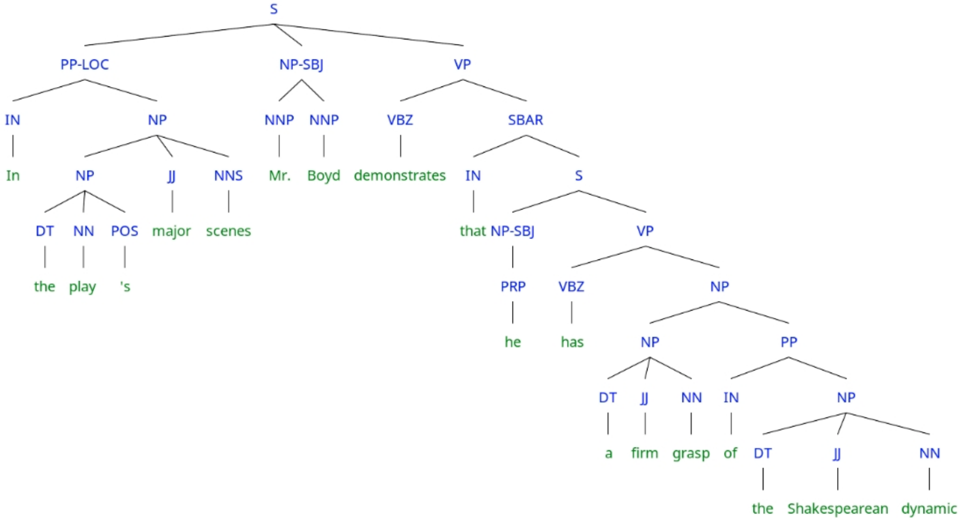
Deep Unsupervised Learning of Syntactic Structure

Yoon Kim

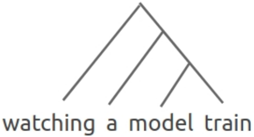
(work with Chris Dyer, Alexander Rush)

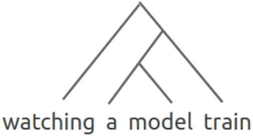
Language has structure



Language has structure

Language has structure



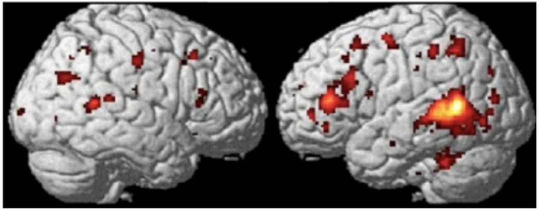


Neurobiological Evidence (Fedorenko et al. 2012)

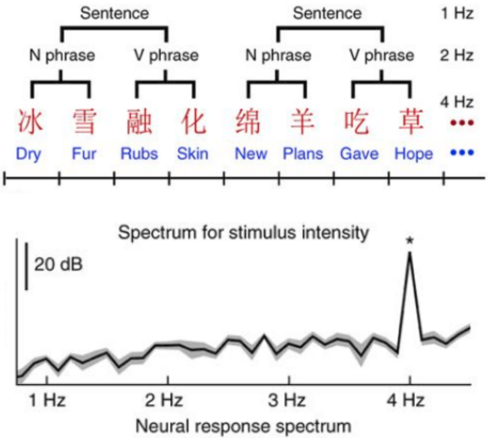
Different neural activity for Jabberwocky sentences versus non-word lists

"after the bonter mellvered the perlen he mested to weer on colmition"

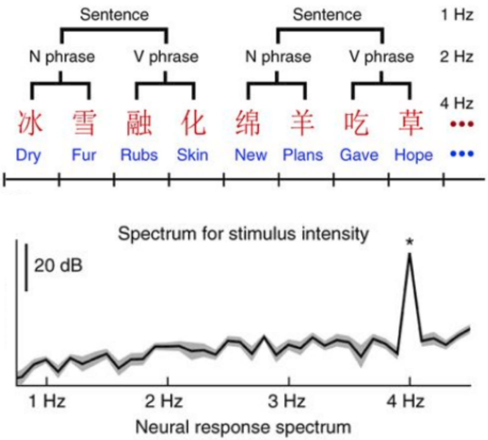
"was during cusarists fick prell pront the pome villpa and wornetist she"

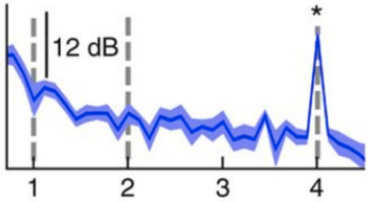


Neurobiological Evidence (Ding et al. 2015)

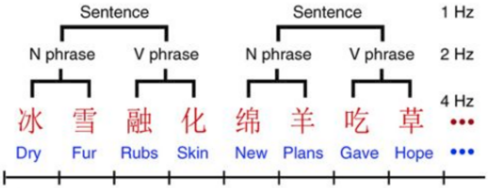


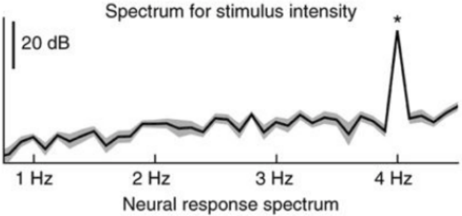
Neurobiological Evidence (Ding et al. 2015)

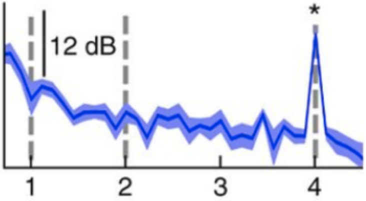


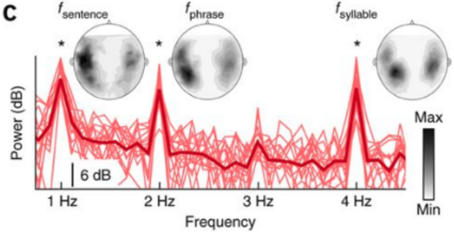


Neurobiological Evidence (Ding et al. 2015)

