LING5702: Lecture Notes 23
Bayesian grammar induction experiments

Contents

23.1 Probabilistic grammar induction model (Jin et al., 2021)

Try to learn constituent (context-free) grammar from child-directed speech.

If category structure doesn't exist, induced grammars will be arbitrary.

For this we start with a (recursive) Bayesian model:

$$\overbrace{\mathsf{P}(\text{grammar} \mid \text{sentences})}^{\text{posterior distribution}} = \frac{\sum_{\text{trees}} \mathsf{P}(\text{grammar}, \text{trees}, \text{sentences})}{\mathsf{P}(\text{sentences})}$$
$$= \frac{\sum_{\text{trees}} \mathsf{P}(\text{grammar}) \cdot \mathsf{P}(\text{trees} \mid \text{grammar}) \cdot \mathsf{P}(\text{sentences} \mid \text{trees})}{\mathsf{P}(\text{sentences})}$$

- $\mathsf{P}(\text{grammar})$ is a Dirichlet distribution (over categorical distributions).
- $\mathsf{P}(\text{trees} \mid \text{grammar})$ is a categorical distribution, generated recursively (see below).
- $\mathsf{P}(\text{sentences} \mid \text{trees})$ is deterministic (sentences are leaves).

23.2 Grammars as matrices

We first encode a grammar:

$$\begin{aligned}
\mathsf{P}(\mathrm{NP} \to \mathrm{NP\ NP} \mid \mathrm{NP}) &= 0.1 \\
\mathsf{P}(\mathrm{NP} \to \mathrm{NP\ PP} \mid \mathrm{NP}) &= 0.4 \\
\mathsf{P}(\mathrm{PP} \to \mathrm{P\ NP} \mid \mathrm{PP}) &= 0.8 \\
\mathsf{P}(\mathrm{PP} \to \mathrm{P\ PP} \mid \mathrm{PP}) &= 0.2 \\
\mathsf{P}(\mathrm{NP} \to \mathrm{books}\ \bot \mid \mathrm{NP}) &= 0.3 \\
\mathsf{P}(\mathrm{NP} \to \mathrm{stories}\ \bot \mid \mathrm{NP}) &= 0.2 \\
\mathsf{P}(\mathrm{P} \to \mathrm{about}\ \bot \mid \mathrm{P}) &= 1.0
\end{aligned}$$

as a matrix, with rows for parent categories and columns for combinations of left and right child:

$$\begin{array}{c} \\ \mathrm{NP} \\ \mathrm{PP} \\ \mathrm{P} \end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} \mathrm{NP\ NP} & \mathrm{NP\ PP} & \mathrm{NP\ P} & \mathrm{PP\ NP} & \mathrm{PP\ PP} & \mathrm{PP\ P} & \mathrm{P\ NP} & \mathrm{P\ PP} & \mathrm{P\ P} & \mathrm{books}\ \bot & \mathrm{stories}\ \bot & \mathrm{about}\ \bot \end{array} \\
\begin{pmatrix} .1 & .4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & .2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .8 & .2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{array}$$

We index the rows via **Kronecker deltas**, where only the indexed element is non-zero:

$$\delta_{\mathrm{PP}} = \begin{array}{c} \\ \mathrm{NP} \\ \mathrm{PP} \\ \mathrm{P} \end{array} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

We index the columns with **Kronecker products** of Kronecker deltas – left factor 'tiles' the right:

$$\underbrace{\begin{array}{c} \begin{array}{ccc} \mathrm{NP} & \mathrm{PP} & \mathrm{P} \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{array}}_{\delta_{\mathrm{PP}}^\top} \otimes \underbrace{\begin{array}{c} \begin{array}{ccc} \mathrm{NP} & \mathrm{PP} & \mathrm{P} \end{array} \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \end{array}}_{\delta_{\mathrm{P}}^\top} = \begin{array}{c} \begin{array}{ccccccccc} \mathrm{NP\ NP} & \mathrm{NP\ PP} & \mathrm{NP\ P} & \mathrm{PP\ NP} & \mathrm{PP\ PP} & \mathrm{PP\ P} & \mathrm{P\ NP} & \mathrm{P\ PP} & \mathrm{P\ P} \end{array} \\ \begin{bmatrix} \underbrace{0\ 0\ 0}_{(\delta_{\mathrm{PP}}^\top)_{[1]}\ \delta_{\mathrm{P}}^\top} & \underbrace{0\ 0\ 1}_{(\delta_{\mathrm{PP}}^\top)_{[2]}\ \delta_{\mathrm{P}}^\top} & \underbrace{0\ 0\ 0}_{(\delta_{\mathrm{PP}}^\top)_{[3]}\ \delta_{\mathrm{P}}^\top} \end{bmatrix} \end{array}$$

