OPEN-SESAME

"Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold"

Swabha Swayamdipta et al. (arXiv, 2017)




EXAMPLE

Hoover Dam **played** *play.v* a **major** *major.a* role in **preventing** *prevent.v* Las Vegas from **drying up** *dry up.v*

- Performer
- PERFORMERS_AND_ROLES
- Role
- Performance
- IMPORT-ANCE
- Factor
- Undertaking
- Preventing_cause
- THWARTING
- Protagonist
- Entity
- Action
- BECOMING_DRY

PROBLEM

- Learning to identify exact boundaries of argument spans is *hard*.

- Current pruning heuristics (based on automatic dependency parses) bound recall below 72.6%.

- But without pruning, label imbalance (# of spans that *are not* args vs. # of spans that *are* args) is extreme!

PROPOSED SOLUTION

- Throw away rule-based syntactic pruning. Improve model to learn "soft" pruning instead.
  - Only hard pruning is based on length (spans > 20 tokens).
- Recall-oriented training.

IMPROVING MODEL

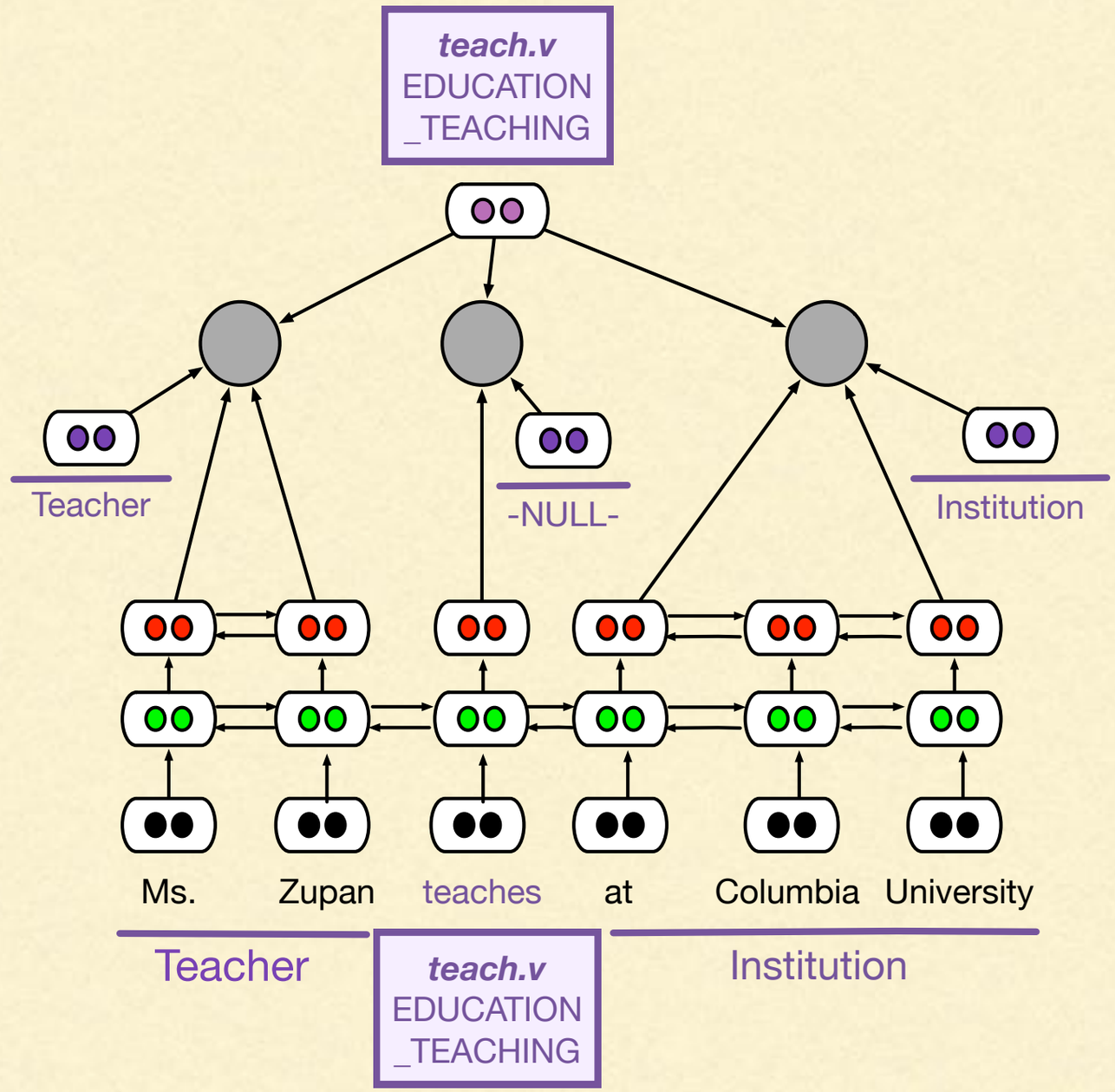
- SEMAFOR is a **linear** model over **hand-engineered** features
- How might we improve?