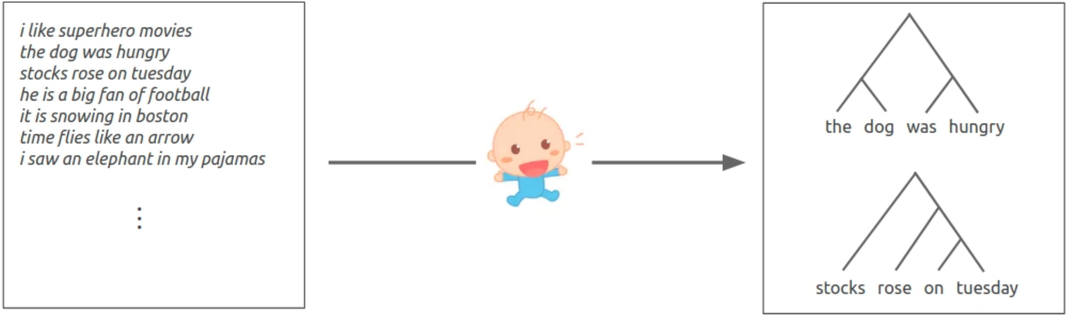




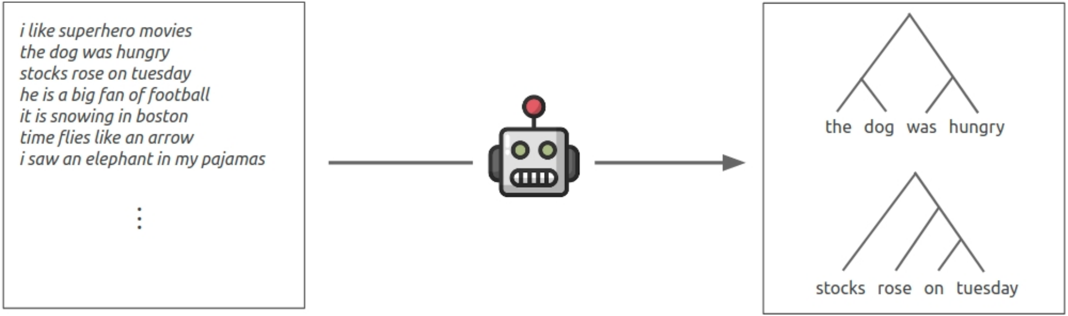




Unsupervised Parsing

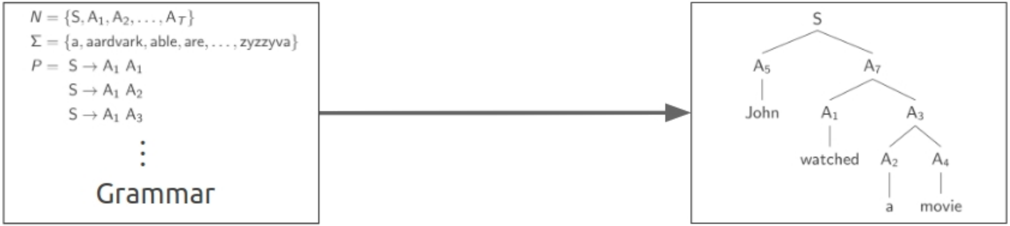


Unsupervised Parsing



Grammar Induction for Unsupervised Parsing

- Classic approach: Hypothesize a **formal grammar** that generates natural language



- (Parse tree implied by the grammar)

Goal of Grammar Induction

- Learning the syntax of human language
- Longstanding problem in AI/NLP

Review: Context-Free Grammars (CFG) for Natural Language

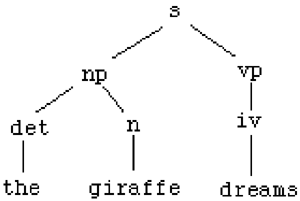
```
s    --> np vp

np   --> det n
vp   --> tv np
     --> iv

det  --> the
     --> a
     --> an

n    --> giraffe
     --> apple

iv   --> dreams
tv   --> eats
     --> dreams
```



Review: CFG Formal Description

$\mathcal{G} = (S, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$ where

$\mathcal{N}$: Set of nonterminals (constituent labels)

$\mathcal{P}$: Set of preterminals (part-of-speech tags)

Σ : Set of terminals (words)

S : Start symbol

$\mathcal{R}$: Set of rules

Each rule $r \in \mathcal{R}$ is one of the following:

$$S \rightarrow A \qquad A \in \mathcal{N}$$

$$A \rightarrow B\ C \qquad A \in \mathcal{N}, \quad B, C \in \mathcal{N} \cup \mathcal{P}$$

$$T \rightarrow w \qquad T \in \mathcal{P}, \quad w \in \Sigma$$

Review: CFG Formal Description

$\mathcal{G} = (S, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$ where

$\mathcal{N}$: Set of nonterminals (constituent labels)

$\mathcal{P}$: Set of preterminals (part-of-speech tags)

Σ : Set of terminals (words)

S : Start symbol

$\mathcal{R}$: Set of rules

Each rule $r \in \mathcal{R}$ is one of the following:

$$S \to A \qquad A \in \mathcal{N}$$

$$A \to B\ C \qquad A \in \mathcal{N}, \quad B, C \in \mathcal{N} \cup \mathcal{P}$$

$$T \to w \qquad T \in \mathcal{P}, \quad w \in \Sigma$$

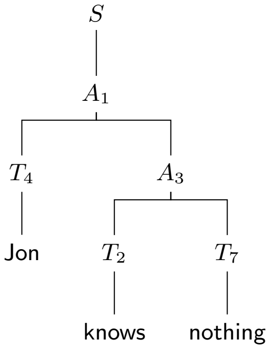
Review: Probabilistic Context-Free Grammars (PCFG)

- Associate probabilities $\boldsymbol{\pi} = \{\pi_r\}_{r \in \mathcal{R}}$ for each rule $r \in \mathcal{R}$.
- Probability of a tree $\boldsymbol{t}$ is given by multiplying the probabilities of rules used in the derivation

$$p_{\boldsymbol{\pi}}(\boldsymbol{t}) = \prod_{r \in \boldsymbol{t}_{\mathcal{R}}} \pi_r$$

where $\boldsymbol{t}_{\mathcal{R}}$ is set of rules used to derive $\boldsymbol{t}$

Review: PCFG Example



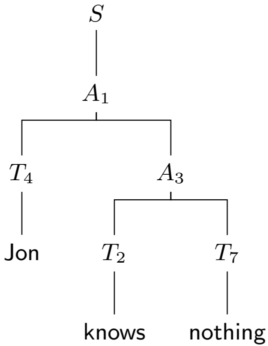
A_i: nonterminals

T_j: preterminals

$$\boldsymbol{t}_{\mathcal{R}} = \{S \to A_1,\ A_1 \to T_4\, A_3,\\ A_3 \to T_2\, T_7,\ T_4 \to \text{Jon},\\ T_2 \to \text{knows},\ T_7 \to \text{nothing}\}$$

$$p_{\boldsymbol{\pi}}(\boldsymbol{t}) = \pi_{S \to A_1} \times \pi_{A_1 \to T_4\, A_3} \times \pi_{A_3 \to T_2\, T_7} \times \\ \pi_{T_4 \to \text{Jon}} \times \pi_{T_2 \to \text{knows}} \times \pi_{T_7 \to \text{nothing}}$$

