A Framework for Lexicalized Grammar Induction Using Variational Bayesian Inference

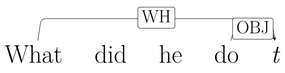
Chris Bruno[a*], Eva Portelance[b*], Daniel Harasim[c], Leon Bergen[d], Timothy J. O'Donnell[a]

(a) McGill University; (b) Stanford University; (c) École Polytechnique Fédérale de Lausanne; (d) UC San Diego; *Co-first authors

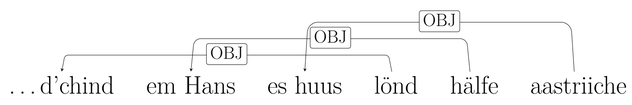
L2HM: Learning Language in Humans and in Machines 2018

Introduction

- We introduce a probabilistic learning model for a class of lexicalized grammar formalisms
- We use these tools to develop a computational framework for investigating ideas in theoretical syntax, by assessing their learnability via *compactness* studies, similar to the methodology in [1, 2]
- We use Minimalist Grammars, designed for being suitable in deriving *long distance dependencies* via movement,

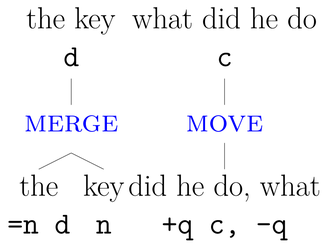


as well as non-context free dependencies such as *crossing dependencies* [3]



Minimalist Grammars

- A *Directional Minimalist Grammar* (DMG, [4, 5]) is a tuple $\langle \mathcal{L}, R \rangle$ where
  - $\mathcal{L}$ is a finite set of lexical items whose syntactic features are of five types:
    1. *category* (e.g. `v, d, p`) - define the syntactic categories (verb, noun . . .);
    2. right selector (e.g. `=d, =p`) - select argument constituent to the right
    3. left selector (e.g. `d=, p=`) - selects argument constituent to the left
    4. licensor (e.g. `+case, +wh`) - select moving constituent;
    5. licensee (e.g. `-case, -wh`) - selected moving constituent.
  - $R = \{\text{merge}, \text{move}\}$ is the set of structure building operations.



- We use *DMGs*, which can merge either to the right or to the left of a node (eg. *the key* vs *key the*). Movement is always to the left.
- A probabilistic MG also contains:
  3. $\boldsymbol{\theta}$ - where each $\boldsymbol{\theta}_c \in \boldsymbol{\theta}$ is a probability distribution over all lexical items l with the category feature c such that

$$\sum_l \boldsymbol{\theta}_{c,l} = 1$$

- We sample each $\boldsymbol{\theta}_c$ from a Dirichlet prior parameterized by $\boldsymbol{\alpha}$.

The Generative Model

- We implement a head-out generative model for derivations in a Merge-only variant of the Directional MG.
- The generative model allows us to forward sample derivations, or sentences, given a Lexicon.
- Let d_l be a derivation headed by the lexical item l.
- Let $cat(l)$ be the category feature of l.
- Let $sel(d_l)$ give the subderivations $d_{l_1}, \ldots, d_{l_k}$ that are selected by the head l in derivation d.

$$MG(d_l) = \begin{cases} \boldsymbol{\theta}_{cat(l),l} \times \prod_{i=1}^{k} MG(d_{l_1}) & sel(l) = d_{l_1} \ldots d_{l_k}, k \geq 1 \\ \boldsymbol{\theta}_{cat(l),l} & sel(l) = \emptyset \end{cases}$$

- This returns the probability of the derivation d_l, which is the product of its lexical items.