So the grammar can be defined in terms of rule probabilities:

$$\mathbf{G} = \sum_c \delta_c \left[\left(\sum_{a,b} \mathsf{P}(c \to a\ b \mid c)\ \delta_a^\top \otimes \delta_b^\top \right) \left(\sum_w \mathsf{P}(c \to w \mid c)\ \delta_w^\top \right) \right]$$

We'll randomly generate this grammar, and use it to randomly generate trees...

23.3 Random sampling of grammars and trees

We define probabilities and samplers for **grammars**, using Dirichlet distributions (over distribs):

$$\mathsf{P}(\text{grammar } \mathbf{G}) = \mathrm{Dirichlet}(\mathbf{G}; \beta)$$

$$\mathbf{G} \sim \text{Dirichlet}(\beta) \tag{1}$$

and probabilities and samplers for **trees given grammars**, using the usual Categorical distrib:

$$\begin{aligned} \text{P}(\text{trees } \tau_{1..N} \mid \text{grammar } \mathbf{G}) &= \prod_{\tau \in \tau_{1..N}} \prod_{\tau_\eta \in \tau} {\delta_{\tau_\eta}}^\top \mathbf{G} \, (\delta_{\tau_{\eta 1}} \otimes \delta_{\tau_{\eta 2}}) \\ \tau_{\eta 1}, \tau_{\eta 2} &\sim \text{Categorical}({\delta_{\tau_\eta}}^\top \mathbf{G}) \end{aligned} \tag{2}$$

We could randomly generate all possible grammars and trees, then analyze those that match data...

That's called **rejection sampling**. But it would take a very long time!

We can't generate random trees and replace the leaves with our data – that distorts the distribution.

23.4 Gibbs sampler (Geman & Geman, 1984; Johnson et al., 2007)

Instead we use a **Gibbs sampler** to iteratively resample from the **posterior** distribution...

First, we define re-usable **likelihoods** of each possible phrase (word i to j) given each category:

$$\mathbf{v}_{i,j} = \mathbf{G}^{(t)} \begin{bmatrix} \sum_{k \in \{i+1..j-1\}} \mathbf{v}_{i,k} \otimes \mathbf{v}_{k,j} \\ [\![i + 1 = j]\!] \, \delta_{w_i} \end{bmatrix} \tag{3}$$

Then we resample trees using these likelihoods to complete the posterior:

1. First we choose a point $k_{i,j}$ to split the sequence of words between i and j:

$$k_{i,j} \sim \text{Categorical}\Big(\sum_{k \in \{i+1..j-1\}} \delta_k \, \delta_{c_{i,j}}^\top \mathbf{G}^{(t)} \, (\mathbf{v}_{i,k} \otimes \mathbf{v}_{k,j}) \Big) \tag{4}$$

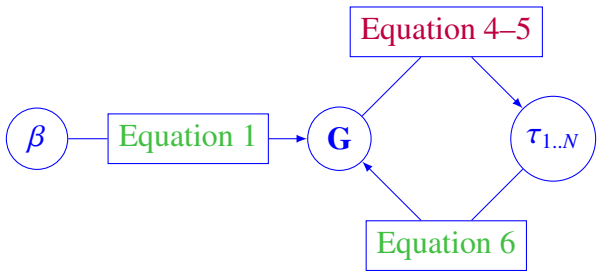
2. Then we choose categories for the word sequences on either side of the split point:

$$c_{i,k}, c_{k,j} \sim \text{Categorical}\Big(\delta_{c_{i,j}}^\top \mathbf{G}^{(t)} \, \text{diag}(\mathbf{v}_{i,k} \otimes \mathbf{v}_{k,j}) \Big) \tag{5}$$

Then we use statistics on the resampled trees for the corpus to resample the grammar:

$$\mathbf{G}^{(t)} \sim \text{Dirichlet}\Big(\beta + \sum_{\tau \in \tau_{1..N}^{(t-1)}} \sum_{\tau_\eta \in \tau} \delta_{\tau_\eta} \, (\delta_{\tau_{\eta 1}} \otimes \delta_{\tau_{\eta 2}})^\top \Big) \tag{6}$$

Schematically:



23.5 Labeled evaluation of induced trees (Jin et al., 2021)

We evaluate consistency against constituent annotations by linguists.