Review: PCFG Example



A_i: nonterminals

T_j: preterminals

$$\boldsymbol{t}_{\mathcal{R}} = \{S \to A_1,\ A_1 \to T_4\, A_3,\ A_3 \to T_2\, T_7,\ T_4 \to \text{Jon},\ T_2 \to \text{knows},\ T_7 \to \text{nothing}\}$$

$$p_{\boldsymbol{\pi}}(\boldsymbol{t}) = \pi_{S \to A_1} \times \pi_{A_1 \to T_4\, A_3} \times \pi_{A_3 \to T_2\, T_7} \times \pi_{T_4 \to \text{Jon}} \times \pi_{T_2 \to \text{knows}} \times \pi_{T_7 \to \text{nothing}}$$

Review: Grammar Induction with PCFGs

- Specify broad grammar structure: number of nonterminals ($|\mathcal{N}| = 30$), preterminals ($|\mathcal{P}| = 60$), set of context-free rules
- Maximize log likelihood (Expectation-Maximization)
  - Given corpus of sentences $\mathbf{x}^{(1)}, \dots \mathbf{x}^{(N)}$,

$$\max_{\boldsymbol{\pi}} \sum_{n=1}^{N} \log p_{\boldsymbol{\pi}}(\mathbf{x}^{(n)})$$

  - Sum over unobserved trees,

$$p_{\boldsymbol{\pi}}(\mathbf{x}) = \sum_{\boldsymbol{t} \in \mathcal{T}(\mathbf{x})} p_{\boldsymbol{\pi}}(\boldsymbol{t})$$

    where $\mathcal{T}(\mathbf{x})$ =set of trees whose leaves are $\mathbf{x}$.

Results from PCFG Induction

Unlabeled F_1 against gold trees on PTB.

Model	F_1
Random Trees	19.5
PCFG	35.0

Results from PCFG Induction

Unlabeled F_1 against gold trees on PTB.

Model	F_1
Random Trees	19.5
PCFG	35.0
Right Branching	39.5

Long history of work showing that MLE with PCFGs fails to discover linguistically meaningful tree structures [Lari and Young 1990].

Common wisdom: "MLE with PCFGs doesn't work"

Rich Prior Work on Unsupervised Constituency Parsing

- Modified objectives [Klein and Manning 2002, 2004; Smith and Eisner 2004].
- Use priors/nonparametric models [Liang et al. 2007; Johnson et al. 2007].
- Handcrafted features [Huang et al. 2012; Golland et al. 2012].
- Other types of regularization (e.g. on recursion depth) [Noji et al. 2016; Jin et al. 2018].
- Activation analysis from neural language models [Shen et al. 2018, 2019]

This Talk: Revisit Core Assumptions about Grammar Induction

1. PCFG with an embedding parameterization can induce meaningful grammars with MLE.
2. Develop more flexible grammars through auxiliary sentence vector + neural variational inference.
3. Learn structured language models with induced trees.

This Talk: Revisit Core Assumptions about Grammar Induction

1. **PCFG with an embedding parameterization can induce meaningful grammars with MLE.**
2. Develop more flexible grammars through auxiliary sentence vector + neural variational inference.
3. Learn structured language models with induced trees.

Simple Modification: PCFG Parameterization

- **Scalar Parameterization**: Associate probabilities π_r to each rule such that they are valid probability distributions.

$$\pi_{T\to w} \geq 0 \qquad\qquad \sum_{w'\in\Sigma} \pi_{T\to w'} = 1$$

- **"Neural" Parameterization**: Associate symbol embeddings $\mathbf{w}_N$ to each symbol N on left hand side of a rule.

$$\pi_{T\to w} = \textsc{NeuralNet}(\mathbf{w}_T) = \frac{\exp(\mathbf{u}_w^\top f(\mathbf{w}_T))}{\sum_{w'\in\Sigma} \exp(\mathbf{u}_{w'}^\top f(\mathbf{w}_T))}$$

(Similar parameterizations for $A \to BC$)

Simple Modification: PCFG Parameterization

- **Scalar Parameterization**: Associate probabilities π_r to each rule such that they are valid probability distributions.

$$\pi_{T \to w} \geq 0 \qquad\qquad \sum_{w' \in \Sigma} \pi_{T \to w'} = 1$$

- **"Neural" Parameterization**: Associate symbol embeddings $\mathbf{w}_N$ to each symbol N on left hand side of a rule.

$$\pi_{T \to w} = \textsc{NeuralNet}(\mathbf{w}_T) = \frac{\exp(\mathbf{u}_w^\top f(\mathbf{w}_T))}{\sum_{w' \in \Sigma} \exp(\mathbf{u}_{w'}^\top f(\mathbf{w}_T))}$$

(Similar parameterizations for $A \to BC$)

Simple Modification: Neural PCFG

$$\pi_{T \to w} \propto \exp\left(\underbrace{\mathbf{u}_w^\top}_{\text{output emb.}} \overbrace{f(\underbrace{\mathbf{w}_T}_{\text{input emb.}})}^{\text{shared neural net}} \right)$$

- Model parameters θ given by input embeddings, output embeddings, and parameters of neural net f.
- Analogous to count-based vs neural language models: parameter sharing through distributed representations (word embedding vs symbol embedding).

Same model assumptions, different parameterization.

Simple Modification: Neural PCFG

$$\pi_{T\to w} \propto \exp\Big(\underbrace{\mathbf{u}_w^\top}_{\text{output emb.}} \overbrace{f(\underbrace{\mathbf{w}_T}_{\text{input emb.}})}^{\text{shared neural net}} \Big)$$

- Model parameters θ given by input embeddings, output embeddings, and parameters of neural net f.
- Analogous to count-based vs neural language models: parameter sharing through distributed representations (word embedding vs symbol embedding).

Same model assumptions, different parameterization.

Neural PCFG: Training

- Maximum likelihood (EM) with dynamic programming for marginalization.
- Practical details: Stochastic gradient ascent on log marginal likelihood with Inside algorithm + Autodiff

$$\theta_{\text{new}} = \theta_{\text{old}} + \lambda \nabla_\theta \underbrace{\log p_\theta(\mathbf{x})}_{\text{inside algorithm}}$$

- (**PyTorch-Struct** includes GPU-optimized implementations of these (and many other) algorithms.)

