

**Data-Oriented Language Learning –
Weight Estimation and Structure Search
in the Unsupervised Learning of STSGs**

Willem Zuidema
ILLC
University of Amsterdam

Plan of the talk

1. PCFGs: Weight Estimation & *Structure Search*
2. Linguistic Motivation for DOP/STSGs
3. STSGs: *Weight Estimation* & Structure Search
4. Calculating the Likelihood of Arbitrary Fragments

Learning PCFGs

- Fixed structure, (un)labeled data: EM
- Variable structure, labeled data: tree bank grammars, augmented non-terminal labels, smoothing
- Variable structure, partially labeled or unlabeled data: ? (Bayesian Model Merging, ABL, Constituent-Context Model)

Structure Search: Basic Operations

INCORPORATE($\alpha\beta \dots \omega$): add the following rules to the grammar

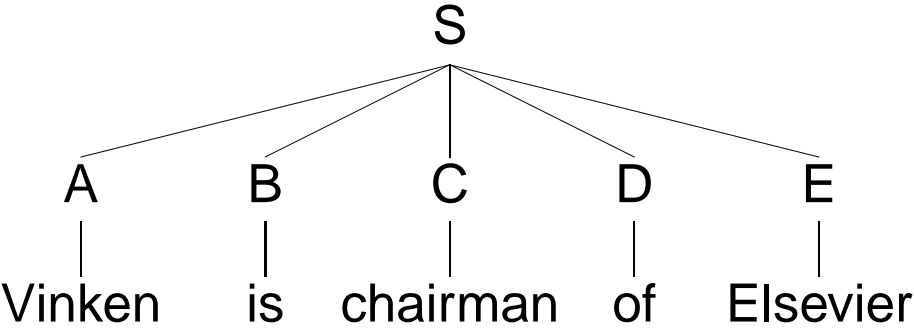
- $S \rightarrow AB \dots W$
- $A \rightarrow \alpha$
- $B \rightarrow \beta$
- ...
- $W \rightarrow \omega$

CHUNK(A, B): replace a sequence (all sequences) AB with C , and add the following rule to the grammar:

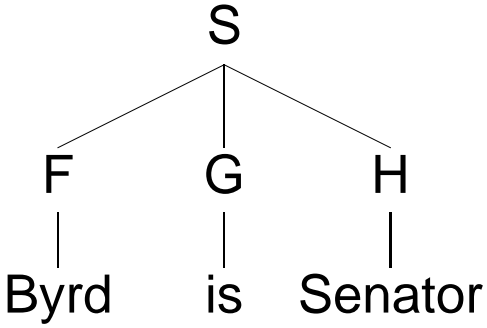
- $C \rightarrow AB$

MERGE(A, B): replace an (all) occurrence(s) of B with A , and remove all redundant rules.

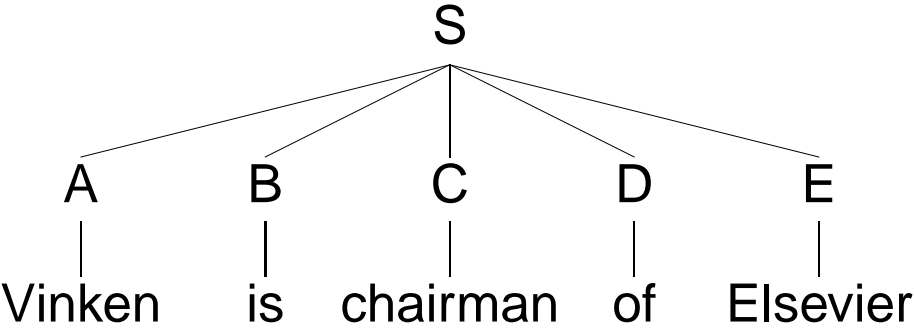
Vinken is chairman of Elsevier NV



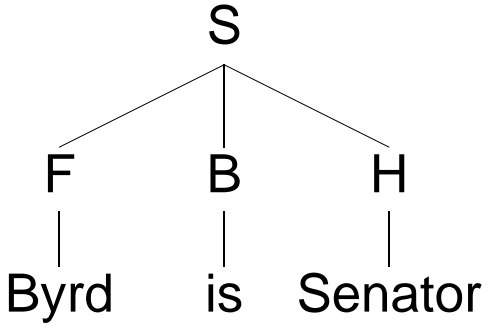
Byrd is Senator



Vinken is chairman of Elsevier NV

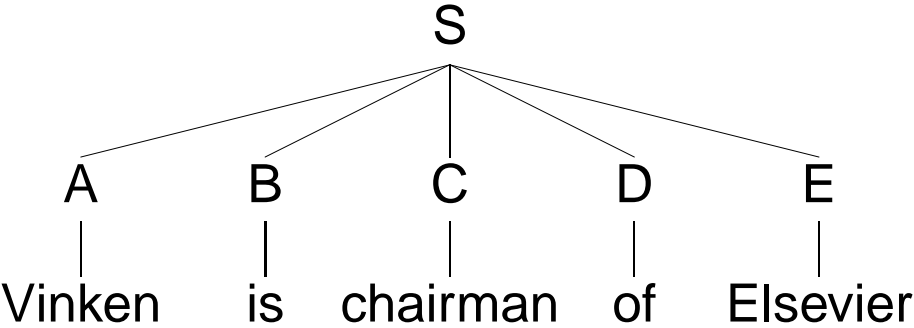


Byrd is Senator

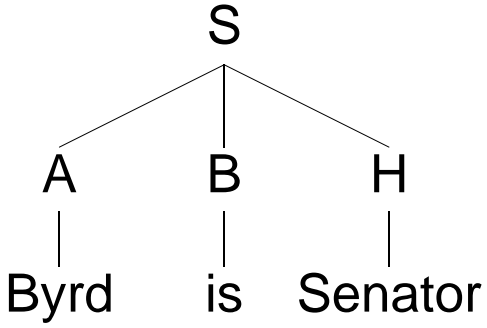


MERGE(B,G)

Vinken is chairman of Elsevier NV

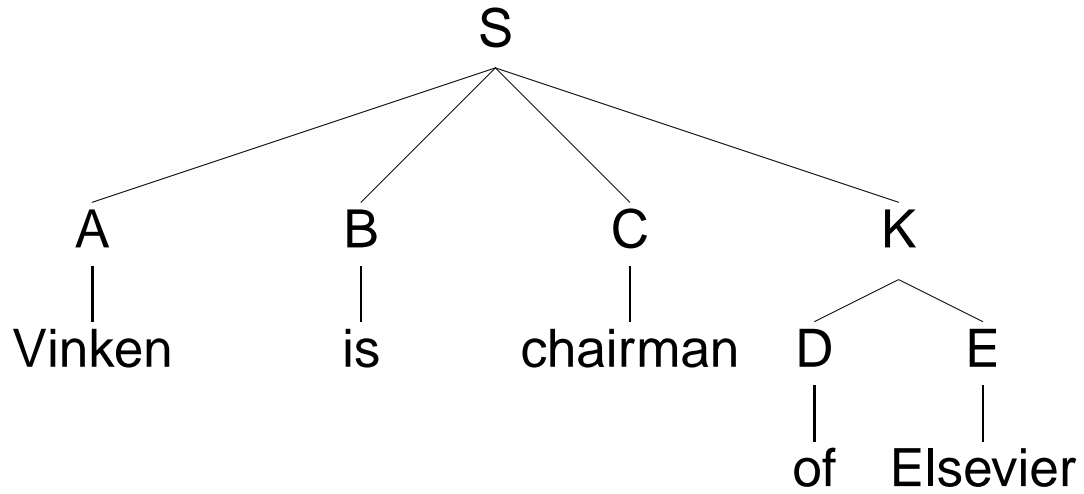


Byrd is Senator

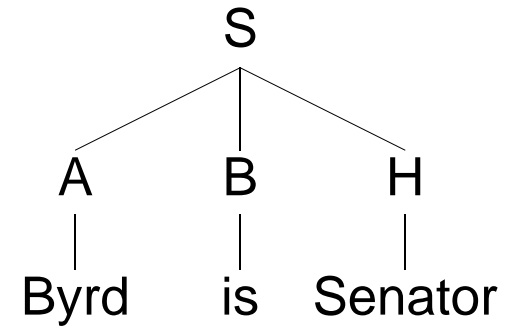


MERGE(B,G), MERGE(A,F)

