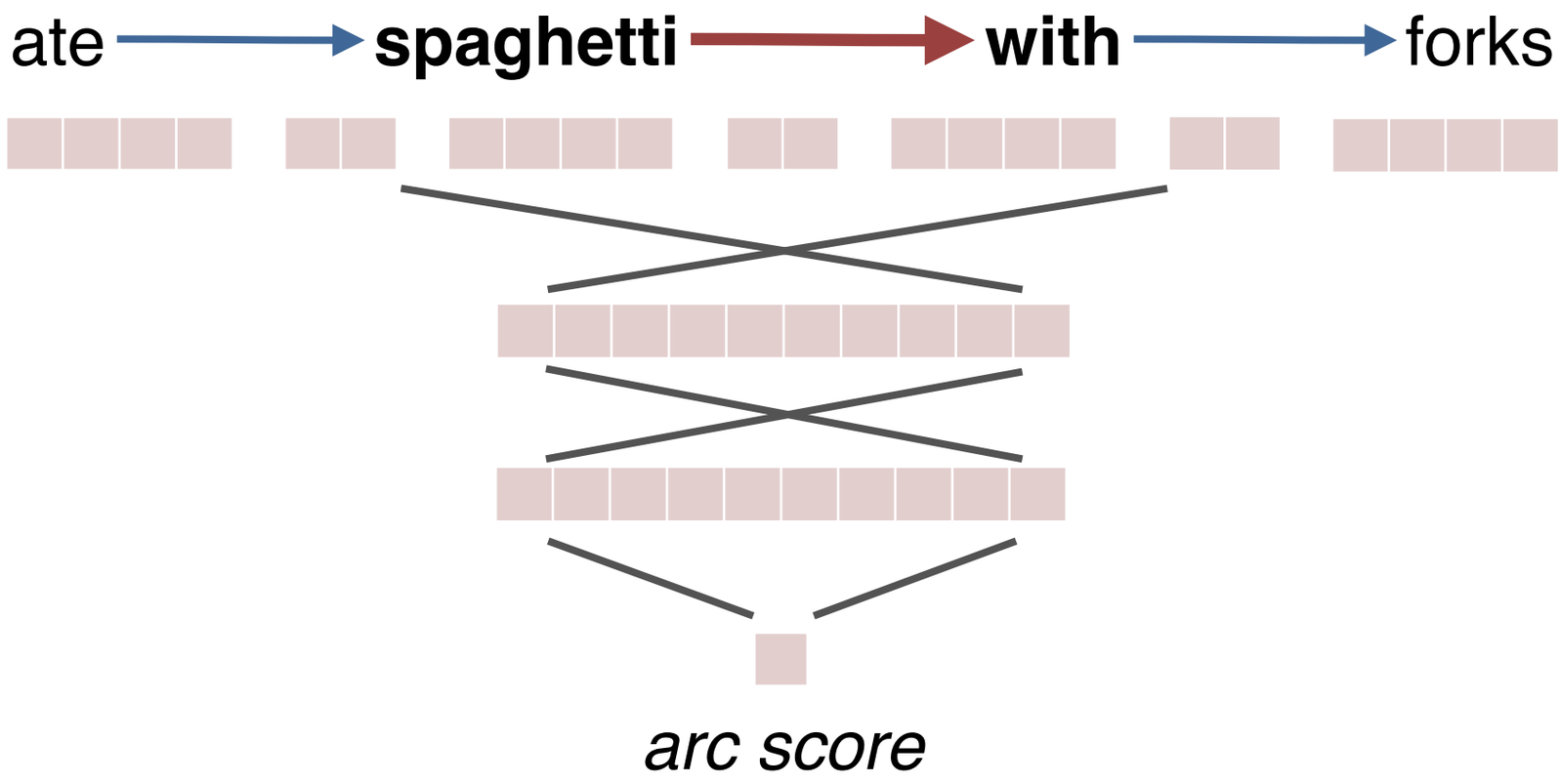
Parsing with neural scoring and randomized greedy inference

Tom Yan and Leon Lin

Feedforward scoring

Structured prediction with semi-local score breakdown:

