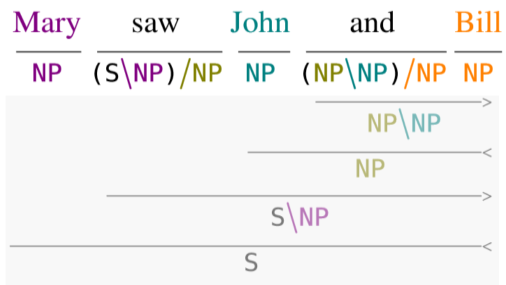
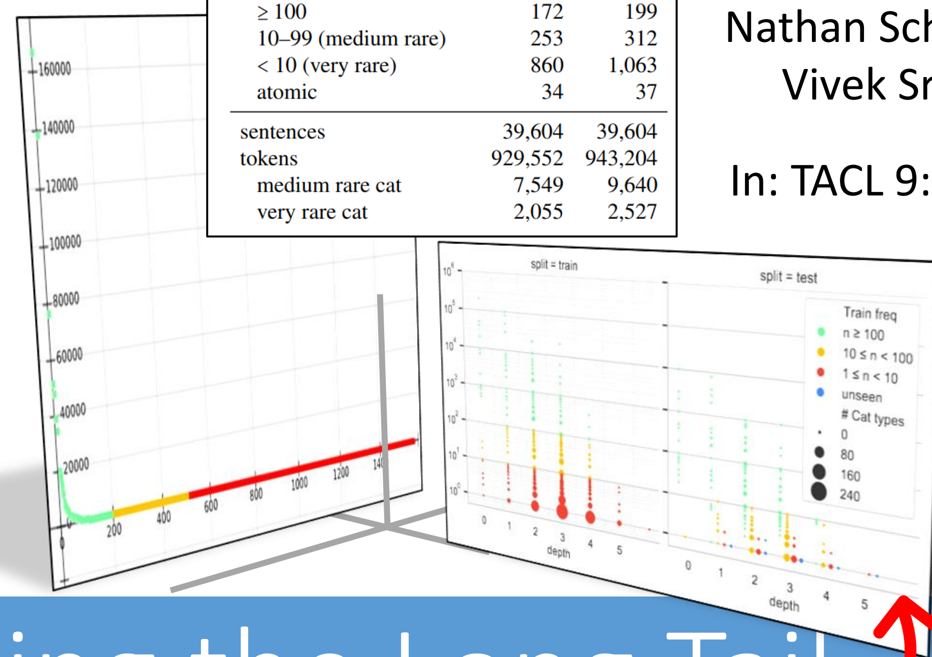


$$\max_{y} p(y|w_k; \boldsymbol{w}), \forall k \in \{0, \ldots, |\boldsymbol{w}|-1\}$$

```
Cat      := FxnCat | AtomCat
FxnCat   := Cat Slash Cat
AtomCat  := N | NP | S | PP | ...
Slash    := / | \
```

	CCGbank	Rebank
cat types	1,285	1,574
≥ 100	172	199
10–99 (medium rare)	253	312
< 10 (very rare)	860	1,063
atomic	34	37
sentences	39,604	39,604
tokens	929,552	943,204
medium rare cat	7,549	9,640
very rare cat	2,055	2,527



Jakob Prange[G]
Nathan Schneider[G]
Vivek Srikumar[U]

In: TACL 9:243-260

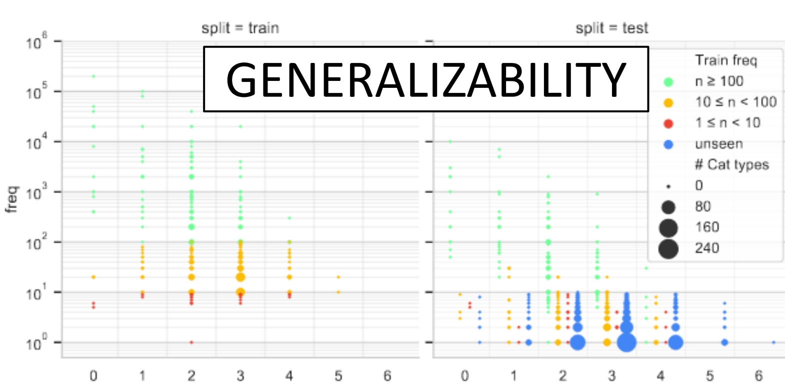
Supertagging the Long Tail with Tree-structured Decoding of Complex Categories

RESULTS

	All	Head	Tail	OOV
Non	**94.83**	**95.27**	27.10	///
Seq	94.48	94.93	34.58	**7.41**
Tree	94.70	95.11	**36.76**	4.94

Tree-structured models improve accuracy on **rare** tags and recover **unseen** ones without losing performance on frequent tags.