Vinken is chairman of Elsevier NV

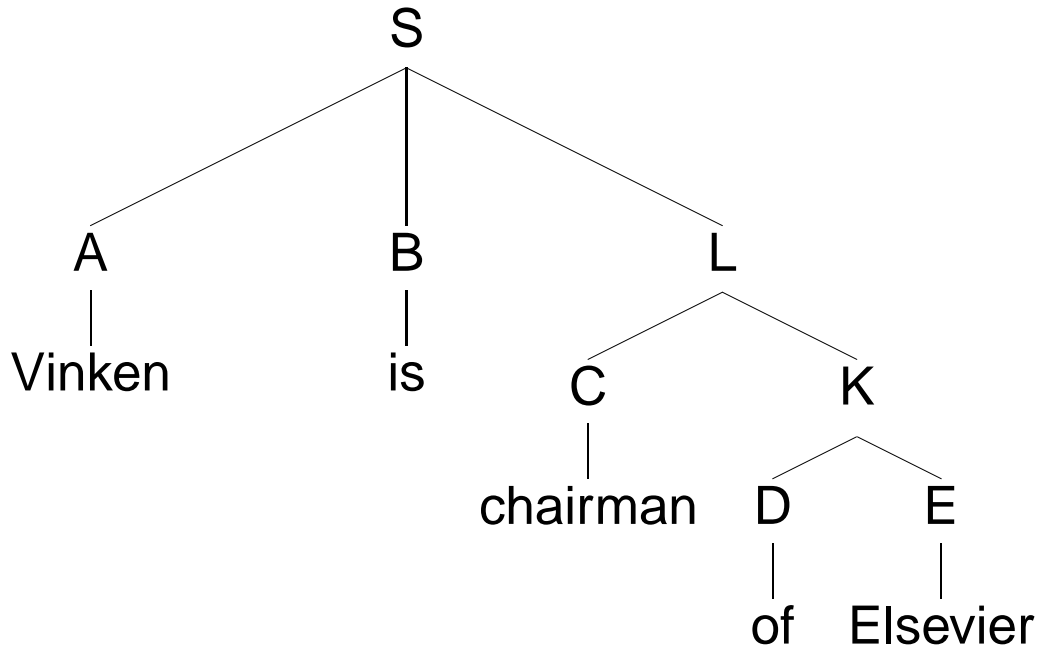


Byrd is Senator

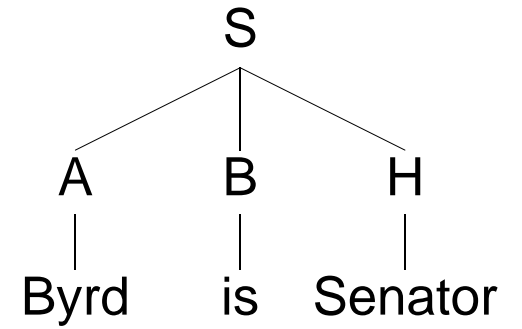


MERGE(B,G), MERGE(A,F), CHUNK(D,E)

Vinken is chairman of Elsevier NV

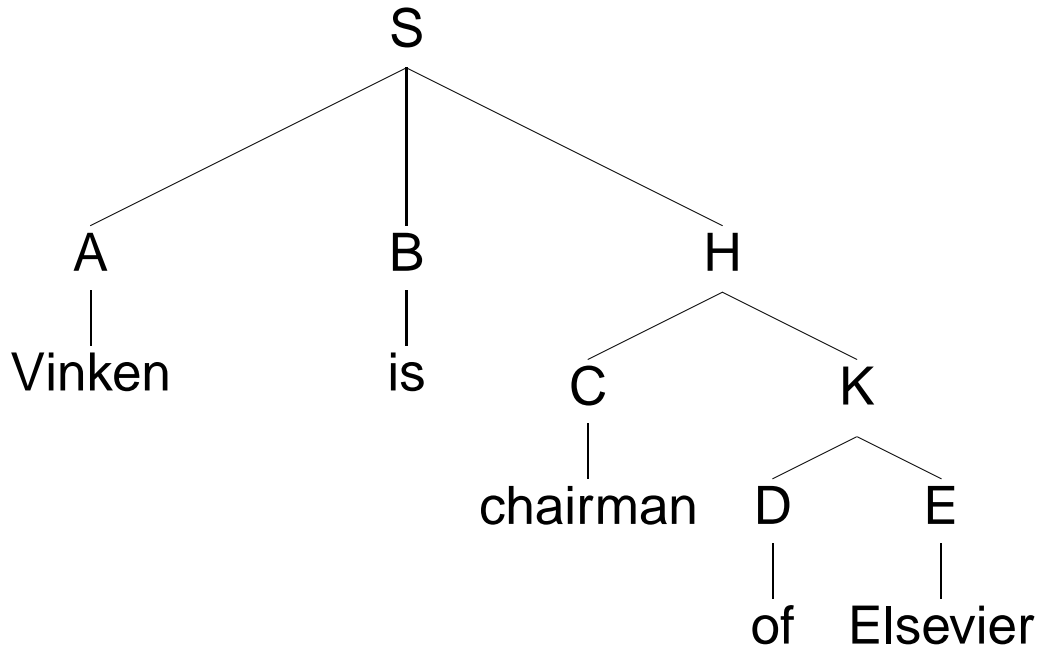


Byrd is Senator

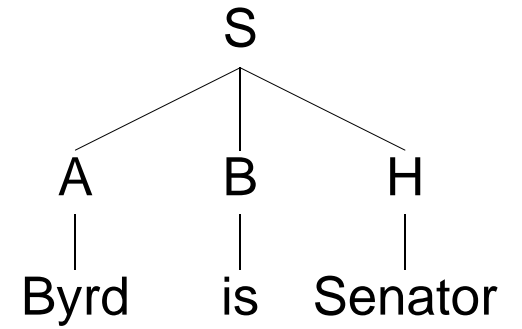


MERGE(B,G), MERGE(A,F), CHUNK(D,E), CHUNK(C,K)

Vinken is chairman of Elsevier NV

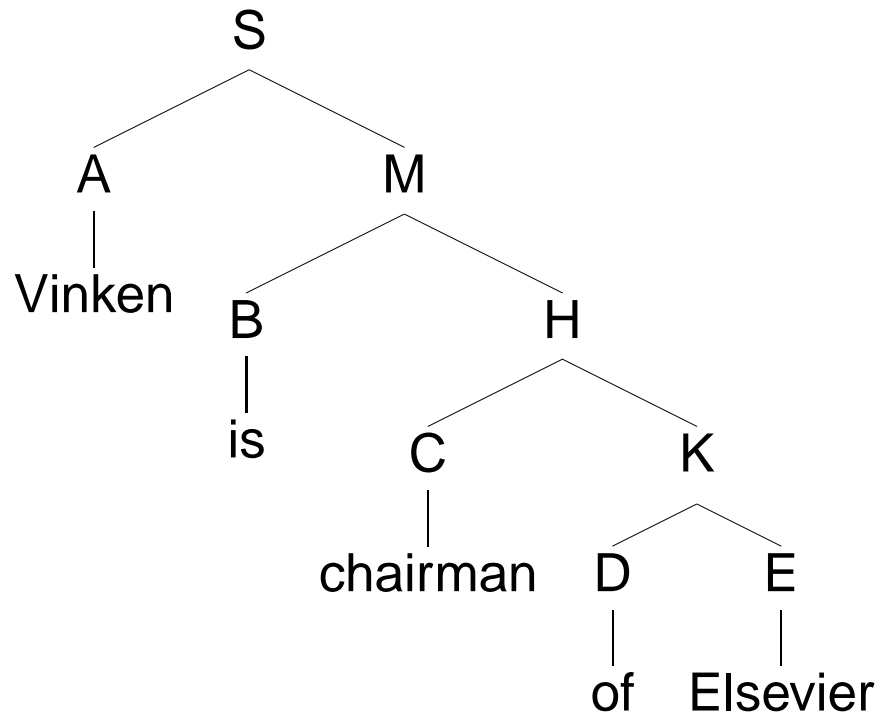


Byrd is Senator

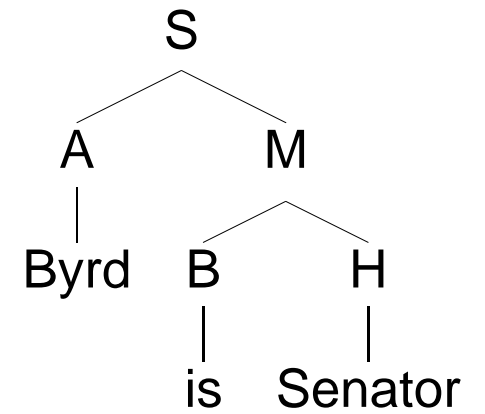


MERGE(B,G), MERGE(A,F), CHUNK(D,E), CHUNK(C,K), MERGE(H,L)