Neural PCFG: Results

Model	F_1
Random Trees	19.5
Right Branching	39.5
Scalar PCFG	35.0

(English Penn Treebank)

Neural PCFG: Results

Model	F_1
Random Trees	19.5
Right Branching	39.5
Scalar PCFG	35.0
Neural PCFG	52.6

(English Penn Treebank)

Neural PCFG Results

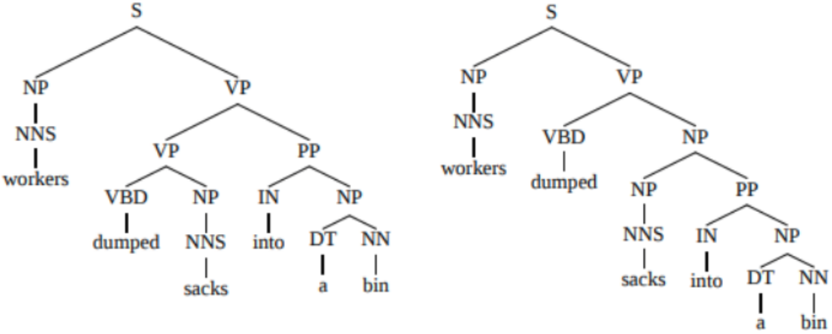
Model	F_1	Training/Test PPL
Random Trees	19.5	–
Right Branching	39.5	–
Scalar PCFG	35.0	≈ 350
Neural PCFG	52.6	≈ 250

This Talk: Revisit Core Assumptions about Grammar Induction

1. PCFG with an embedding parameterization can induce meaningful grammars with MLE.
2. **Develop more flexible grammars through auxiliary sentence vector + neural variational inference.**
3. Learn structured language models with induced trees.

Review: Limitations of simple PCFGs

No sensitivity to lexical context



(example from `http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/lexpcfgs.pdf`)

Review: Limitations of simple PCFGs

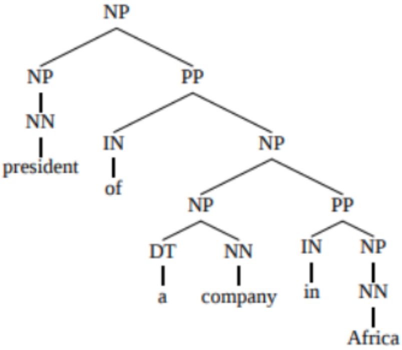
No sensitivity to lexical context

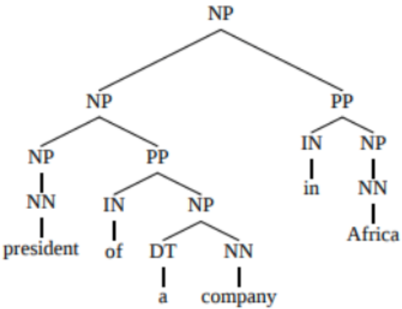
Rules
S $\rightarrow$ NP VP
NP $\rightarrow$ NNS
VP $\rightarrow$ VP PP
VP $\rightarrow$ VBD NP
NP $\rightarrow$ NNS
PP $\rightarrow$ IN NP
NP $\rightarrow$ DT NN
NNS $\rightarrow$ workers
VBD $\rightarrow$ dumped
NNS $\rightarrow$ sacks
IN $\rightarrow$ into
DT $\rightarrow$ a
NN $\rightarrow$ bin

Rules
S $\rightarrow$ NP VP
NP $\rightarrow$ NNS
NP $\rightarrow$ NP PP
VP $\rightarrow$ VBD NP
NP $\rightarrow$ NNS
PP $\rightarrow$ IN NP
NP $\rightarrow$ DT NN
NNS $\rightarrow$ workers
VBD $\rightarrow$ dumped
NNS $\rightarrow$ sacks
IN $\rightarrow$ into
DT $\rightarrow$ a
NN $\rightarrow$ bin

Review: Limitations of simple PCFGs

No sensitivity to structural context





(example from http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/lexpcfgs.pdf)

Review: Limitations of simple PCFGs

Johnson et al. [2007]: Supervised PCFG + Unsupervised fine tuning decreases parsing accuracy while corpus likelihood improves!

"It is easy to demonstrate that the poor quality of the PCFG models is the cause of these problems rather than search or other algorithmic issues. If one initializes either the IO or Bayesian estimation procedures with treebank parses and then runs the procedure using the yields alone, the accuracy of the parses uniformly decreases while the (posterior) likelihood uniformly increases with each iteration, demonstrating that improving the (posterior) likelihood of such models does not improve parse accuracy."

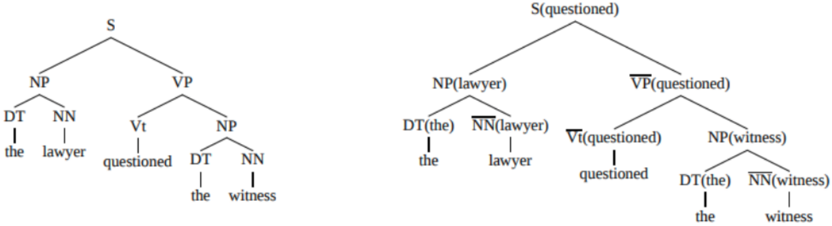
Classic Solutions: Lexicalization

- No sensitivity to lexical context $\Longrightarrow$ Lexicalized PCFGs [Collins 1997]
- Rules are lexicalized, e.g.

$$A \to BC \implies A(w) \to B(w)C(h)$$

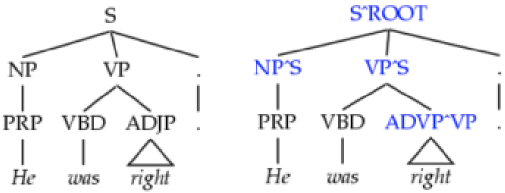
$w, h \in \Sigma$

- Integrates notion of headedness



Classic Solutions: Higher-order Grammars

- No sensitivity to structural context $\Longrightarrow$ Horizontal/Vertical Markovization [Klein and Manning 2003]
- Richer dependencies through grandparents/siblings.



Classic Solutions: Enriching PCFGs

- Lexicalized PCFG [Collins 1997]
- Horizontal/Vertical Markovization [Klein and Manning 2003]
- Latent Variable PCFG [Petrov et al. 2006]

Expensive to apply in the unsupervised case due to explosion in number of rules.

Compound PCFG

- Goal: Capture these in a soft manner.
- **Compound** generative process (Bayesian PCFG):

$$(1)\ \ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$(2)\ \ \boldsymbol{\pi}_{\mathbf{z}} = \textsc{NeuralNetwork}([\mathbf{w}_N; \mathbf{z}]),\ \text{for example,}$$

$$\pi_{\mathbf{z}, T \to w} = \frac{\exp(\mathbf{u}_w^\top f([\mathbf{w}_T; \mathbf{z}]))}{\sum_{w' \in \Sigma} \exp(\mathbf{u}_{w'}^\top f([\mathbf{w}_T; \mathbf{z}]))}$$

$$(3)\ \ t \sim \textsc{pcfg}(\boldsymbol{\pi}_{\mathbf{z}})$$

$$(4)\ \ \mathbf{x} = \text{yield}(t)$$

Compound PCFG

- Goal: Capture these in a soft manner.
- **Compound** generative process (Bayesian PCFG):