$$\text{score}(x, y) = \sum_{(h,m) \in y} \text{arc-score}(x, h, m)$$

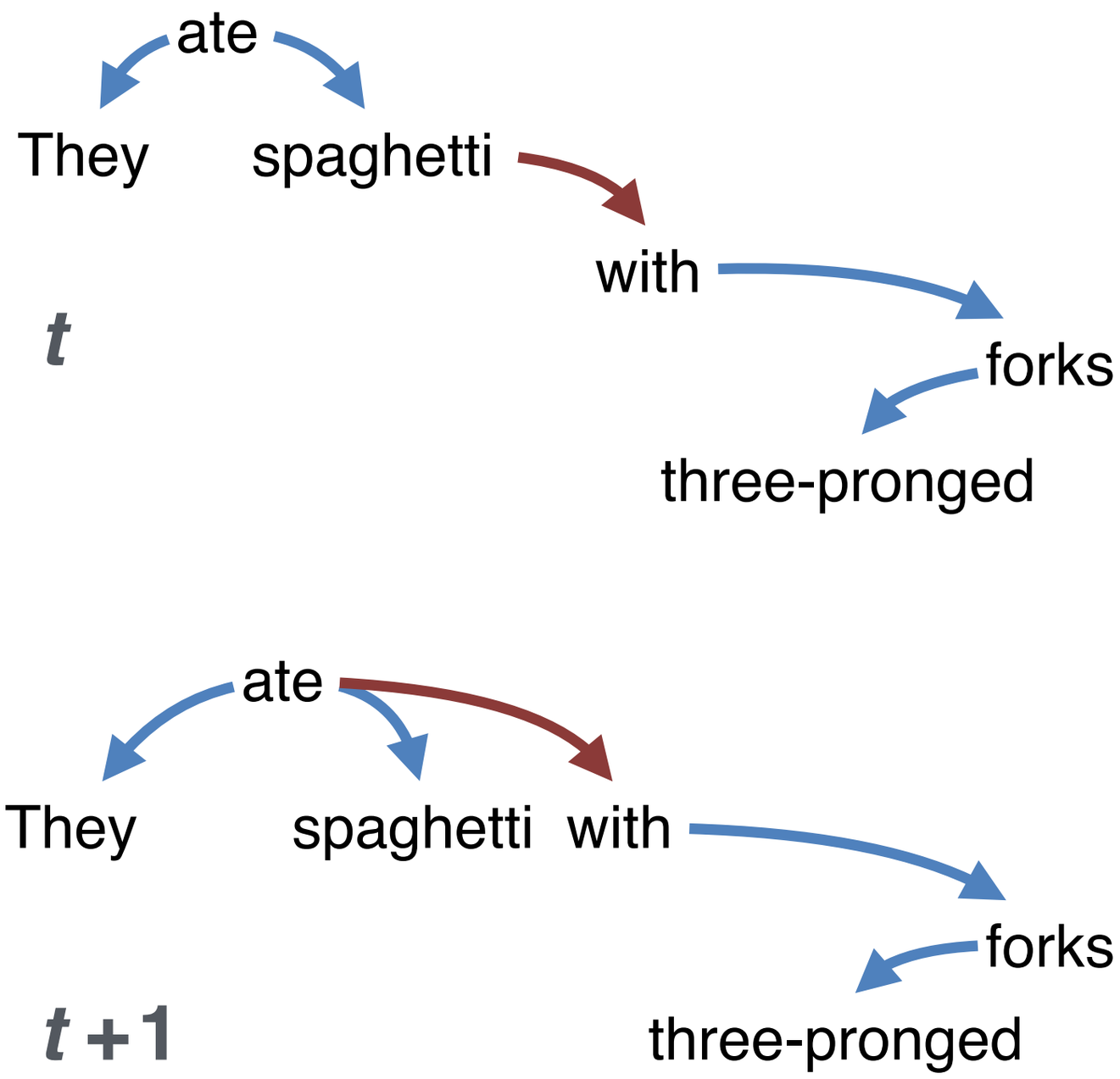


Greedy inference

Iterate over words from bottom to top of tree, each time reassigning the head to be score-maximizing.

Repeat until at a local maximum.

See Yuan Zhang and Tao Lei, EMNLP (2014).



Max-margin training

We train scoring end-to-end with inference. We want

$$\text{score}(x, y) + L(y, y_{\text{gold}}) \leq \text{score}(x, y_{\text{gold}})$$

Find the worst violating parse tree y using greedy inference with random restarts, and train against the gradient

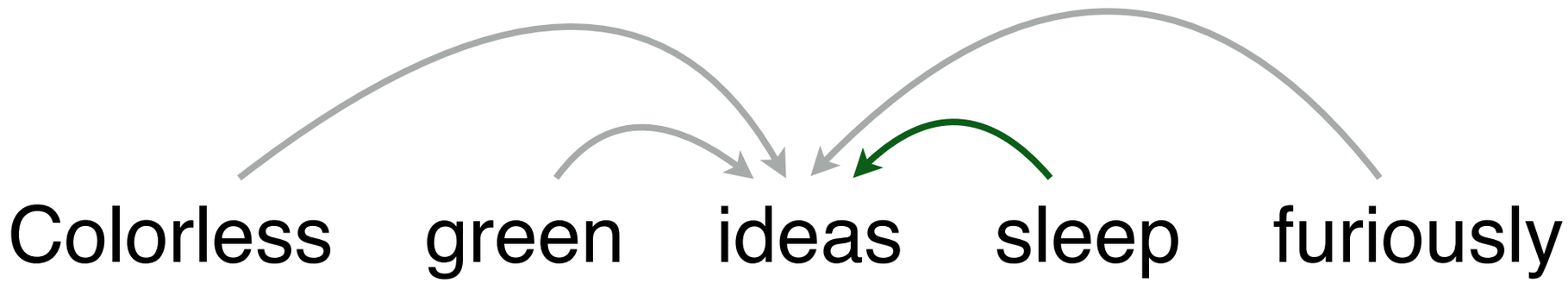
$$\frac{\partial}{\partial \theta} \left[\text{score}(x, y) - \text{score}(x, y_{\text{gold}}) \right]$$

where θ are the network weights.

Per-arc training

As a baseline, we train the network to simply classify arcs as present (1) or not present (0).

Use negative sampling to obtain a negative example for each modifier.



Arc feature choices

Part of speech

- One-hot*
- Low-dim. embeddings

Word form

- One-hot
- Word embeddings

Arc length

- One-hot*
- Single coordinate

Context

- Neighboring words*
- Other arcs (higher-order features)

Training choices

Update frequency

- Batch
- Minibatch*
- Sentence by sentence

Optimization step size

- Constant*
- AdaGrad
- Momentum

Convergence

- Early stopping* (based on dev set)

Unlexicalized results

Unlabeled attachment scores

	Unigram, weaker inference	Unigram, stronger inference	5-gram
Per-arc classification	**69.54**	69.25	79.12
Max-margin training	68.30	**71.28**	**81.39**

(Dependency-annotated corpus based on PTB)

Future work

- Lexicalization
- Higher-order features
- Pruning trees for speed
- Pre-training networks with per-arc method before using end-to-end max-margin training
- Can the scoring function adapt to different inference algorithms during end-to-end training?