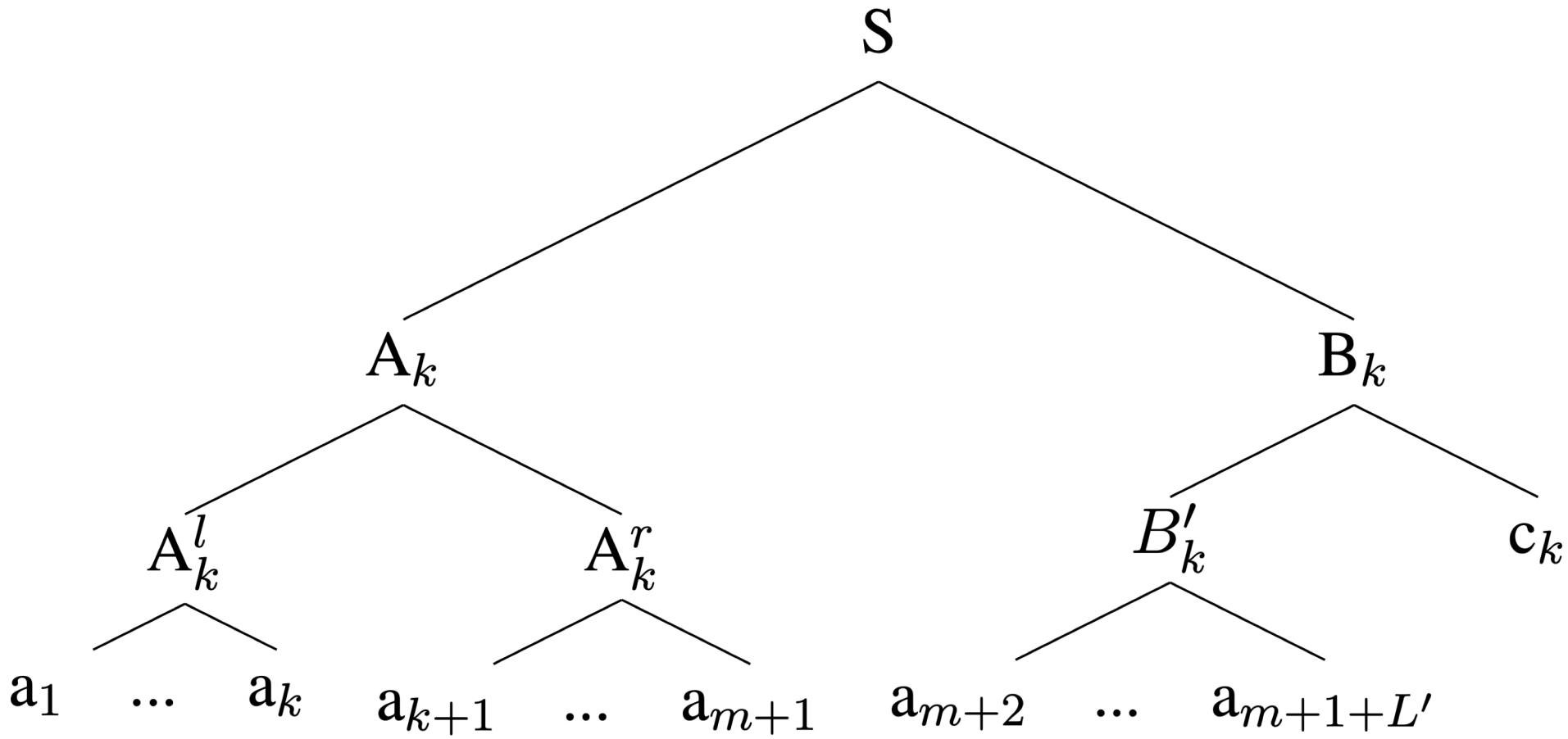


Theorems for transition-based parsing

- Similar to those for syntactic distance
- Positive results
  - With full context, comes full representational power
- Negative results:
  - If context is bounded in one (or both) direction(s)
  - Then there exists a PCFG such that the parsing accuracy can be made arbitrarily low