$$(1)\ \ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$(2)\ \ \boldsymbol{\pi}_{\mathbf{z}} = \textsc{NeuralNetwork}([\mathbf{w}_N;\, \mathbf{z}]),\ \text{for example,}$$

$$\boldsymbol{\pi}_{\mathbf{z}, T \to w} = \frac{\exp(\mathbf{u}_w^\top f([\mathbf{w}_T;\, \mathbf{z}]))}{\sum_{w' \in \Sigma} \exp(\mathbf{u}_{w'}^\top f([\mathbf{w}_T;\, \mathbf{z}]))}$$

$$(3)\ \ \boldsymbol{t} \sim \textsc{pcfg}(\boldsymbol{\pi}_{\mathbf{z}})$$

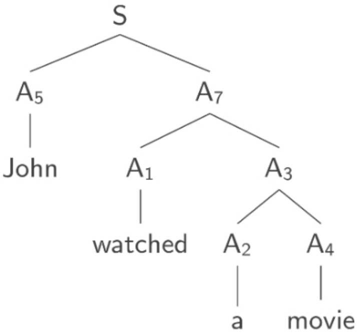
$$(4)\ \ \mathbf{x} = \text{yield}(\boldsymbol{t})$$

Compound PCFG

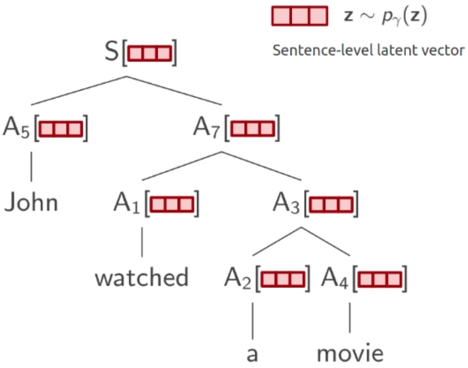
$$\boldsymbol{\pi}_{\mathbf{z},T\to w} \propto \exp(\underbrace{\mathbf{u}_w^\top f([\mathbf{w}_T}_{\text{fixed across sents}};\overbrace{\mathbf{z}}^{\text{varies}}]))$$

- Input/output embeddings and neural net f shared across sentences, but rule probabilities for each sentence can vary through $\mathbf{z}$
- Intuition: $\mathbf{z}$ can encode lexical/structural information specific to the sentence.

Neural PCFG vs. Compound PCFG



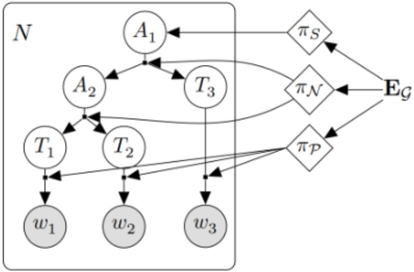
Neural PCFG

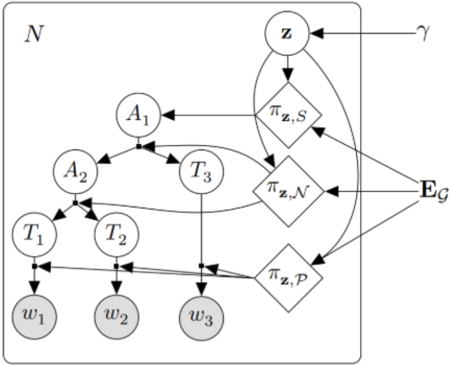


Compound PCFG

Neural PCFG vs. Compound PCFG

The model reduces to a PCFG conditioned on $\mathbf{z}$





Compound PCFG: Training and Inference

For maximum likelihood, log marginal likelihood given by

$$\log p_\theta(\mathbf{x}) = \log \left(\int \underbrace{\sum_{\boldsymbol{t} \in \mathcal{T}(\mathbf{x})} p_\theta(\boldsymbol{t} \mid \mathbf{z})}_{p_\theta(\mathbf{x} \mid \mathbf{z})} p(\mathbf{z}) \, \mathrm{d}\mathbf{z} \right)$$

- Intractable due to integral over $\mathbf{z}$.

Compound PCFG: Training and Inference

Variational Inference: Introduce variational posterior for $\mathbf{z}$

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x})}\Big[\log \underbrace{\sum_{\boldsymbol{t} \in \mathcal{T}(\mathbf{x})} p_\theta(\boldsymbol{t} \mid \mathbf{z})}_{p_\theta(\mathbf{x} \mid \mathbf{z})}\Big] - \mathrm{KL}[\, q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\,]$$

- Inference network over $\mathbf{x}$ produces parameters for the Gaussian variational posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$.
- Given a sample $\mathbf{z}$, can calculate with dynamic programming

$$p_\theta(\mathbf{x} \mid \mathbf{z}) = \sum_{\boldsymbol{t} \in \mathcal{T}(\mathbf{x})} p_\theta(\boldsymbol{t} \mid \mathbf{z})$$

Compound PCFG: Training and Inference

Variational Inference: Introduce variational posterior for $\mathbf{z}$

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x})}\Big[\log \underbrace{\sum_{\boldsymbol{t} \in \mathcal{T}(\mathbf{x})} p_\theta(\boldsymbol{t} \mid \mathbf{z})}_{p_\theta(\mathbf{x} \mid \mathbf{z})}\Big] - \mathrm{KL}[\, q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z}) \,]$$

- Inference network over $\mathbf{x}$ produces parameters for the Gaussian variational posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$.
- Given a sample $\mathbf{z}$, can calculate with dynamic programming

$$p_\theta(\mathbf{x} \mid \mathbf{z}) = \sum_{\boldsymbol{t} \in \mathcal{T}(\mathbf{x})} p_\theta(\boldsymbol{t} \mid \mathbf{z})$$

Compound PCFG: Training and Inference

Collapsed Variational Inference

$$\log p_\theta(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}\,|\,\mathbf{x})}}_{\text{reparameterized sample}} \Big[\, \underbrace{\log p_\theta(\mathbf{x}\,|\,\mathbf{z})}_{\text{inside algorithm}} \,\Big] - \underbrace{\mathrm{KL}[\, q_\phi(\mathbf{z}\,|\,\mathbf{x}) \,\|\, p(\mathbf{z}) \,]}_{\text{analytic KL between 2 Gaussians}}$$

"VAE with a PCFG decoder"

Compound PCFG: Training and Inference

Collapsed Variational Inference

$$\log p_\theta(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}\,|\,\mathbf{x})}}_{\text{reparameterized sample}} \left[\underbrace{\log p_\theta(\mathbf{x}\,|\,\mathbf{z})}_{\text{inside algorithm}} \right] - \underbrace{\mathrm{KL}[\, q_\phi(\mathbf{z}\,|\,\mathbf{x}) \,\|\, p(\mathbf{z}) \,]}_{\text{analytic KL between 2 Gaussians}}$$

"VAE with a PCFG decoder"