Use **homogeneity** for labeled evaluation (don't penalize induced subclasses):

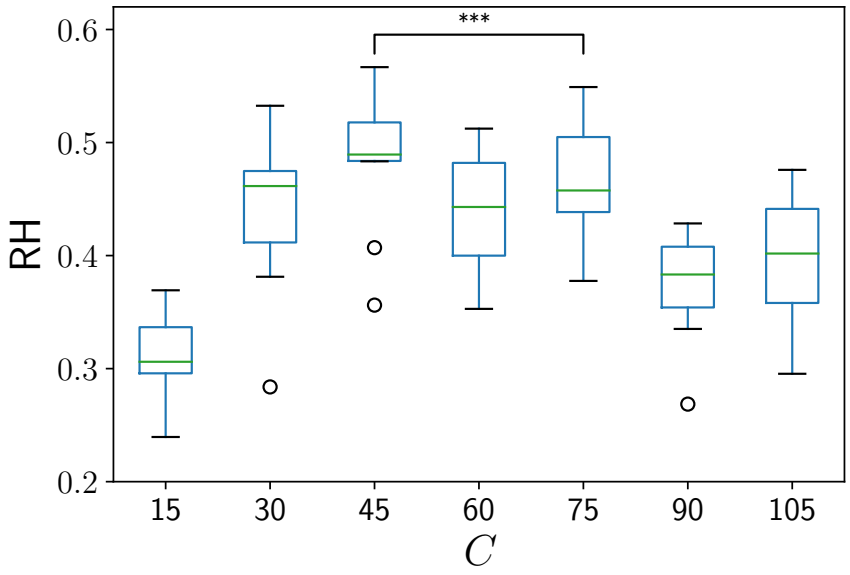
$$\begin{aligned}\mathsf{Hom}(\tilde{\tau}_{1..N}, \tau_{1..N}) &= 1 - \overbrace{\frac{\sum_{\tilde{c}\in\tilde{C}} \sum_{c\in C} \mathsf{P}(\tilde{c}, c) \log \mathsf{P}(\tilde{c} \mid c)}{\sum_{\tilde{c}\in\tilde{C}} \mathsf{P}(\tilde{c}) \log \mathsf{P}(\tilde{c})}}^{\text{conditional entropy over entropy}} \\ &= 1 - \underbrace{\frac{\sum_{n\in 1..N} \sum_{i,j \text{ s.t. } \tilde{\tau}_{n,i,j}\in\tilde{C}, \tau_{n,i,j}\in C} \log \mathsf{P}(\tilde{\tau}_{n,i,j} \mid \tau_{n,i,j})}{\sum_{n\in 1..N} \sum_{i,j \text{ s.t. } \tilde{\tau}_{n,i,j}\in\tilde{C}, \tau_{n,i,j}\in C} \log \mathsf{P}(\tilde{\tau}_{n,i,j})}}_{\text{same thing expressed over instances}}\end{aligned}$$

Multiply this by **recall** (don't penalize extra induced branches):

$$\mathsf{RH}(\tilde{\tau}_{1..N}, \tau_{1..N}) = \underbrace{\frac{\sum_{n\in 1..N} \sum_{i,j} [\![\tau_{n,i,j} \in C, \tilde{\tau}_{n,i,j} \in \tilde{C}]\!]}{\sum_{n\in 1..N} \sum_{i,j} [\![\tilde{\tau}_{n,i,j} \in \tilde{C}]\!]}}_{\text{correctly predicted brackets over true}} \cdot \mathsf{Hom}(\tilde{\tau}_{1..N}, \tau_{1..N})$$