Vinken is chairman of Elsevier NV



Byrd is Senator



MERGE(B,G), MERGE(A,F), CHUNK(D,E), CHUNK(C,K), MERGE(H,L), CHUNK(B,H)

Structure Search: Search Strategies

Distributional Information: application of “chunk” and/or “merge” depends on (local) co-occurrence statistics (Wolff, 1983; Adriaans, 1992; Langley, 1994; Van Zaanen, 2001; Zuidema, 2003; Solan et al. 2003-2005; and many others, as reviewed in Pinker, 1979)

Maximum Likelihood: application of “chunk” depends on (global) **likelihood** of data (Klein & Manning, 2003-2005)

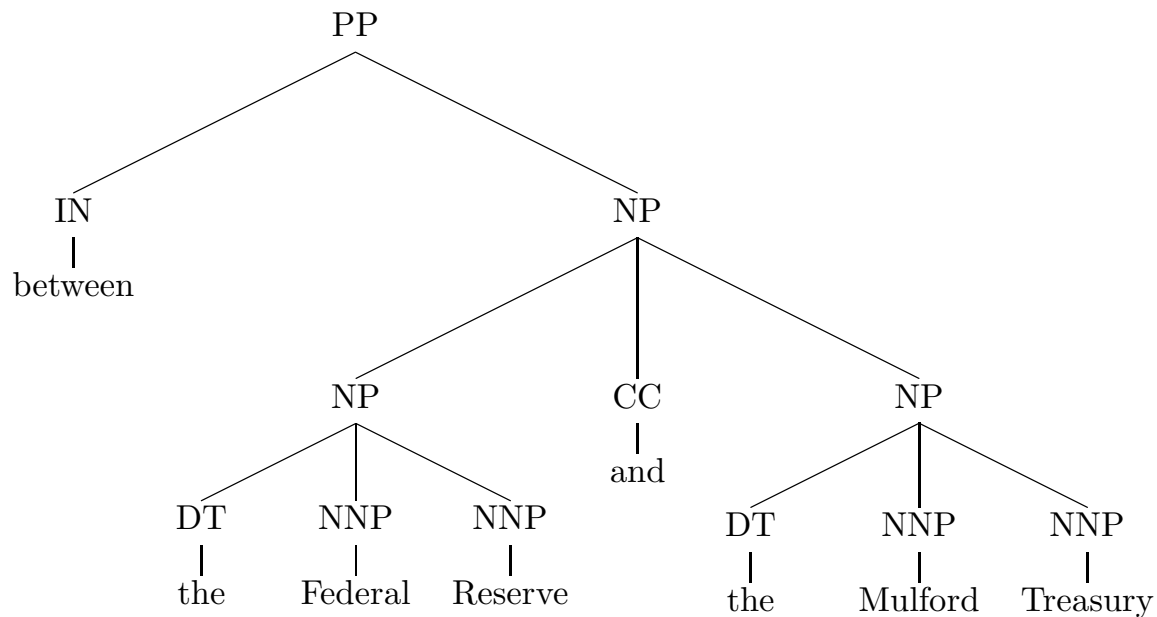
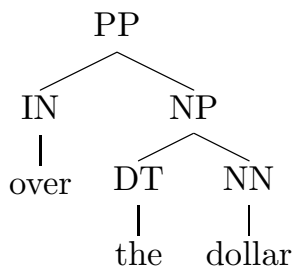
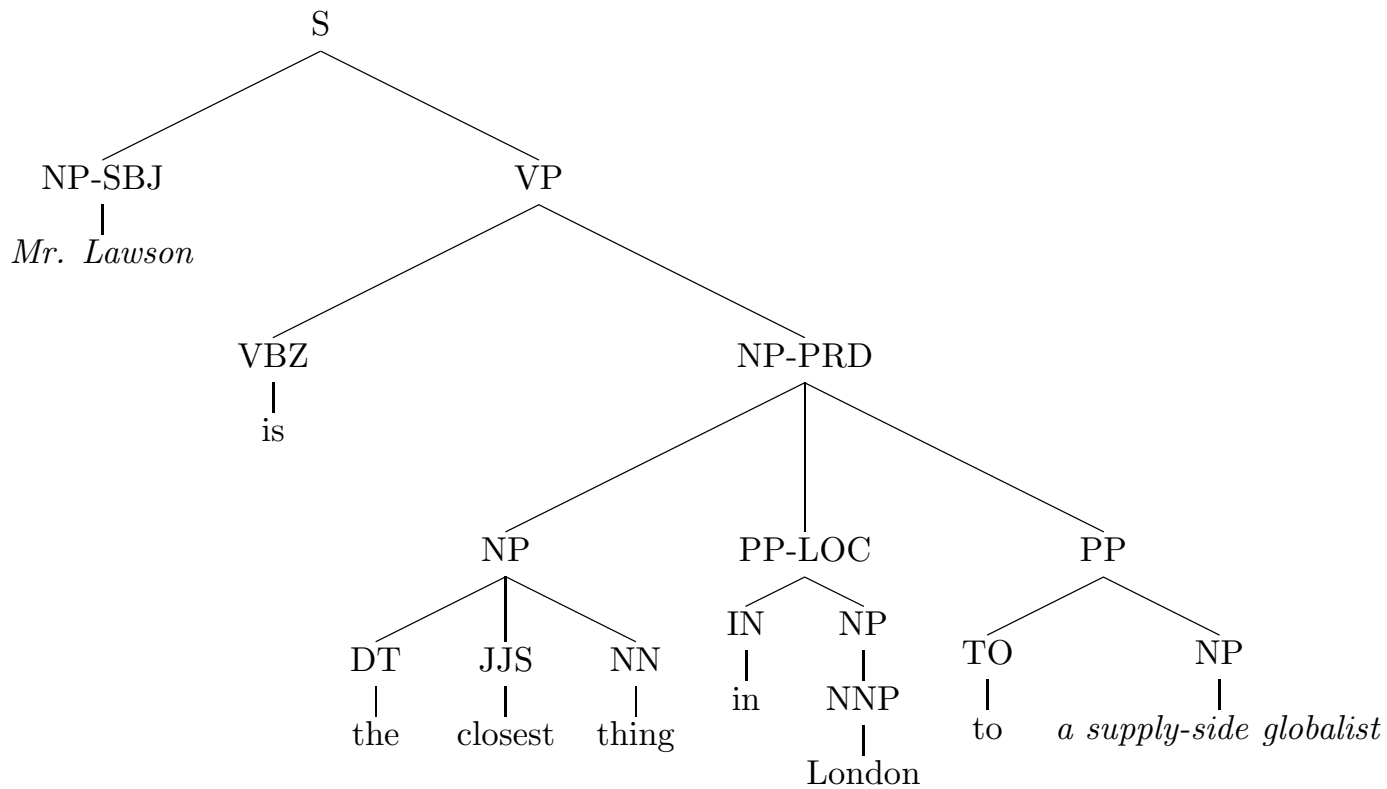
Bayesian Posterior Probability: application of “chunk” and/or “merge” depends on (global) **likelihood** and description length-based prior probability (Stolcke, 1994)