Compound PCFG: Results on PTB

Model	F_1	Training/Test PPL
Random Trees	19.5	–
Right Branching	39.5	–
Scalar PCFG	35.0	≈ 350
Neural PCFG	52.6	≈ 250
Compound PCFG	60.1	≈ 190

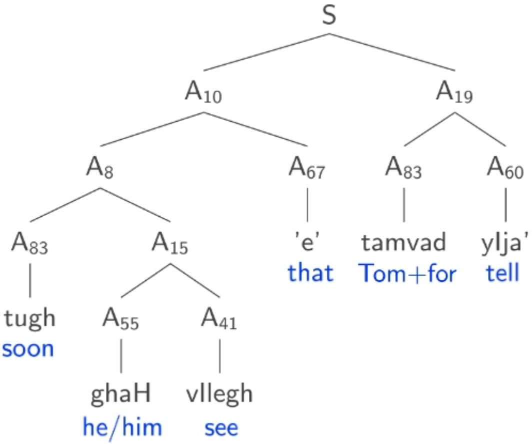
Compound PCFG: Comparison against other unsupervised parsers

Model	English (PTB)
PRPN [Shen et al. 2018]	38.1
Ordered Neurons [Shen et al. 2019]	49.4
DIORA [Drozdov et al. 2019]	58.9
Constituency Tests [Cao et al. 2020]	62.8
Right Branching	39.5
Scalar PCFG	35.0
Neural PCFG	52.6
Compound PCFG	60.1

Compound PCFG: Results on other languages

Model	English	Chinese	Japanese
Random Trees	19.5	16.0	15.3
Left Branching	8.7	9.7	25.5
Right Branching	39.5	20.0	1.2
Scalar PCFG	35.0	15.0	15.7
Neural PCFG	52.6	29.5	44.6
Compound PCFG	60.1	39.8	47.4

Parsing Klingon

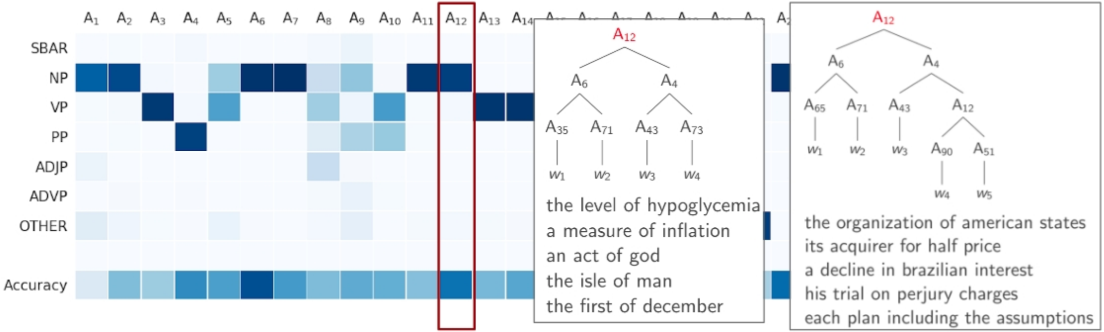


tugh ghaH vllegh 'e' tamvaD ylja'

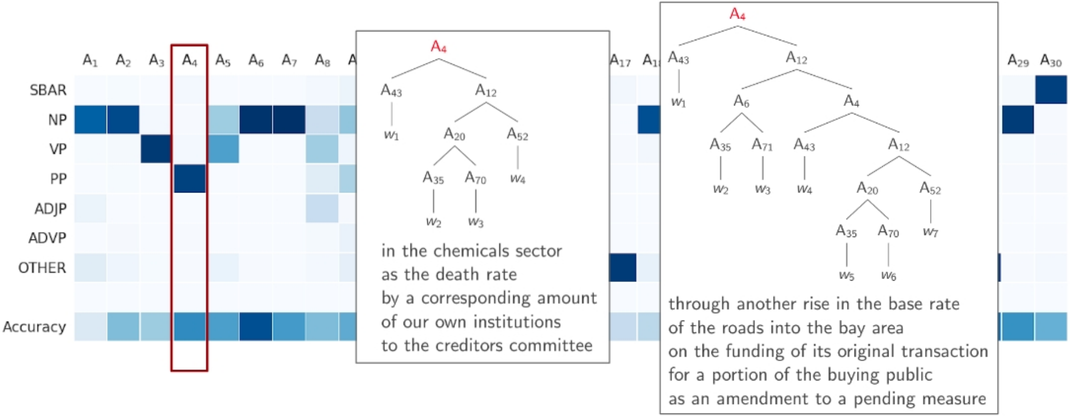
tell Tom that I will see him soon

Model Analysis: Nonterminal Alignment ($|\mathcal{N}| = 30$)

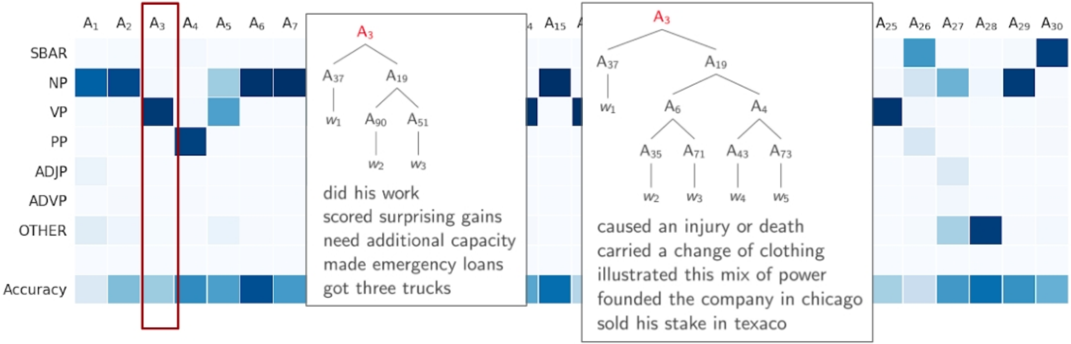
Model Analysis: Nonterminal Alignment ($|\mathcal{N}| = 30$)



