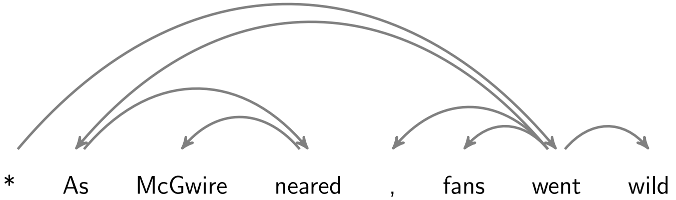
Vine Pruning for Efficient Multi-Pass Dependency Parsing

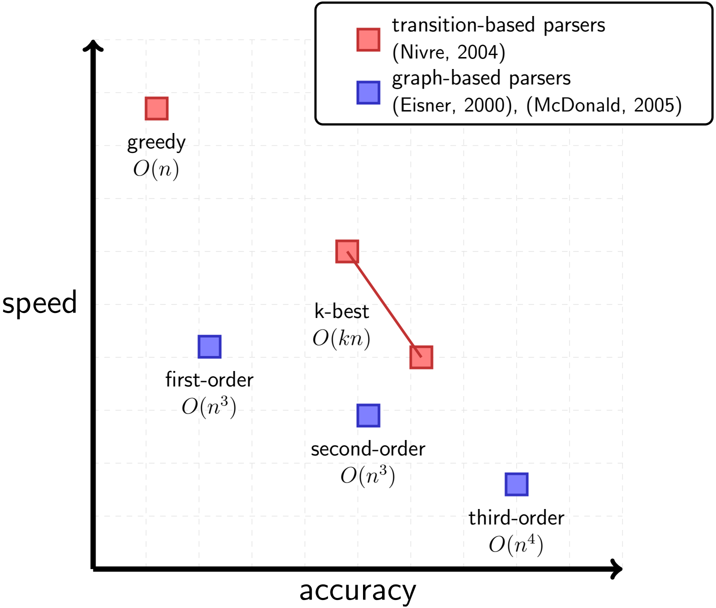
Alexander M. Rush and Slav Petrov

Dependency Parsing

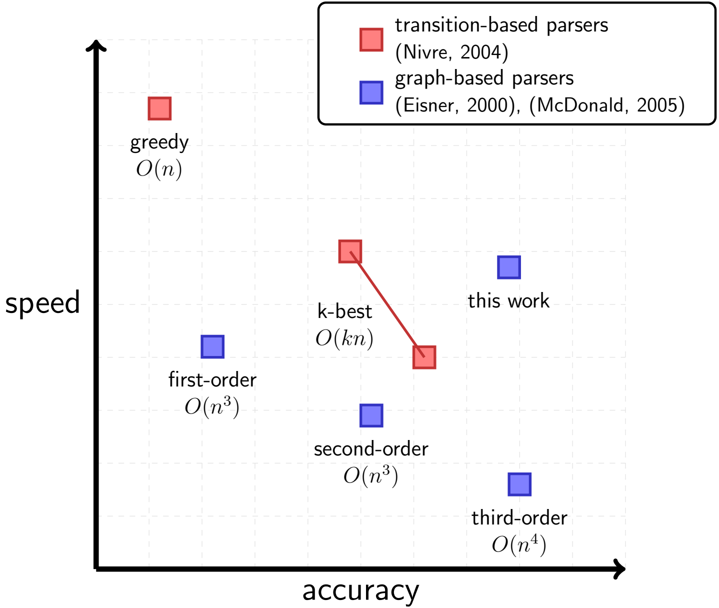
* As McGwire neared , fans went wild



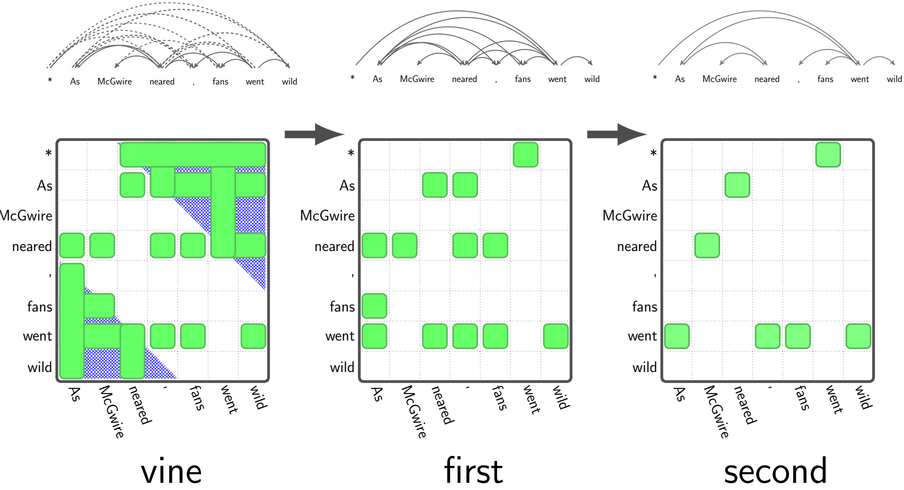