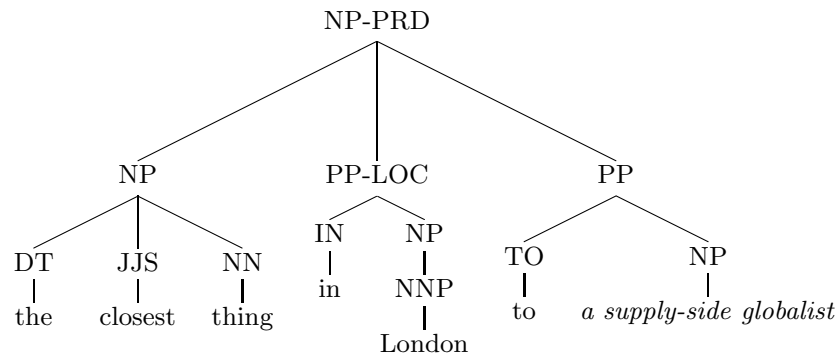
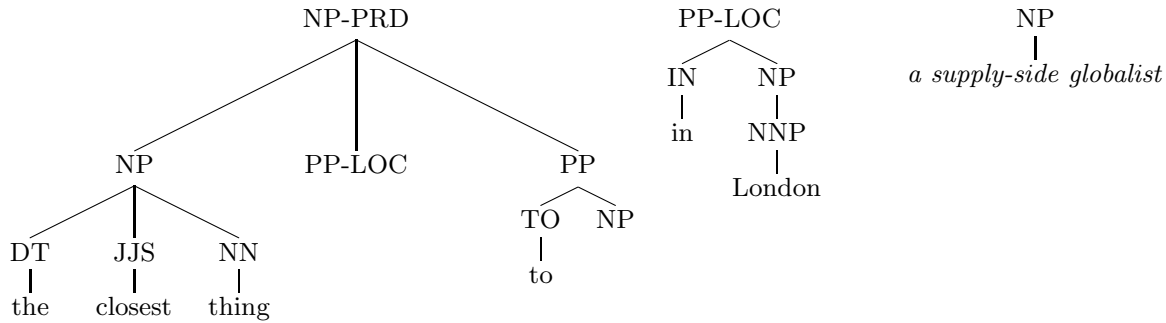
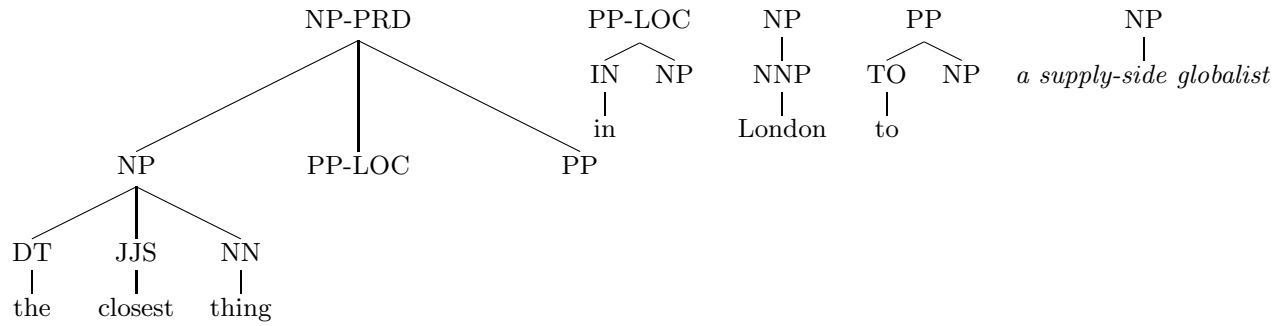
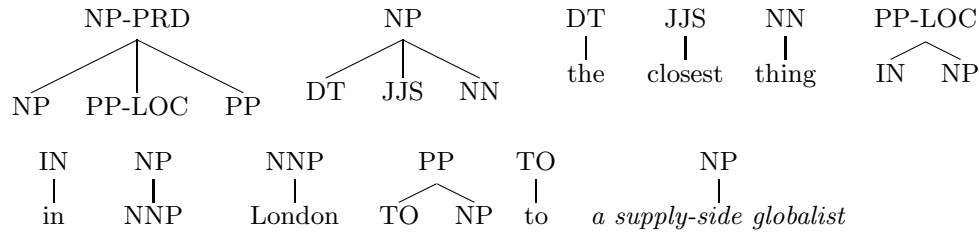
- $\operatorname{argmax}_G P(G|D) = \operatorname{argmax}_G P(D|G)P(G)$
- approximation of likelihood using Viterbi parses
- best first search, multi-level best first, beam search

Constructions

- (1)
 - a. Grandpa kicked the bucket.
 - b. Logician after logician wants to do linguistics.
- (2)
 - a. What time is it?
 - b. #How late is it?
- (3)
 - a. When is the next train *from Amsterdam to Paris*?
 - b. BA carried *more people than* cargo in 1987.
 - c. Lawson is *the closest thing* in London *to* a supply-side globalist.

(Fillmore, Kay, Goldberg, Jackendoff; Bod, 1998)





Learning STSGs

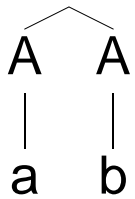
- Fixed structure, (un)labeled data: -
- Variable structure, labeled data: DOP – tree bank grammars, all subtrees (with some heuristic constraints), weights based on frequency f' in sample of size N (Bod, 2003):

$$w(t) = \frac{f'(t)}{\sum_{t' \in r(t)} f'(t')}$$

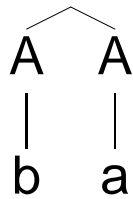
- Variable structure, partially labeled or unlabeled data: ?

Will DOP2003 work as well across tree banks?

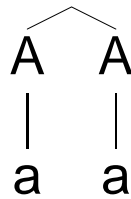
$t_1 = S$



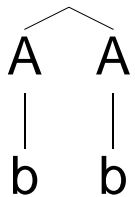
$t_2 = S$



$t_3 = S$



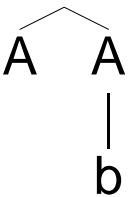
$t_4 = S$



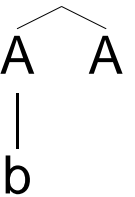
$t_5 = S$



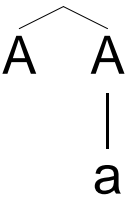
$t_6 = S$



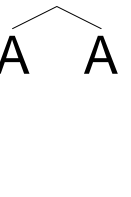
$t_7 = S$



$t_8 = S$



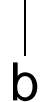
$t_9 = S$



$t_{10} = A$



$t_{11} = A$



Consider the situation where $f_3 = f_4 = 0$ and $f_1 > 0$ and $f_2 > 0$.

This implies:

$$\begin{aligned} w_3 + w_5w_{10} + w_8w_{10} + w_9w_{10}w_{10} &= 0 \\ w_4 + w_6w_{11} + w_7w_{11} + w_9w_{11}w_{11} &= 0, \end{aligned}$$

but DOP2003 will give all weights $w_5 \dots w_{11}$ nonzero values: it will always generalize.

(DOP* (Zollmann & Sima'an, in press) will, in the limit, only give non-zero weights to w_3 and w_4).

STSGs & Structure Search

- Induce a (P)CFG & parse sentences (or induce tree bank), use DOP as in supervised learning (Van Zaanen, 2003)
 - uncertainty about constituents is lost in intermediate representation;
 - if CFG-assumption is good enough for the training data, why no for the test data?
- Chunk & Merge operations, beam search for *maximal posterior probability*.

→ For better data-oriented language learning – whether from labeled, partially labeled or unlabeled data – we need good methods to calculate and approximate **likelihood** of the data.