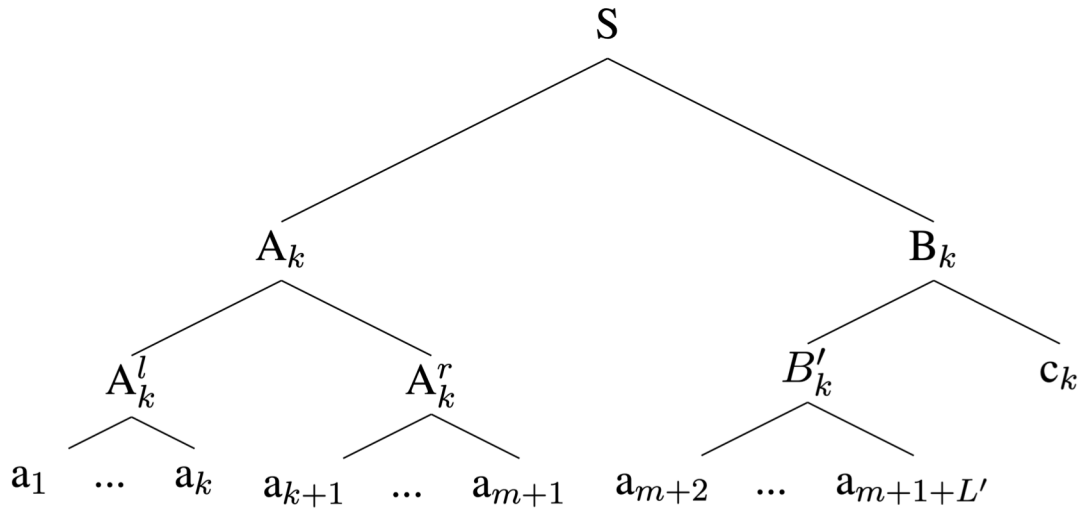
Brief discussion: bidirectional models

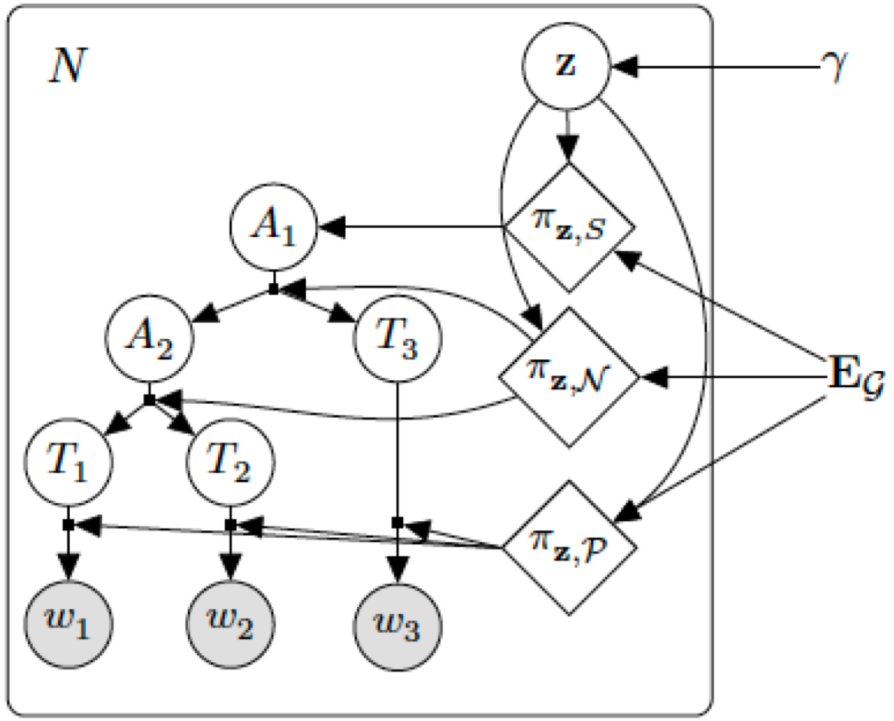
Recall: proof ideas - counterexample grammar



Are there any models that escape this limitation?

- Yes, e.g. Compound PCFG[1]
- In the process of constructing a tree given a sentence
- Has information from both left and right





1. Yoon Kim, Chris Dyer, and Alexander Rush. Compound probabilistic context-free grammars for grammar induction. ACL 2019

Summary of results

Amount of context is a key factor for representational power

Flavor of main result:

For left-to-right parsing approaches, there exists a family of PCFGs, such that their parsing accuracy is arbitrarily bad.

But with full context, they can (in theory) possibly work perfectly, i.e. full representational power.

Takeaways

- Formalized some theoretical advantage of bidirectional context over unidirectional context for parsing
- You can use PCFG as a sandbox to estimate the representational power when proposing a new model
- Paper
  - The Limitations of Limited Context for Constituency Parsing
  - https://arxiv.org/abs/2106.01580
  - Also in this year's Association for Computational Linguistics (ACL-IJCNLP 2021)