Styles of Dependency Parsing



Styles of Dependency Parsing



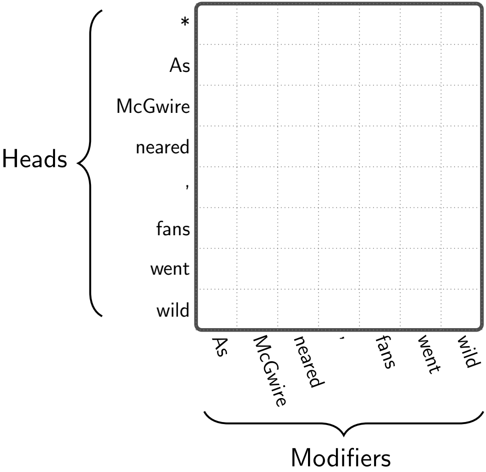
Preview: Coarse-to-Fine Cascades



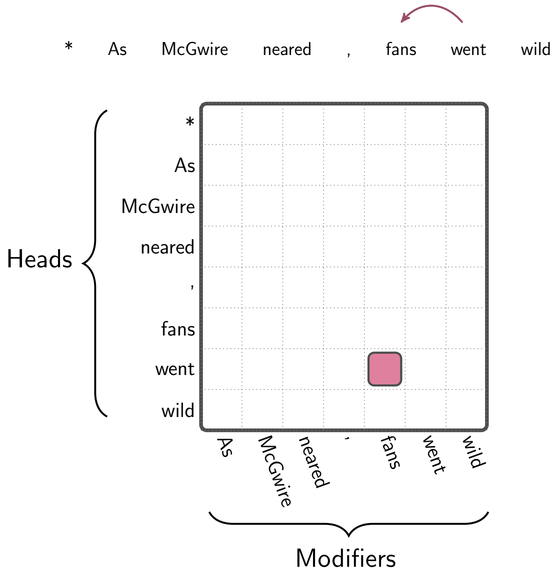
linear-size dependency representation

Representation

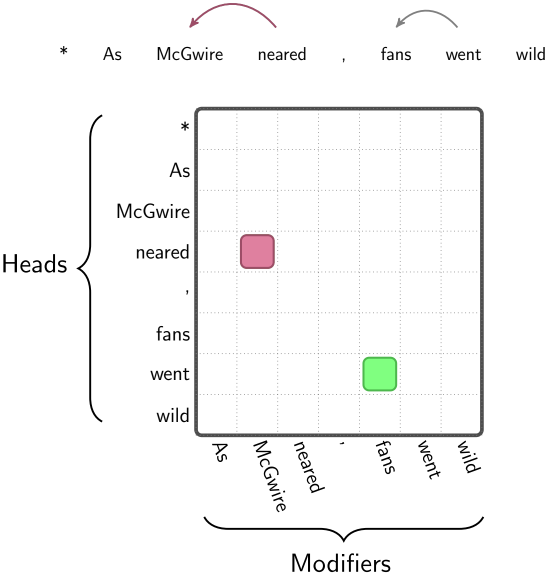
* As McGwire neared , fans went wild