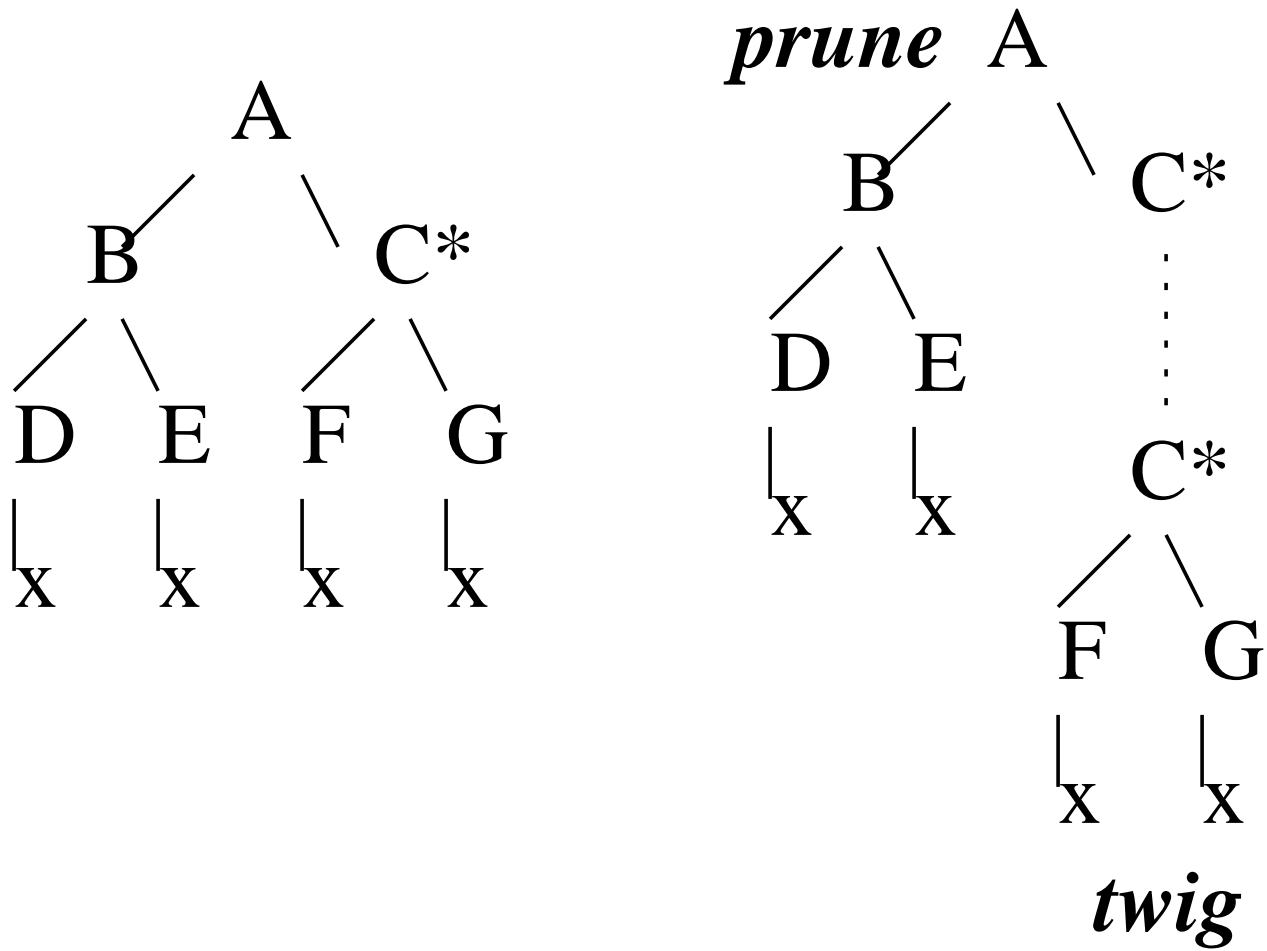
Likelihood

How do we calculate likelihood?

- of unlabeled text? sum of probabilities of all parses
- of arbitrary subtrees (fragments)? ?

→ good approximations of the likelihood of text

→ general equation in terms of usage probabilities $u(\tau)$ of elementary trees τ .

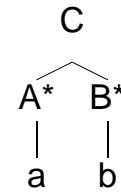
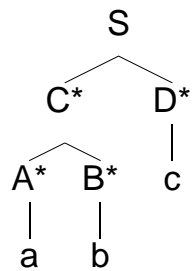


Prune: subtree resulting from root operation

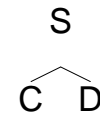
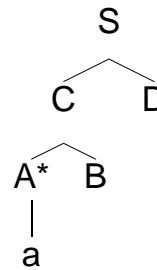
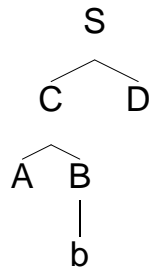
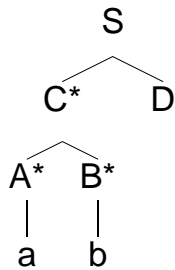
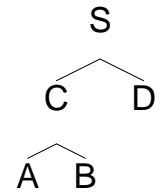
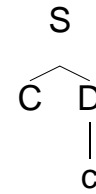
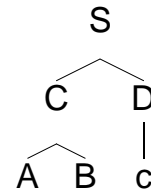
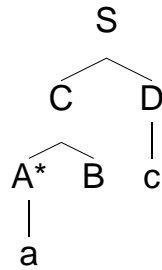
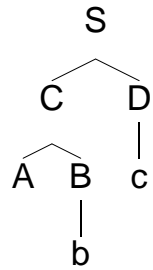
Twig: subtree resulting from frontier operation

How do we find all subtrees of a given tree?

→ all prunes of all twigs



twigs



prunes

Subtrees, twigs and prunes

the set of twigs:

$$tw(t) = \{t' \mid t' = t \vee \exists t'' (t'' \circ t' = t)\}$$

the set of prunes:

$$pr(t) = \{t' \mid t' = t \vee \exists t'', t''', \dots (t' \circ t'' \circ t''' \circ \dots = t)\}$$

the x-prune:

$pr_x(t)$ = the tree that is created by pruning t at each of the nodes in x

the set of subtrees:

$$st(t) = \{t' \mid \exists t'' (t'' \in tw(t) \wedge t' \in pr(t''))\}.$$

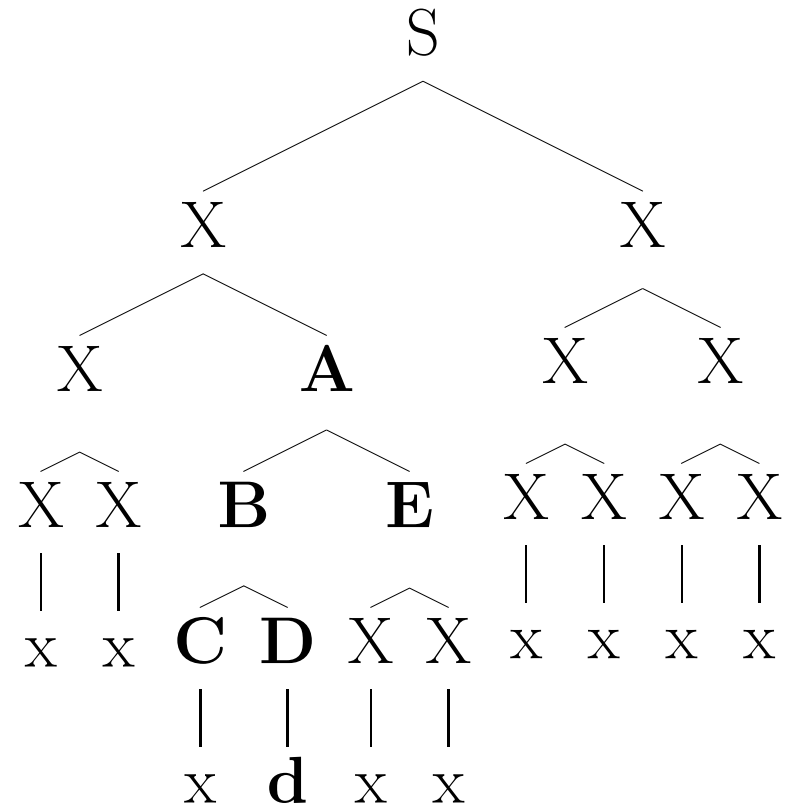
the sets of supertwigs, superprunes and supertrees:

$$\begin{aligned}\widehat{tw}(t) &= \{t' \mid t \in tw(t')\} \\ \widehat{pr}_x(t) &= \{t' \mid t \in pr_x(t')\} \\ \widehat{st}(t) &= \{t' \mid t \in st(t')\}\end{aligned}$$

Consider a focal subtree t
and its derivations $d_1 \circ \dots \circ d_n$

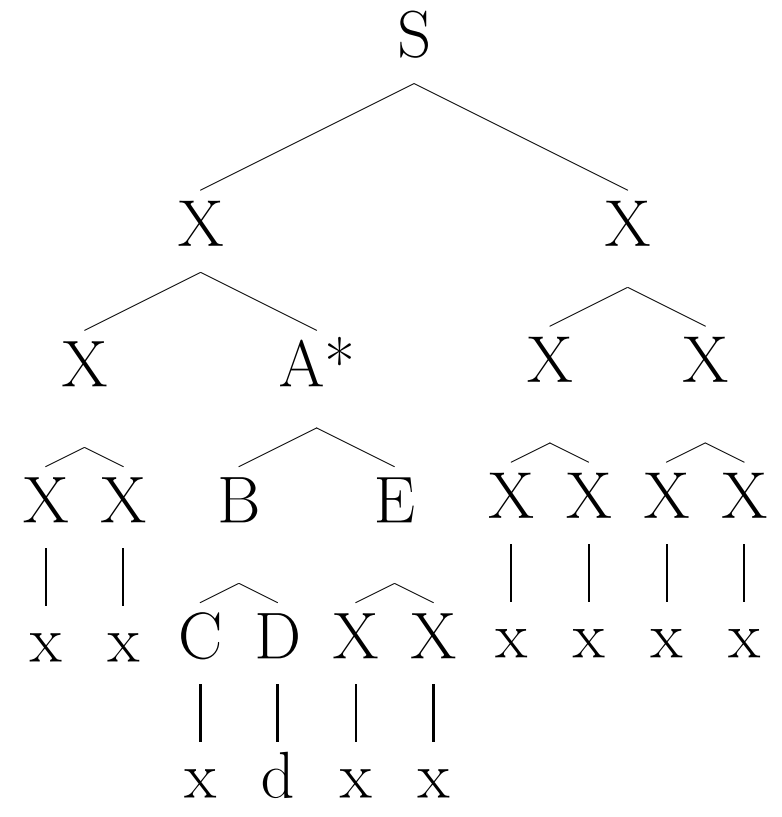
and each of its complete supertrees T
and its derivations $D_1 \circ \dots \circ D_l$