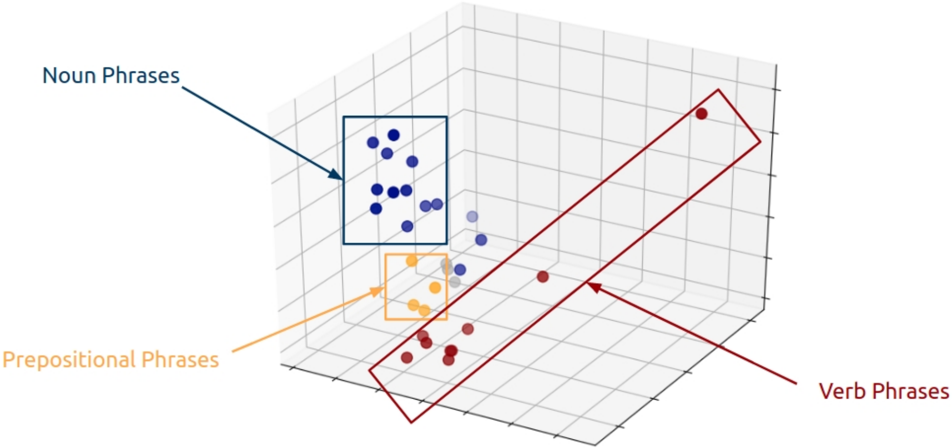
Model Analysis: Nonterminal Alignment ($|\mathcal{N}| = 30$)



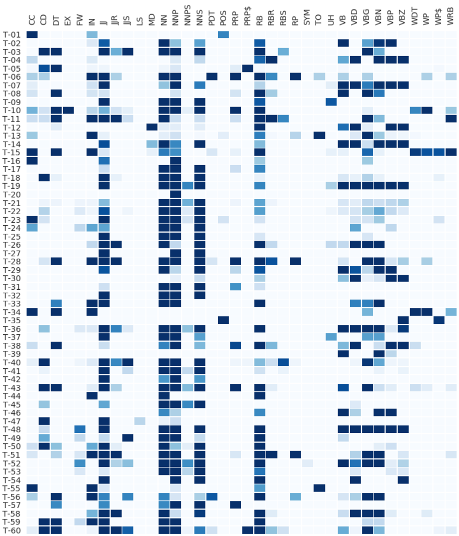
Model Analysis: Nonterminal Alignment ($|\mathcal{N}| = 30$)



Model Analysis: Nonterminal Alignment ($|\mathcal{N}| = 30$)



Model Analysis: Preterminal Alignment ($|\mathcal{P}| = 60$)

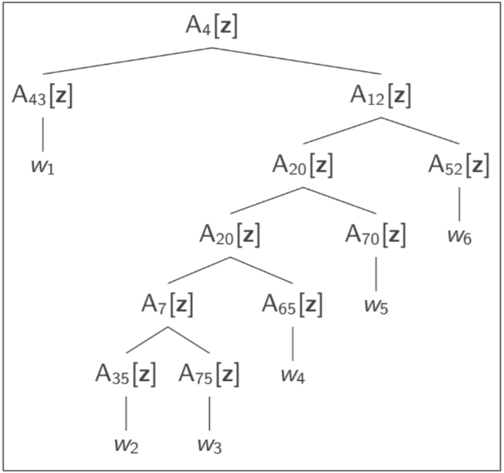


Model Analysis: What does $\mathbf{z}$ learn?

Nearest neighbors based on variational posterior mean vector

$\langle$unk$\rangle$ corp. received an N million army contract for helicopter engines
boeing co. received a N million air force contract for developing cable systems for the $\langle$unk$\rangle$ missile
general dynamics corp. received a N million air force contract for $\langle$unk$\rangle$ training sets
grumman corp. received an N million navy contract to upgrade aircraft electronics
thomson missile products with about half british aerospace 's annual revenue include the $\langle$unk$\rangle$ $\langle$unk$\rangle$ missile fa
already british aerospace and french $\langle$unk$\rangle$ $\langle$unk$\rangle$ $\langle$unk$\rangle$ on a british missile contract and on an air-traffic control

Model Analysis: What does $\mathbf{z}$ learn?



Cluster 1

of the company 's capital structure
in the company 's divestiture program
by the company 's new board
in the company 's core business

Cluster 2

above the treasury 's N-year note
above the treasury 's seven-year note
above the treasury 's comparable note
above the treasury 's five-year note

1st Principal Component of **z**

This Talk: Revisit Core Assumptions about Grammar Induction

1. PCFG with an embedding parameterization can induce meaningful grammars with MLE.
2. Develop more flexible grammars through auxiliary sentence vector + neural variational inference.
3. **Learn structured language models with induced trees.**

Compound PCFG as a Language Model

Model	F_1	Test PPL
Scalar PCFG	35.0	≈ 350
Neural PCFG	52.6	≈ 250
Compound PCFG	60.1	≈ 190

Compound PCFG as a Language Model

Model	F_1	Test PPL
Scalar PCFG	35.0	≈ 350
Neural PCFG	52.6	≈ 250
Compound PCFG	60.1	≈ 190
RNN LM	–	86.2

Good parser, poor language model.

Review: Recurrent Neural Network Grammars (RNNG) [Dyer et al. 2016]

- Structured joint generative model of sentence $\mathbf{x}$ and tree $\mathbf{z}$

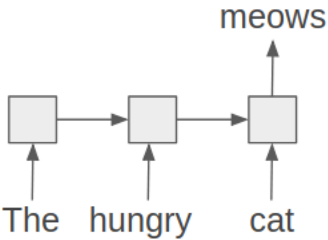
$$p_\theta(\mathbf{x}, \mathbf{z})$$

- Generate next word conditioned on partially-completed syntax tree
- Like RNN LM, no independence assumptions.

Review: RNN LMs

"Flat" left-to-right generation

$$x_t \sim p_\theta(x \mid x_1, \ldots, x_{t-1}) = \text{softmax}(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{b})$$



RNNG [Dyer et al. 2016]

Introduce binary variables $\mathbf{z} = [z_1, \ldots, z_{2T-1}]$ (unlabeled binary tree)

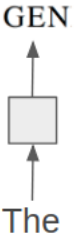
Sample action $z_t \in \{\textsc{generate}, \textsc{reduce}\}$ at each time step:

$$z_t \sim \text{Bernoulli}(p_t) \qquad\qquad p_t = \sigma(\mathbf{w}^\top \mathbf{h}_{\text{prev}} + b)$$

RNNG [Dyer et al. 2016]

If $z_t = \textsc{generate}$

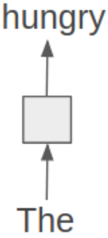
Sample word from context representation



RNNG [Dyer et al. 2016]

(Similar to standard RNNLMs)

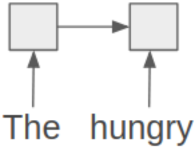
$$x \sim \text{softmax}(\mathbf{W}\mathbf{h}_{\text{prev}} + \mathbf{b})$$



RNNG [Dyer et al. 2016]

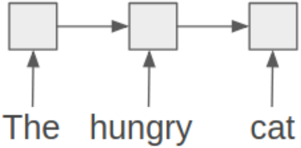
Obtain new context representation with $\mathbf{e}_{\text{hungry}}$

$$\mathbf{h}_{\text{new}} = \text{LSTM}(\mathbf{e}_{\text{hungry}}, \mathbf{h}_{\text{prev}})$$