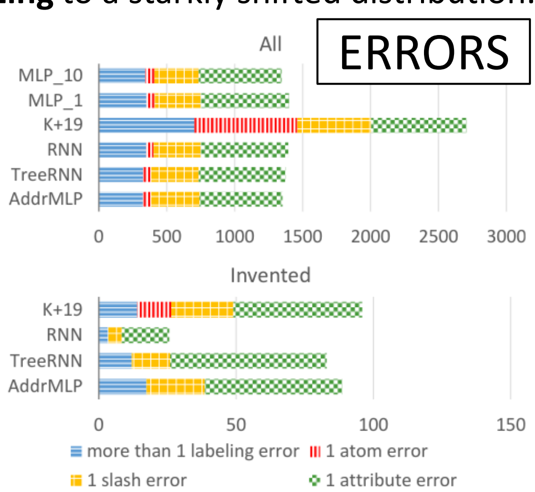
GENERALIZABILITY



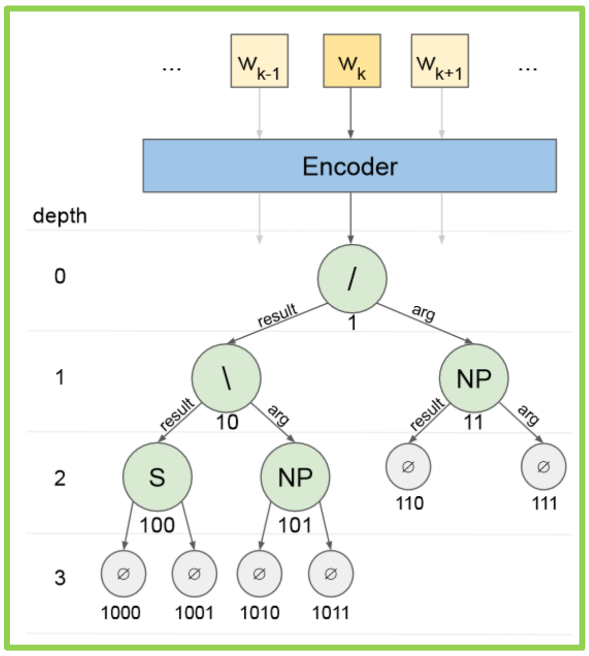
	All	Head	Tail	OOV
Non	88.79	**92.87**	19.29	///
Seq	88.73	92.64	25.11	**11.62**
Tree	**89.01**	92.70	**26.48**	10.96

Their better **across-the-board** performance pays off in **generalizing** to a starkly shifted distribution.

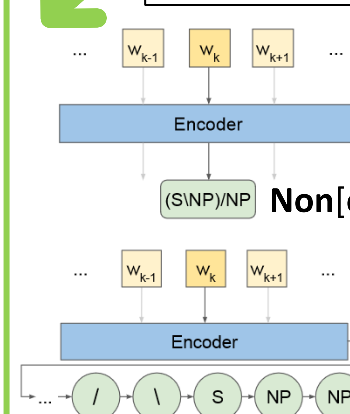
ERRORS



They strike a good **balance** between **inventiveness** and error rate.



BASELINES



$$H_0 = \text{RoBERTa}(\boldsymbol{w})$$
$$\boldsymbol{o}_{k,i} = \text{MLP}(\boldsymbol{h}_{k,i})$$

MODELING

TreeRNN: $\boldsymbol{h}_{k,c(i)} = \text{GRU}_c(\text{Embed}(y_{k,i}), \boldsymbol{h}_{k,i})$

AddrMLP: $\boldsymbol{h}_{k,i} = \boldsymbol{h}_{k,0} + \text{Linear}(\text{Features}(i, \boldsymbol{y}_{k,\text{anc}(i)}))$

EXAMPLES

	orders began	piling	up
Gold		**(S[ng]\NP)/PR**	**PR**
MLP_10		S[ng]\NP	ADV
MLP_1		S[ng]\NP	ADV
K+19		S[ng]\NP	ADV
RNN		(S[ng]\NP)/PP	S[adj]\NP
AddrMLP		✓	✓

	garnered	from	1984 to 1986
Gold	**(S[pss]\NP)**	**(ADV/ADV)/NP**	
MLP_10	✓	✓	
MLP_1	✓	✓	
K+19	✓	✓	
RNN	✓	✓	
AddrMLP	(S[pss]\NP)/PP	(PP/ADV)/NP	

	Why	constructive	?
Gold	**S[wq]/(S[adj]\NP)**	**S[adj]\NP**	
MLP_10	<UNKNOWN>	✓	
MLP_1	(S/S)/(S[adj]\NP)	✓	
K+19	✓	✓	
RNN	✓	✓	
AddrMLP	✓	✓	

They capture inter-category relations without being explicitly trained to do so.
Errors are **self-consistent**.

EFFICIENCY

Model	Params *millions*	Train time *hours*	Infer speed *sents/s*
Nonconstructive Classification			
MLP_10	2.0	9	191
MLP_1	2.4	11	195
Constructive: Sequential			
K+19	11.8	120	0.3
RNN	4.8	68	135
Constructive: Tree-structured			
TreeRNN	8.3	10	125
AddrMLP	1.3	10	126

The best tree-structured model, AddrMLP, is the **smallest** model, and similarly **fast** as nonconstructive ones.

[G]NERT @ Georgetown [U]U of Utah NLP