$$\mathbf{E}[f(t)] = \sum_{p \in T^*} (P(p)C(t, p))$$



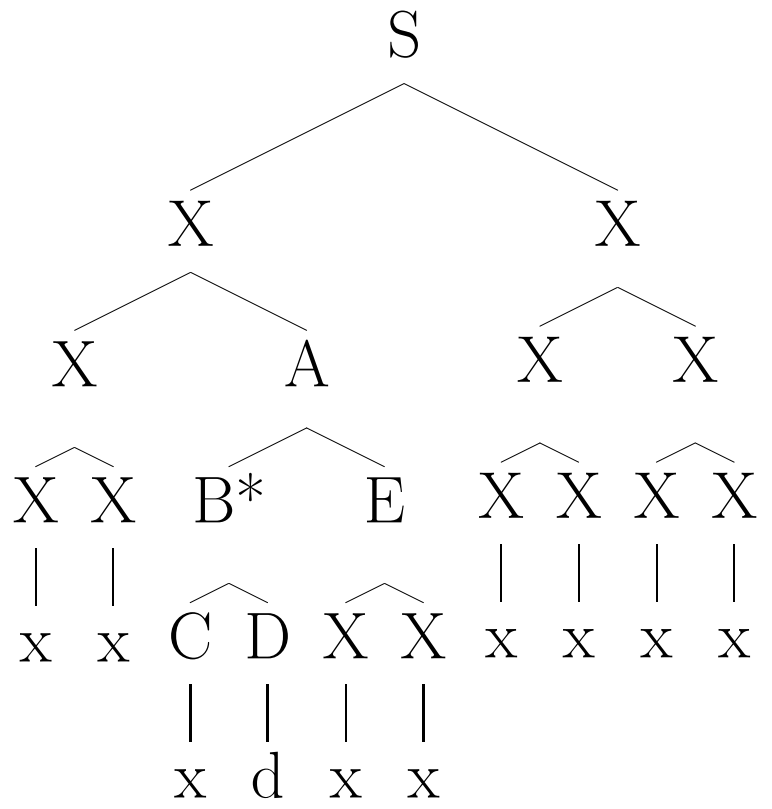
$$D_1 \circ \dots \underbrace{D_k}_{\dots A} \circ \underbrace{D_{k+1}}_{A \dots} \circ \dots \circ D_l$$

$$t = d_1 \circ \dots \circ d_n$$



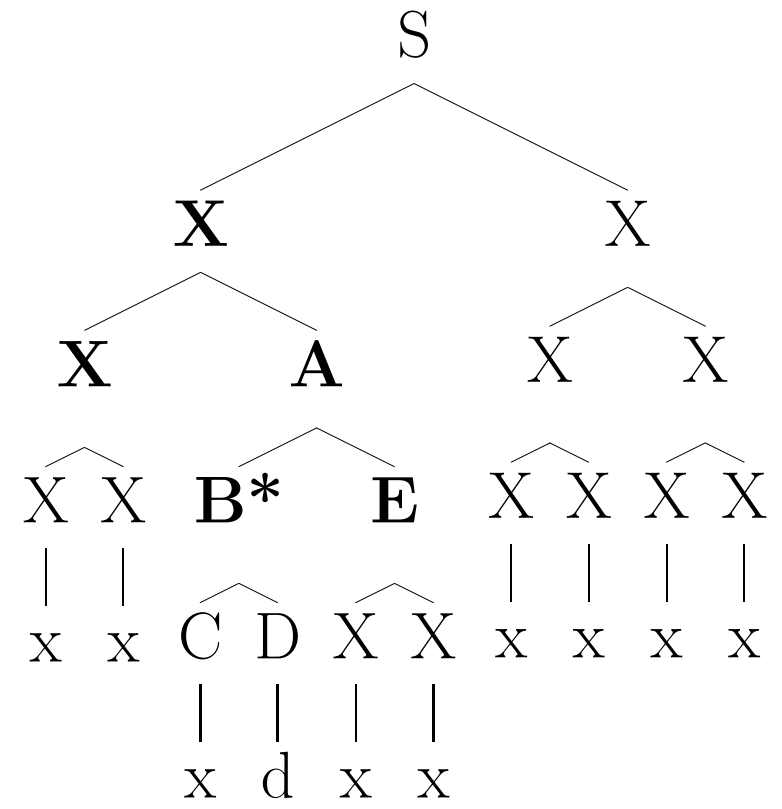
$$D_1 \circ \dots \underbrace{D_k}_{\dots B} \circ \underbrace{D_{k+1}}_{B \dots} \circ \dots \circ D_l$$

$$t = d_1 \circ \dots \circ d_n$$



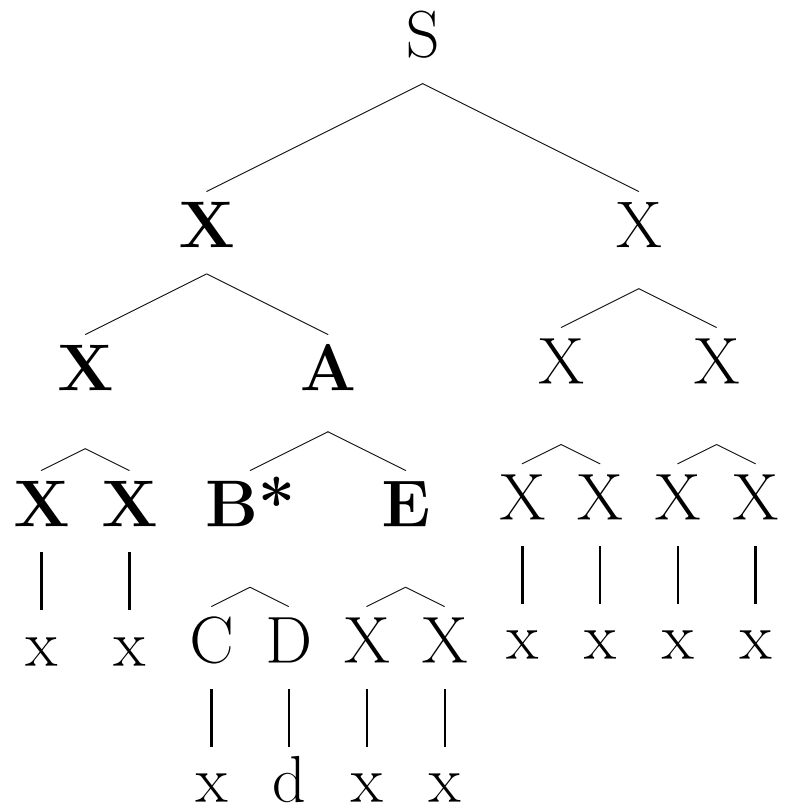
$$D_1 \circ \dots \underbrace{D_k}_{\dots B} \circ \underbrace{D_{k+1}}_{B \dots} \circ \dots \circ D_l$$