23.6 Tuning on exploratory data

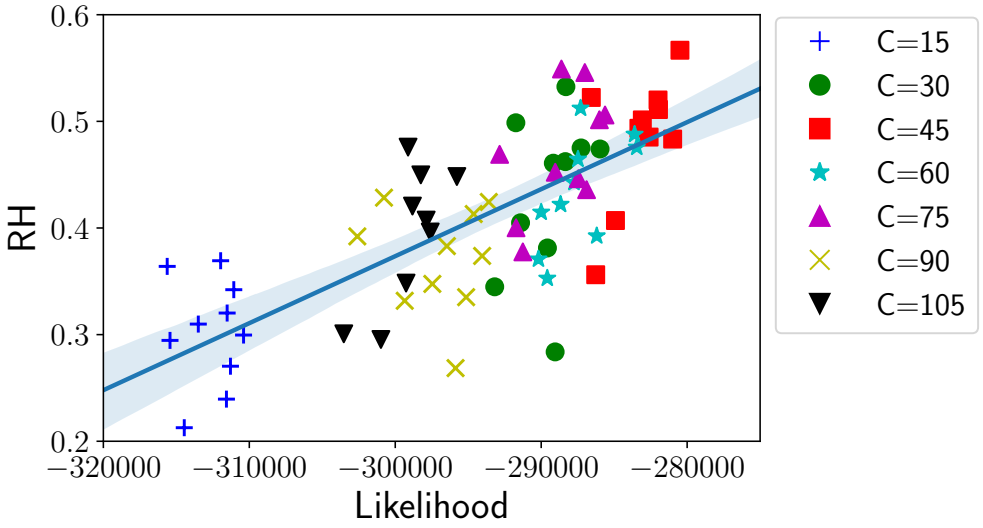
We find the number of categories $C = 45$ is optimal on an exploratory data set:



Data: MacWhinney (2000) 'Adam'.

23.7 Model selection

And we find models can reliably be selected using likelihood:



Data: MacWhinney (2000) 'Adam'.

23.8 Results on test set, child-directed speech transcripts

We get 44% of categories, comparable to other models that are less simple:

System	F1	RH
Seginer (2007)	0.52	-
Ponvert et al. (2011)	0.56	-
Shain et al. (2016)	0.66	-
Kim et al. (2019) without z	0.51	**0.44**
Kim et al. (2019) with z	0.31	0.39
Jin et al. (2021) (D=∞,C=45)	0.62	**0.44**
Right-branching	**0.76**	0.00

Data: MacWhinney (2000) 'Eve'.

Random grammar, constituents and categories would be less than .10 RH.

Note this is with no environment data, which presumably could help a lot.

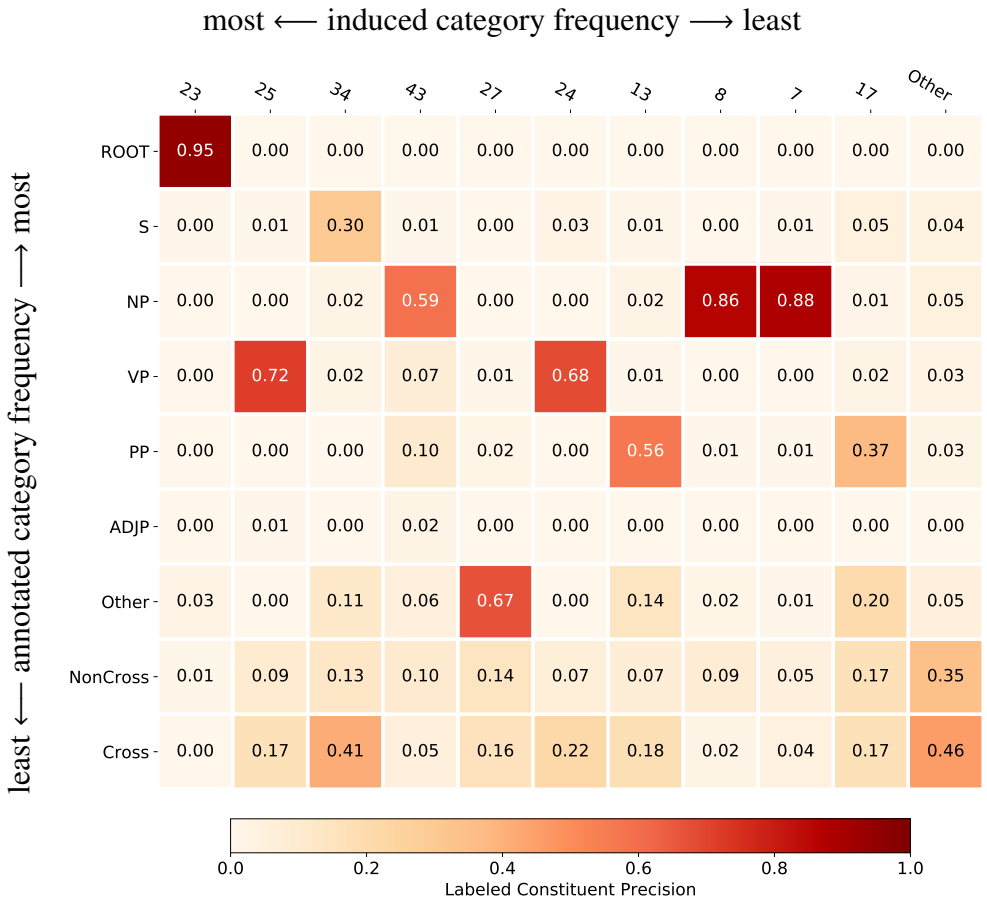
23.9 Discovered rules look like linguists'