SEGMENTAL RNN *(Kong et al., ICLR 2016)*

- Models probability of a labeled segmentation ***s***, conditioned on a sequence ***x***.

- *token BiLSTM* encodes each token in context.
- *span BiLSTM* encodes each candidate span (based on *token BiLSTM*).
- A span's score is based on the span encoding, a frame encoding, and a role label encoding.
- Globally normalized, using dynamic programming.

SEGMENTAL RNN



SOFTMAX-MARGIN

- Incorporates a cost into the loss that heavily penalizes recall errors (α=2).

$$\text{loss}(\boldsymbol{x}, \boldsymbol{s}^*) = -\log \frac{\exp \phi(\boldsymbol{s}^*, \boldsymbol{x})}{Z}$$

$$Z = \sum_{\boldsymbol{s}} \exp \{\phi(\boldsymbol{s}, \boldsymbol{x}) + \text{cost}(\boldsymbol{s}, \boldsymbol{s}^*)\}$$

$$\text{cost}(\boldsymbol{s}, \boldsymbol{s}^*) = \alpha \text{FN}(\boldsymbol{s}, \boldsymbol{s}^*) + \text{FP}(\boldsymbol{s}, \boldsymbol{s}^*)$$

RE-INCORPORATING SYNTAX

- Three approaches
- Features from dependency parse
- Features from phrase-based parse
- "Scaffold" - a multitask objective, used during training, then thrown away at test time

RE-INCORPORATING SYNTAX

- Syntax features are both:
  - hand-engineered (e.g. `#incoming_deps`, `lca`, ...), and
  - neural (`path_lstm` a la *Roth & Lapata, arXiv 2016*)

RE-INCORPORATING SYNTAX

- Scaffold task:
  - Learn to predict whether or not a span is a constituent in Penn TreeBank
  - Shares the BiLSTM encoder with the SegRNN
  - single hyperparameter determines how much to care about scaffold task vs. main task