D_k can be any supertwig of d_1



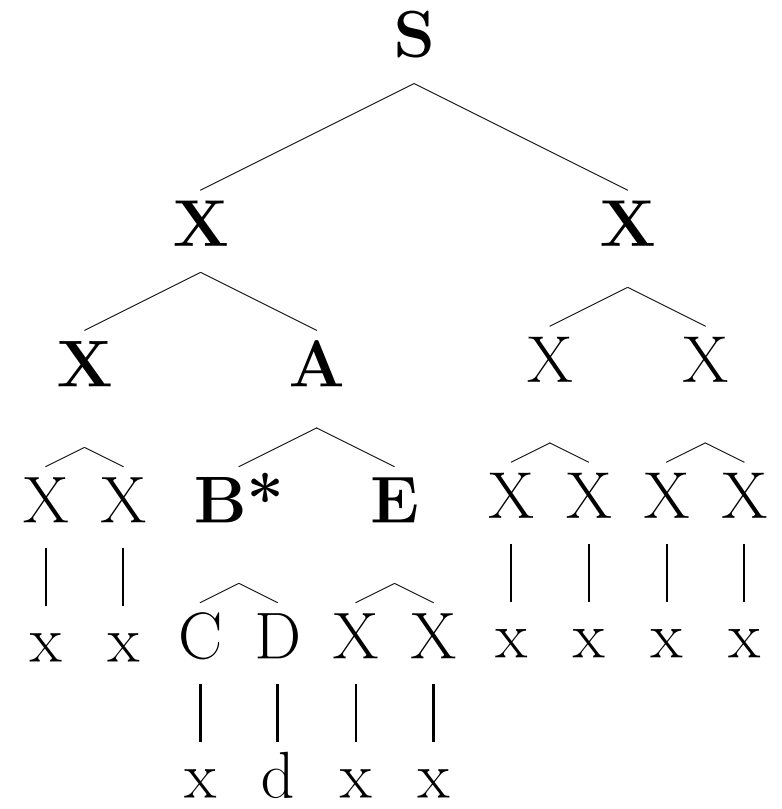
$$D_1 \circ \dots \underbrace{D_k}_{\dots B} \circ \underbrace{D_{k+1}}_{B \dots} \circ \dots \circ D_l$$

D_k can be any supertwig of d_1



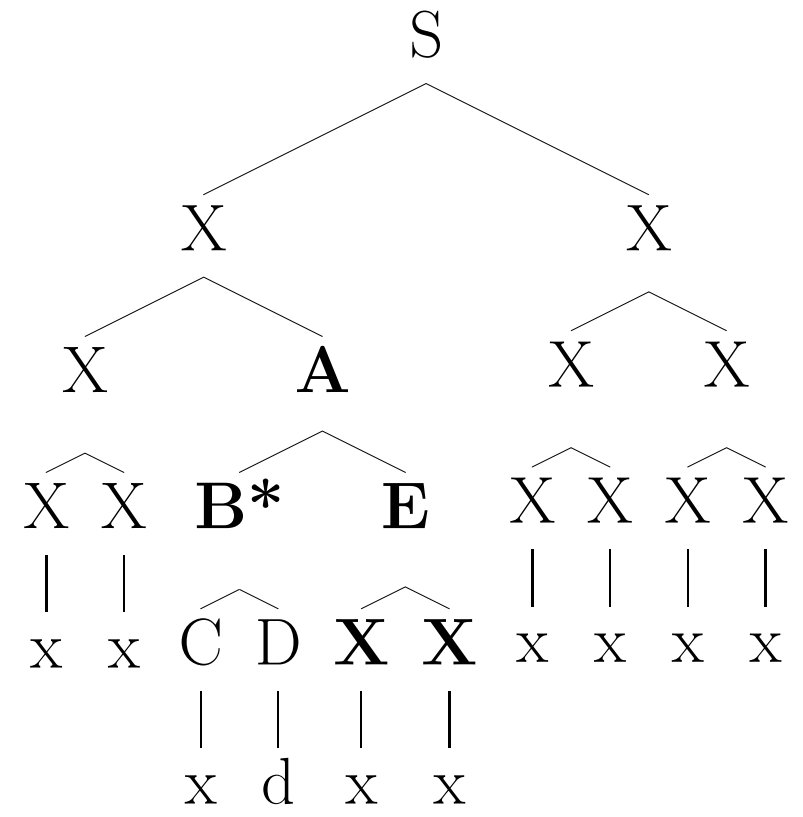
$$D_1 \circ \dots \underbrace{D_k}_{\dots B} \circ \underbrace{D_{k+1}}_{B \dots} \circ \dots \circ D_l$$

D_k can be any supertwig of d_1



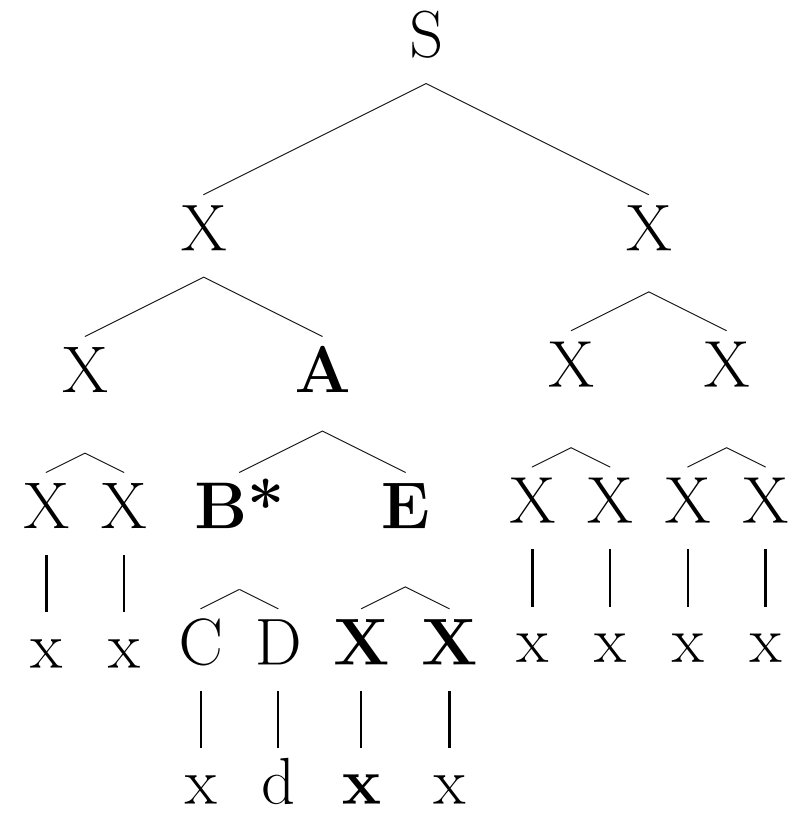
$$D_1 \circ \dots \underbrace{D_k}_{\dots B} \circ \underbrace{D_{k+1}}_{B \dots} \circ \dots \circ D_l$$

D_k can be any E-superprune of d_1



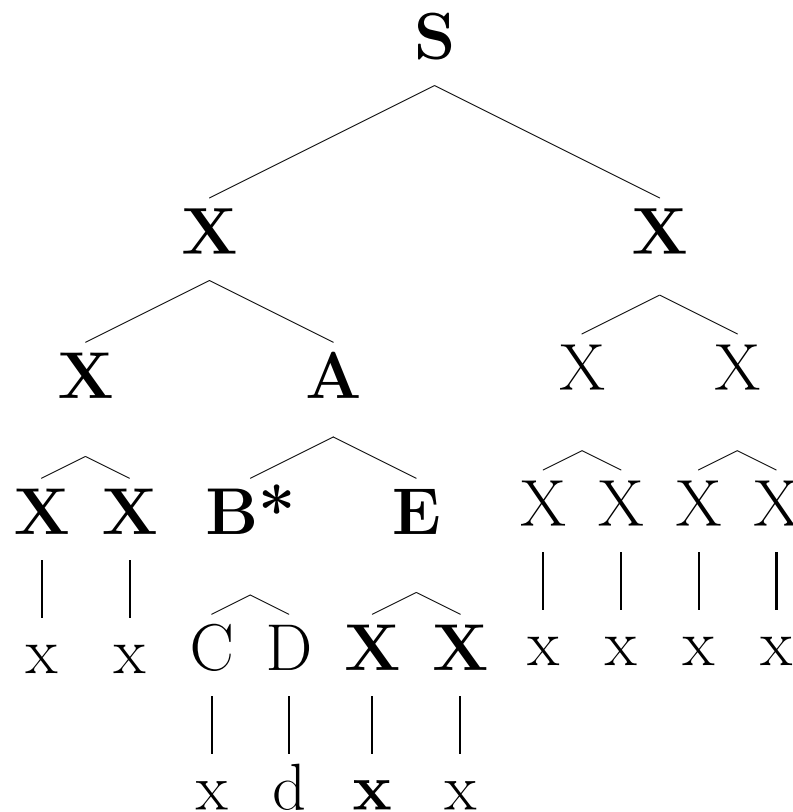
$$D_1 \circ \dots \underbrace{D_k}_{\dots B} \circ \underbrace{D_{k+1}}_{B \dots} \circ \dots \circ D_l$$