Here's what the induced rules look like:

Rank Rule	Corresponding gold rules, counts and examples
1. 4→37 30 (5229)	?? →NP COP (0.53); ?? →NP AUX (0.09); ?? →NP VBZ (0.08); ?? →VB NP (0.05) dirt | is ; he | 's ; it | 's ; he | 's
2. 24→5 25 (4613)	?? →?? ?? (0.57); VP →MD VP (0.18); ?? →MD ?? (0.07); ?? →?? VB (0.04) will n't | step on your candy ; do n't | know ; 'm | afraid you 'll forget ; do n't | want to play
3. 42→33 25 (4294)	?? →NP VP (0.53); ?? →NP ?? (0.25); S →NP VP (0.10); ?? →NP VB (0.06) you | do ; you | show him ; you | do in the kitchen ; these | things to ride on
4. 36→11 8 (4155)	?? →VB NP (0.53); VP →VB NP (0.13); ?? →VB ?? (0.08); ?? →VBP NP (0.05) ask | ursula ; have | a bump ; put | the pillows ; see | that
5. 23→38 27 (4126)	ROOT →WHNP SQ (0.47); ROOT →WHADVP SQ (0.19); ?? →?? ?? (0.04) that 's a train part | is n't it ; who | 's there ; what | for ; you got your fingers in it | did n't you
6. 25→36 13 (3642)	?? →?? ?? (0.49); VP →VB PP (0.10); ?? →VP ?? (0.05); VP →VB ADVP (0.03) eat yourself | up ; do | when you go to school ; draw | on it ; play | with the record
7. 23→34 17 (3514)	?? →?? ?? (0.71); ?? →?? PP (0.06); ?? →?? SBAR (0.04); ?? →S ?? (0.04) paul stay away | away away from there ; no i do n't know | what delfc means
8. 34→4 43 (3227)	?? →?? ?? (0.57); ?? →?? NP (0.17); ?? →?? VBG (0.04); ?? →?? JJ (0.03) they 're | in your box ; they 're just | playing ; those are | stamps you use ; it says | here
9. 23→4 43 (3087)	?? →?? ?? (0.85); ?? →?? NP (0.02); ?? →?? VP (0.01); ROOT →VB NP (0.01) just like | adam ; you told | the carpenter you had a big burp ; they are | taking baths
10. 23→6 34 (3005)	ROOT →INTJ S (0.28); ?? →?? ?? (0.19); ?? →INTJ ?? (0.19); ROOT →INTJ FRAG (0.06) because | you 'll break it ; because | you 're still there ; oh | hurry up
11. 8→0 32 (2931)	NP →DT NN (0.50); ?? →?? ?? (0.14); NP →PRP$ NN (0.10); NP →DT NNS (0.07) any | noise ; the little | boy ; any | more ; the | policeman
12. 7→0 10 (2899)	NP →DT NN (0.55); NP →PRP$ NN (0.17); ?? →?? ?? (0.10); ?? →DT NN (0.03) your | wrist ; our | rug ; the | toy ; the other | side
13. 43→0 32 (2776)	NP →DT NN (0.45); ?? →?? ?? (0.23); NP →PRP$ NN (0.07); ?? →DT NN (0.06) any | more ; cowboy | hat ; morning or | afternoon ; a | lobster
14. 27→35 42 (2631)	?? →?? ?? (0.64); ?? →AUX ?? (0.24); ?? →MD ?? (0.06); SQ →COP NP (0.02) are | you going to do ; do | n't you tell ursula what you have ; did | you hurt yourself
15. 25→11 8 (2461)	VP →VB NP (0.60); ?? →VB NP (0.07); VP →VBP NP (0.05); ?? →VB ?? (0.05) close | it ; want | some more paper ; seen | everything ; like | it
16. 5→35 31 (2387)	?? →AUX NOT (0.80); ?? →MD NOT (0.17); ?? →COP NOT (0.01); VP →AUX NOT (0.01) do | n't ; did | n't ; do | n't ; does | n't
17. 27→30 37 (2278)	SQ →COP NP (0.51); ?? →COP NP (0.24); ?? →AUX NP (0.05); ?? →COP ?? (0.03) about | the treasure house ; is | it ; is | that ; is | it
18. 13→40 7 (2236)	PP →IN NP (0.67); ?? →IN ?? (0.08); ADVP →RB RB (0.07); ?? →IN NP (0.04) in | yours ; of | those ; around | here ; at | paul
19. 23→12 9 (2228)	?? →?? ?? (0.78); ?? →SBARQ ?? (0.05); ?? →SQ ?? (0.03); ?? →?? PP (0.03) is she dancing | on the horse 's back ; what kind | of paper ; what happens | when you press it
20. 0→0 1 (1844)	?? →DT JJ (0.56); ?? →DT NN (0.15); ?? →?? JJ (0.04); ?? →PRP$ JJ (0.03) a | nice ; a | dozen ; one | half ; a | few