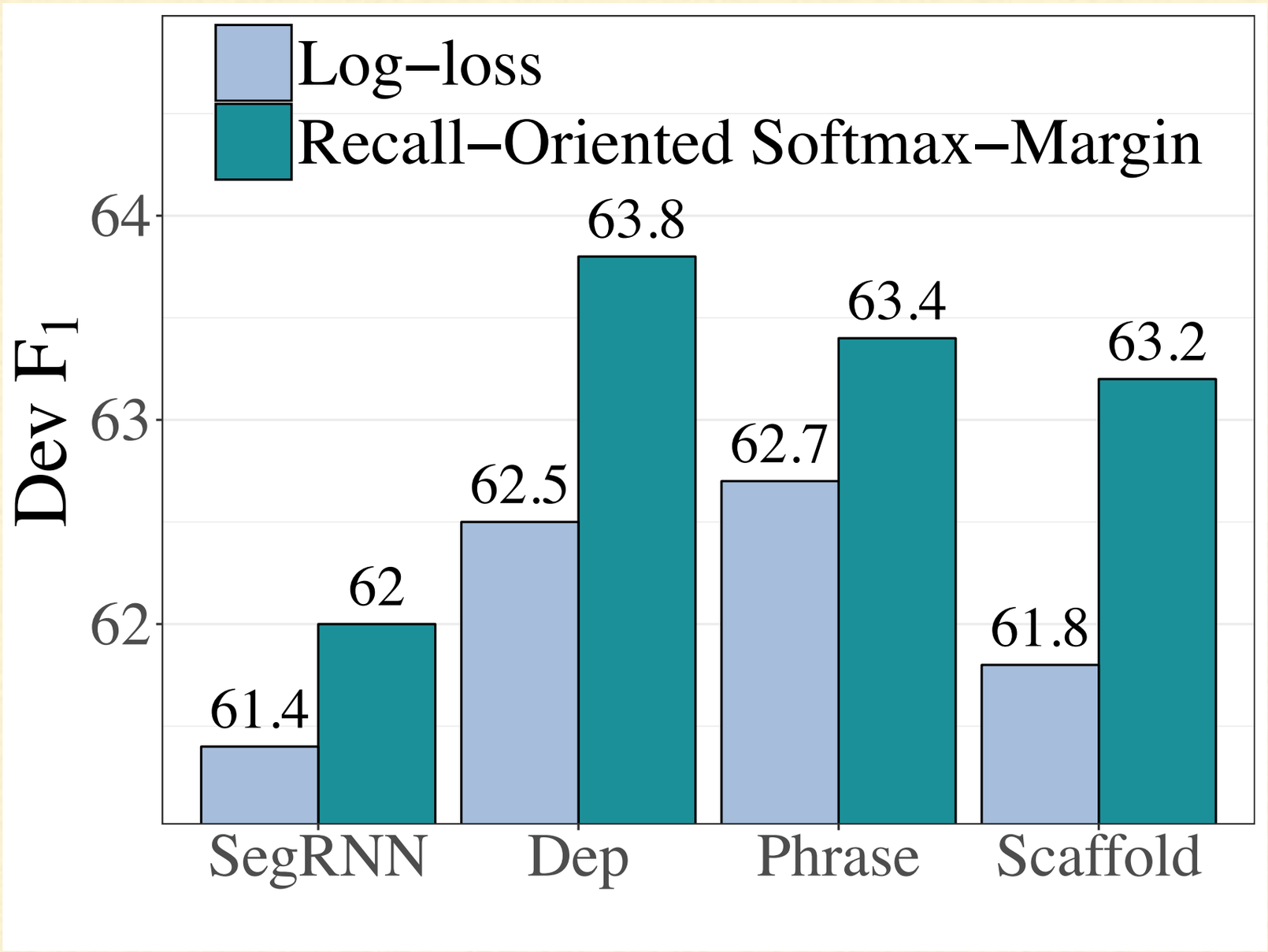
EFFECT OF SYNTAX

	Arg Id (gold frames)		
	P	R	F
Kshirsagar 2015	66.0	60.4	63.1
Yang 2017	70.2	60.2	65.5
no syntax (ens)	69.5	63.6	66.4
dep feats (ens)	70.2	65.6	67.8
phrase feats (ens)	71.7	66.3	**68.9**
scaffold (ens)	72.0	65.0	68.3

USING PREDICTED FRAMES

	Frame+Arg Id (predicted frames)		
	P	R	F
Kshirsagar '15			67.9
Roth '17 (ens)	74.8	65.5	69.9
FitzGerald '15 (ens)	66.4	66.4	**70.9**
Yang '17 (joint)	78.8	74.5	**76.6**
no syntax (ens)	70.5	69.4	69.9
dep feats (ens)	70.7	70.4	70.6
phrase feats (ens)	71.2	70.5	**70.9**
scaffold (ens)	71.5	69.9	70.7

EFFECT OF RECALL-ORIENTED TRAINING



FUTURE WORK

- Multitask Learning, a la *Peng et al, ACL 2017*
- Joint Frame Id / Arg Id, a la *Yang & Mitchell, EMNLP 2017*
- Multilingual? (Please may we use your beautiful data?)

THANKS!

- Code available at:
  - https://github.com/Noahs-ARK/open-sesame
- TODO: Target identification
- TODO: update demo
  - http://demo.ark.cs.cmu.edu/parse/