Variational Bayesian Inference

- In order to learn probabilities to a grammar, we calculate the posterior $P(D, \boldsymbol{\theta} | S, \boldsymbol{\alpha})$, where D is a sequence of deriavtions over a corpus S.
- Approximate by minimize the KL distance between the true posterior P and the *variational approximation* Q.

$$Q^*(D, \boldsymbol{\theta}) = \arg \min_{Q(D,\boldsymbol{\theta})} KL(Q(D, \boldsymbol{\theta}) || P(D, \boldsymbol{\theta} | S, \boldsymbol{\alpha})).$$

- Variational independence assumption (where $\boldsymbol{d}_n \in D, 1 \leq n \leq N$ is a derivation):

$$Q(D, \boldsymbol{\theta}) = Q(D)Q(\boldsymbol{\theta}) = \prod_{n=1}^{N} Q(\boldsymbol{d}_n) \prod_{k=1}^{K} Q(\boldsymbol{\theta}_k).$$

- The optimal variational distributions are

$$Q^*(\boldsymbol{\theta}) = \prod_k \text{Dir}(\boldsymbol{\theta}_k; \boldsymbol{w}_k)$$

$$w_{km} = \alpha_{km} + \sum_{n=1}^{N} \sum_{\boldsymbol{d}_i \in \Phi(s_n)} Q(\boldsymbol{\delta}_i | s_n) c(l_{km}; \boldsymbol{\delta}_i)$$

$$Q^*(D) = \frac{1}{Z_D} \prod_{n=1}^{N} \prod_{i : l_{k_{ni} m_{ni}} \in \boldsymbol{d}_n} \theta^*_{k_{ni} m_{ni}}$$

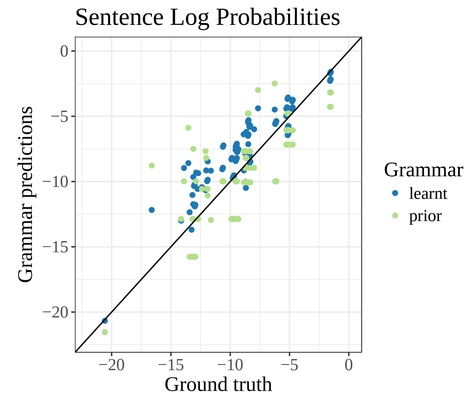
$$\theta^*_{km} = e^{\psi(w_{km}) - \psi(\sum_m w_{km})}$$

- **Algorithm.** Update each w_{km} and each θ_{km} until the KL coverges, where $\boldsymbol{w}$ is initialized to $\boldsymbol{\alpha}$.
- This algorithm is guaranteed to find a posterior which is at least a local minimum.

Experiment 1: Grammar Recoverability

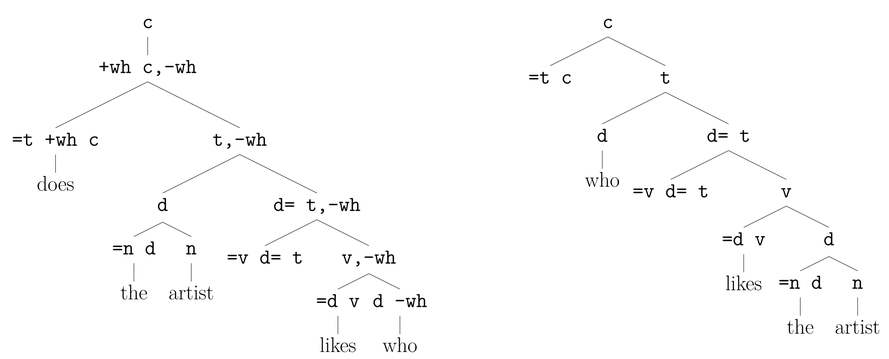
- We start with a recursive grammar of 24 lexical items without movement and sample a corpus of 1000 sentences.
- We run our inference algorithm for 10 iterations starting from a uniform prior conditioning on the corpus to test the learning algorithm.
- We compare the probability distribution over 135 unique newly sampled sentences given the ground truth grammar to the retrieved learnt distribution and the prior distribution.
- **Results**:

	Earth mover's	KL Divergence
Prior	184.74	4.02
Learnt	45.2	0.62



Experiment 2: Learning Movement

- Grammar of 49 lexical items including wh-words, nouns, determiners, and transitive verbs. Each sentence of containing a wh-word is ambiguous between 2 parses, with and without movement. Its language contains 21,888 sentences.
- Training set is a random sample of 218 sentences (1% of the language).
- The learned grammar uses movement for wh-objects, but not wh-subjects:



Experiment 3: Recovering English Dependencies

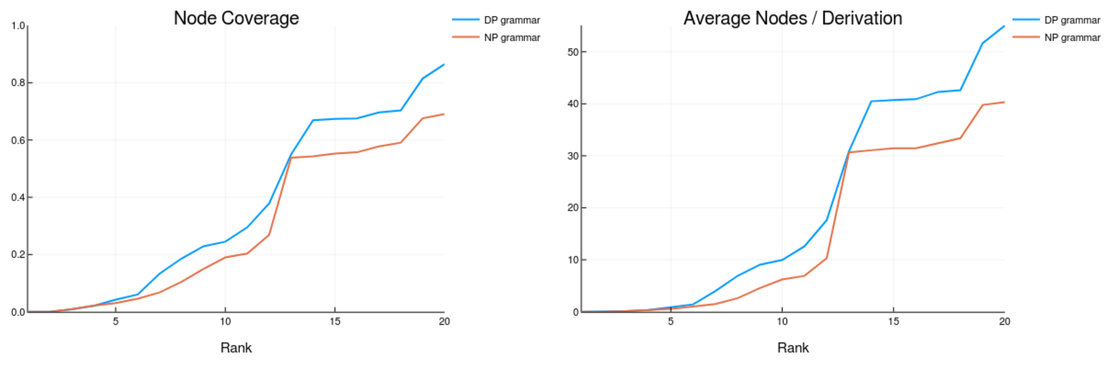
- Universal Dependencies English ParTUT corpus, with given train/test splits, with 1754/151 sentences respectively, averaging 24/22 words per sentence.
- Trained for 2 iterations with two grammars, semi-supervized conditioned on gold dependencies:
  - G_{DP}: Simplistic, hand-built grammar inspired by Minimalist Theory, respecting the DP hypothesis.
  - G_{NP}: The same, but respecting the NP-hypothesis.
- Accuracy results:

	train		test	
	prec	recall	prec	recall
G_{DP}	35.53	35.60	35.58	35.38
G_{NP}	42.21	42.28	41.01	41.01
G_{DP} uniform	28.62	28.69	27.33	27.33
G_{NP} uniform	34.94	35.01	34.37	34.37
G_{DP} best	47.35	47.38	50.58	50.57
G_{NP} best	56.11	56.16	56.92	56.89

- The *best* results pick the best performing parse for each sentence, giving an approximate upper bound, showing that our grammars are not capable of recovering around half of the gold dependencies, but our learning algorithm improves upon the uniform grammar approaching the best results.
- The NP grammar has higher accuracy, likely because the gold dependencies are Noun-headed.

Experiment 4: Compactness comparisons

- Given a grammar G, the grammar of rank k is the subset of G containing the k highest scored lexical items in each category.
- A more compact grammar is expected to be more successful at parsing for lower values of k and is expected to have more complex parses.
- The DP grammar does better at both properties:



References

[1] Leon Bergen, Edward Gibson, and Timothy J O'Donnell. A learnability analysis of argument and modifier structure. 2015.

[2] Ezer Rasin and Roni Katzir. A learnability argument for constraints on underlying representations. In *NELS 45*, 2014.

[3] Stuart M Shieber. Evidence against the context-freeness of natural language. *The Formal complexity of natural language*, 33:320–332, 1985.

[4] Edward Stabler. Derivational minimalism. *Logical Aspects of Computational Linguistics*, pages 68–95, 1997.

[5] Edward P Stabler. Computational perspectives on minimalism. *Oxford handbook of linguistic minimalism*, pages 617–643, 2011.

NSERC CRSNG

SSHRC CRSH
Social Sciences and Humanities Research Council of Canada
Conseil de recherches en sciences humaines du Canada

CRBLM

Contact: chris.bruno@mail.mcgill.ca; portelan@stanford.edu