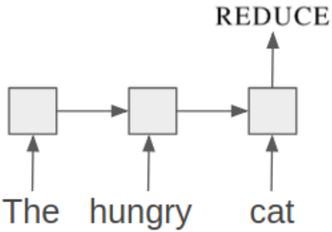
RNNG [Dyer et al. 2016]

$$\mathbf{h}_{\text{new}} = \text{LSTM}(\mathbf{e}_{\text{cat}}, \mathbf{h}_{\text{prev}})$$



RNNG [Dyer et al. 2016]

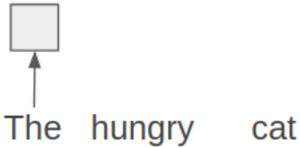
If $z_t = \text{REDUCE}$



RNNG [Dyer et al. 2016]

If $z_t = \text{REDUCE}$

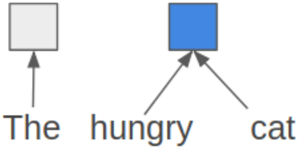
Pop last two elements



RNNG [Dyer et al. 2016]

Obtain new representation of constituent

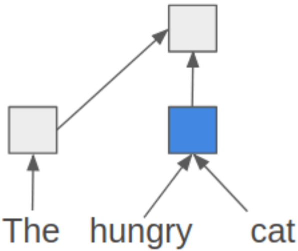
$$\mathbf{e}_{(\text{hungry cat})} = \text{TreeLSTM}(\mathbf{e}_{\text{hungry}}, \mathbf{e}_{\text{cat}})$$



RNNG [Dyer et al. 2016]

Move the new representation onto the stack

$$\mathbf{h}_{\text{new}} = \text{LSTM}(\mathbf{e}_{(\text{hungry cat})}, \mathbf{h}_{\text{prev}})$$



Compound PCFG + RNNG

- Compound PCFG to parse training set, train an RNNG on induced trees, fine-tune with unsupervised RNNG.

Model	Test PPL
Neural PCFG	252.6
Compound PCFG	196.3
RNN LM	86.2
URNNG + Compound PCFG	83.7
URNNG + Gold Trees	78.3

Compound PCFG + RNNG

- Compound PCFG to parse training set, train an RNNG on induced trees, fine-tune with unsupervised RNNG.

Model	Test PPL
Neural PCFG	252.6
Compound PCFG	196.3
RNN LM	86.2
URNNG + Compound PCFG	83.7
URNNG + Gold Trees	78.3

Syntactic Evaluation [Marvin and Linzen 2018]

Two minimally different sentences:

The senators near the assistant are old

*The senators near the assistant is old

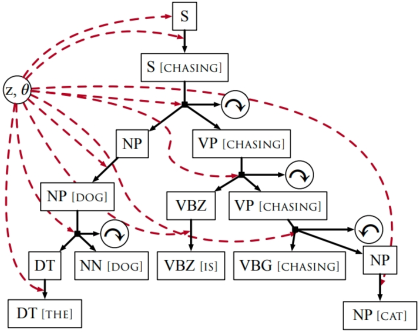
- Model must assign higher probability to the correct one.

Syntactic Evaluation [Marvin and Linzen 2018]

Model	Test PPL	Syntactic Eval.
RNN LM	86.2	60.9%
URNNG + Compound PCFG	83.7	76.1%
URNNG + Gold Trees	78.3	76.1%

Compound PCFG Extensions

- Lexicalized Compound PCFG [Zhu et al. 2020]



- Visually Grounded Compound PCFG [Zhao and Titov 2020]

Discussion

Limitations

- Can be slower to train due to DP.
- Latent vector to approximate richer grammars.

"We assume that the goal of learning a context-free grammar needs no justification."

[Carroll and Charniak 1992]

- What is the role of grammars (and other linguistic structures) in ELMo/BERT era?

Discussion

Limitations

- Can be slower to train due to DP.
- Latent vector to approximate richer grammars.

"We assume that the goal of learning a context-free grammar needs no justification."

[Carroll and Charniak 1992]

- What is the role of grammars (and other linguistic structures) in ELMo/BERT era?

Future Work

- Separation of "what to say" from "how to say it" for structured generation.
- Some languages are provably not context-free $\implies$ neural parameterizations of mildly context-sensitive formalisms (e.g. tree-adjoining grammars).
- Investigate why MLE with scalar parameterization fails but neural parameterization works.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised Parsing with Constituency Tests. In *Proceedings of EMNLP*.

Glenn Carroll and Eugene Charniak. 1992. Two Experiments on Learning Probabilistic Dependency Grammars from Corpora. In *AAAI Workshop on Statistically-Based NLP Techniques*.

Michael Collins. 1997. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of ACL*.

Andrew Drozdov, Patrick Verga, Mohit Yadev, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised Latent Tree Induction with Deep Inside-Outside Recursive Auto-Encoders. In *Proceedings of NAACL*.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of NAACL*.

Dave Golland, John DeNero, and Jakob Uszkoreit. 2012. A Feature-Rich Constituent Context Model for Grammar Induction. In *Proceedings of ACL*.

Yun Huang, Min Zhang, and Chew Lim Tan. 2012. Improved Constituent Context Model with Features. In *Proceedings of PACLIC*.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2018. Unsupervised Grammar Induction with Depth-bounded PCFG. In *Proceedings of TACL*.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian Inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of NAACL*.

Dan Klein and Christopher Manning. 2002. A Generative Constituent-Context Model for Improved Grammar Induction. In *Proceedings of ACL*.

Dan Klein and Christopher Manning. 2004. Corpus-based Induction of Syntactic Structure: Models of Dependency and Constituency. In *Proceedings of ACL*.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of ACL*.

Karim Lari and Steve Young. 1990. The Estimation of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm. *Computer Speech and Language*, 4:35–56.

Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. 2007. The Infinite PCFG using Hierarchical Dirichlet Processes. In *Proceedings of EMNLP*.

Rebecca Marvin and Tal Linzen. 2018. Targeted Syntactic Evaluation of Language Models. In *Proceedings of EMNLP*.

Hiroshi Noji, Yusuke Miyao, and Mark Johnson. 2016. Using Left-corner Parsing to Encode Universal Structural Constraints in Grammar Induction. In *Proceedings of EMNLP*.

Slav Petrov, Leon Barret, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of ACL*.

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

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *Proceedings of ICLR*.

Noah A. Smith and Jason Eisner. 2004. Annealing Techniques for Unsupervised Statistical Language Learning. In *Proceedings of ACL*.

Yanpeng Zhao and Ivan Titov. 2020. Visually Grounded Compound PCFGs. In *Proceedings of EMNLP*.

Hao Zhu, Yonatan Bisk, and Graham Neubig. 2020. The Return of Lexical Dependencies: Neural Lexicalized PCFGs. In *TACL*.