23.10 Precision on exploratory data

Here's which induced categories got confused for which annotated categories:



23.11 Discovered preterminal categories

And here's the induced preterminal categories, which seem to learn case and agreement classes:

Rank	Induced category	Category count	Attested category and relative frequency
1.	0	11327	DT (0.77); PRP$ (0.16)
2.	33	9983	NP (0.99): you (.86), they (.05), we (.02) – nom,pl agreement
3.	11	8853	VB (0.71); VBP (0.12); VBD (0.07)
4.	30	8031	COP (0.64); AUX (0.09); VBZ (0.07); VP (0.05)
5.	32	7865	NN (0.81); NNS (0.10)
6.	37	7402	NP (0.86): that (0.38), it (0.27), he (0.07) – nom,3sg agreement
7.	35	7333	AUX (0.72); MD (0.17)
8.	38	6900	WHNP (0.54); WHADVP (0.23); WP (0.07)
9.	40	6712	IN (0.80); RB (0.05)
10.	8	6013	NP (0.89): it (0.36), them (0.06), me (0.06) – acc
11.	6	5424	INTJ (0.60); ADVP (0.10); NP (0.09); CC (0.05)
12.	28	4004	NP (0.99): I (0.70), he (0.10), it (0.06) – nom,1sg agreement
13.	10	3880	NN (0.88); NNS (0.08)
14.	43	3171	ADJP (0.27); NP (0.22); VP (0.13); JJ (0.10); VBG (0.07)
15.	31	3086	NOT (0.99)
16.	36	3043	VB (0.60); VP (0.18); VBP (0.05)
17.	13	2705	PRT (0.32); ADVP (0.32); NP (0.12)
18.	1	2483	JJ (0.66); NN (0.20)
19.	3	2388	TO (0.80); IN (0.15)
20.	18	2220	RB (0.20); NOT (0.19); VBG (0.13); IN (0.11)

References

Geman, S. & Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6), 721–741.

Jin, L., Schwartz, L., Doshi-Velez, F., Miller, T., & Schuler, W. (2021). Depth-Bounded Statistical PCFG Induction as a Model of Human Grammar Acquisition. *Computational Linguistics*, 47(1), 181–216.

Johnson, M., Griffiths, T., & Goldwater, S. (2007). Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference* (pp. 139–146). Rochester, New York: Association for Computational Linguistics.

Kim, Y., Dyer, C., & Rush, A. (2019). Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2369–2385).

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk*. Mahwah, NJ: Lawrence Elrbaum Associates, third edition.

Ponvert, E., Baldridge, J., & Erik, K. (2011). Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (pp. 1077–1086). Portland, Oregon.

Seginer, Y. (2007). Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (pp. 384–391).

Shain, C., Bryce, W., Jin, L., Krakovna, V., Doshi-Velez, F., Miller, T., Schuler, W., & Schwartz, L. (2016). Memory-bounded left-corner unsupervised grammar induction on child-directed input. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 964–975). Osaka, Japan: The COLING 2016 Organizing Committee.