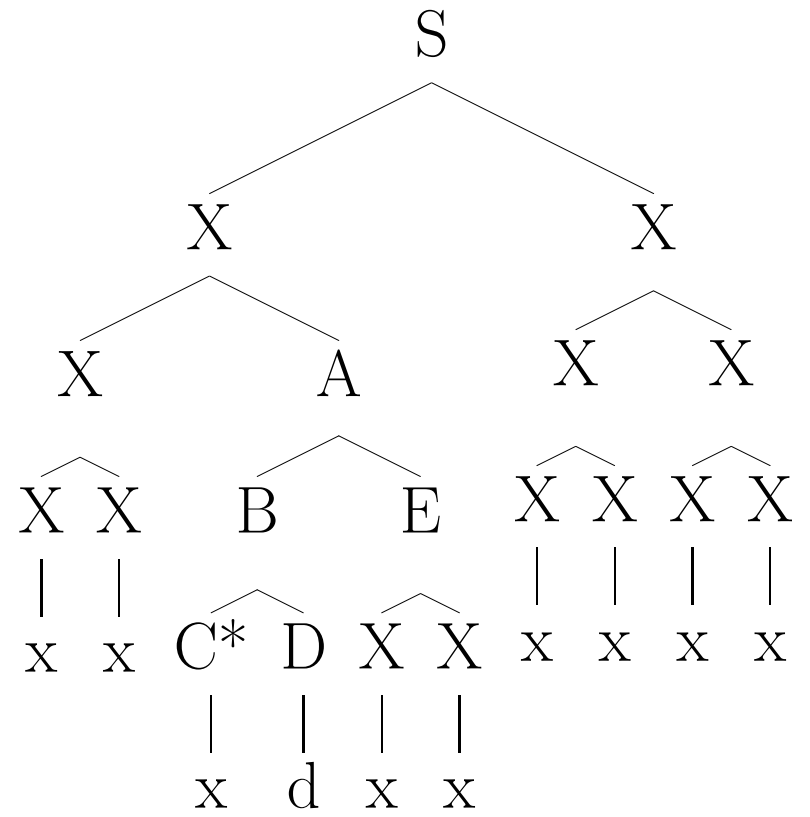
D_k can be any E-superprune of d_1

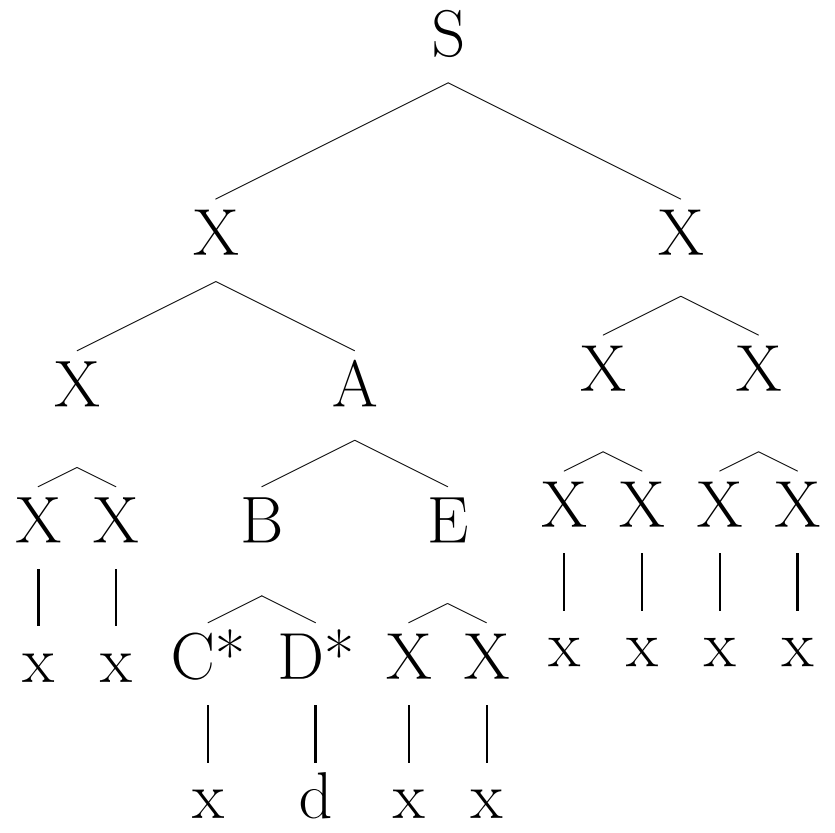


$$D_1 \circ \dots \underbrace{D_k}_{\dots B} \circ \underbrace{D_{k+1}}_{B \dots} \circ \dots \circ D_l$$

D_k can be any E-superprune of any supertwig of d_1







arbitrary subtrees

$$\mathbf{E}[f(t)] = \sum_{d \in D(t)} (\alpha(d)\beta(d))$$

$$\alpha(d) = \sum_{\tau \in \widehat{t\omega}(d_1)} \left(\sum_{\tau' \in \widehat{pr_{x(t)}}(\tau)} u(\tau') \right)$$

$$\beta(d) = \prod_{\substack{t' \in \\ \langle d_2, \dots, d_n \rangle}} \left(\sum_{\tau' \in \widehat{pr_{x(t)}}(t')} w(\tau') \right)$$

arbitrary subtrees

$$\mathbf{E}[f(t)] = \sum_{d \in D(t)} (\alpha(d)\beta(d))$$

$$\alpha(d) = \sum_{\tau \in \widehat{tw}(d_1)} \left(\sum_{\tau' \in \widehat{pr}_{x(t)}(\tau)} u(\tau') \right) = \alpha_{d_1, x(t)}$$

$$\beta(d) = \prod_{\substack{t' \in \\ \langle d_2, \dots, d_n \rangle}} \left(\sum_{\tau' \in \widehat{pr}_{x(t)}(t')} w(\tau') \right)$$

$$= \prod_{\substack{t' \in \\ \langle d_2, \dots, d_n \rangle}} \frac{1}{\sum_{t'' : r(t') = r(t'')} u(t'')} \underbrace{\left(\sum_{\tau' \in \widehat{pr}_{x(t)}(t')} u(\tau') \right)}_{\beta_{t', x(t)}^*}$$

fully lexicalized subtrees

$$\mathbf{E}[f(t)] = \sum_{d \in D(t)} \left[\underbrace{\sum_{\tau \in \widehat{tw}(d_1)} u(\tau)}_{\alpha} \prod_{\substack{\tau' \in \\ \langle d_2, \dots, d_n \rangle}} (w(\tau')) \right]_{\beta}$$

Keeping track of the “supertwig” and “superprune” usage probabilities with every change of these probabilities.

change-usage-probabilities(τ, δ)

$$u(\tau)_+ = \delta$$

for each prune τ' of τ at sites x

$$\beta_{\tau',x}^* + = \delta$$

for each twig τ'' of τ

for each prune τ'' of τ' at sites x

$$\alpha_{\tau'',x} + = \delta$$

```
(setq *G0* '((S A (D z))      2/3)
            ((S A D)          1/3)
            ((D z)            8/9)
            ((D y)            1/9)
            ((A (B x) (C y)) 3/10)
            ((A x)            7/10)))
```

```
>((S A D) #S(INFO :SCORE 334 :FREQ 1000 :STS 334
:SPS ((NIL . 334) ((D) . 666)) :STPS ((NIL . 334) ((D) . 666))))
>((D Z) #S(INFO :SCORE 298 :FREQ 964
:STS 964 :SPS ((NIL . 298)) :STPS ((NIL . 964))))
>((S (A B (C Y)) D) #S(INFO :SCORE 0 :FREQ 289
:STS 0 :SPS NIL :STPS NIL))
```

```
(report '(D Z))
```

```
> (964.0 964)
```

```
(report '(A (B X) C))
```

```
> (289.0 289)
```

```
(report '(S (A (B X) (C Y)) (D Y)))
```

```
> (10.404 10)
```

Conclusions

- For better data-oriented language learning – whether from labeled, partially labeled or unlabeled data – we need good methods to calculate and approximate **likelihood** of the data.
- I have presented an general equation for the likelihood of arbitrary fragments in terms of usage probabilities that can be used in an EM-style estimator from labeled data;
- The operations CHUNK, MERGE and perhaps PRUNE can be used for structure search for maximizing a posterior probability measure in learning from unlabeled data.