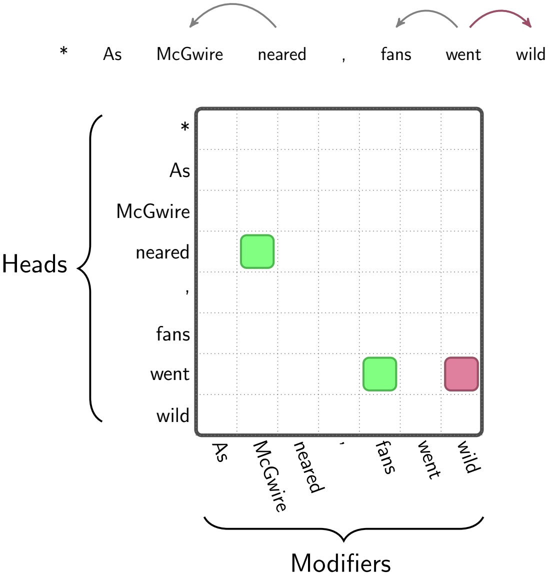
Representation



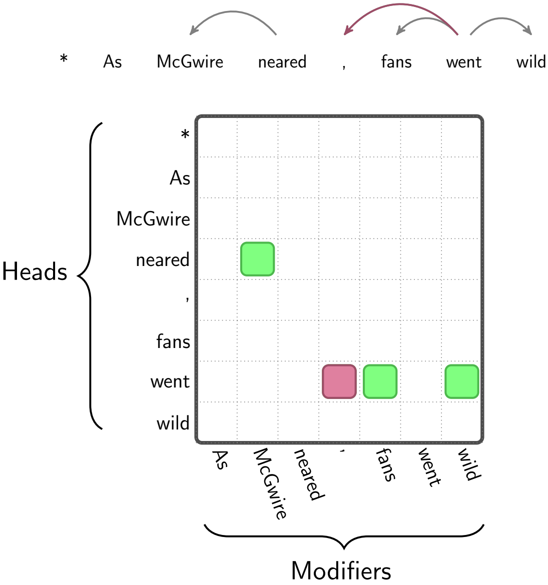
Representation



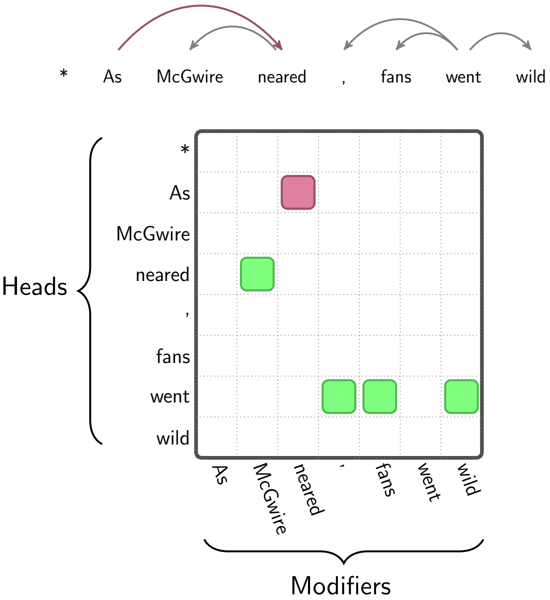
Representation



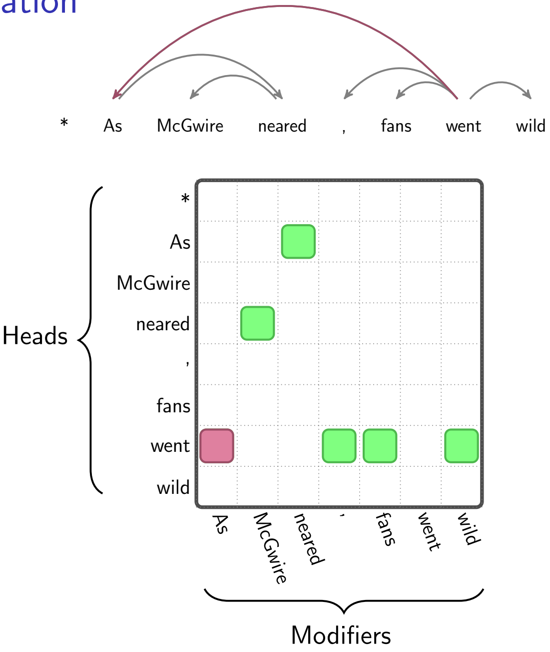
Representation



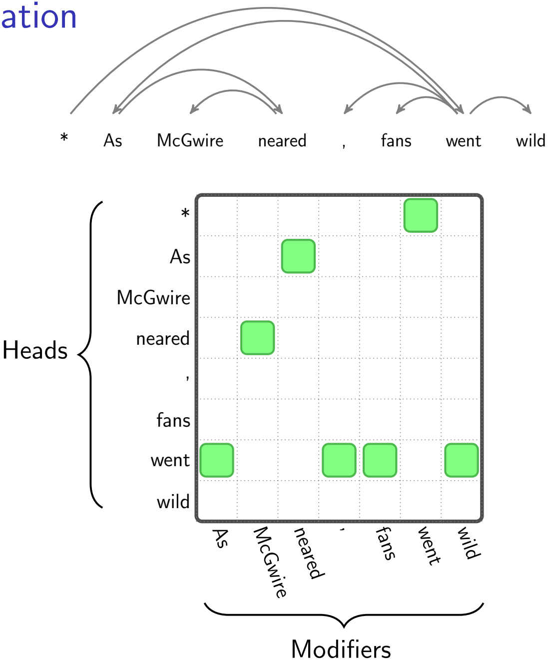
Representation



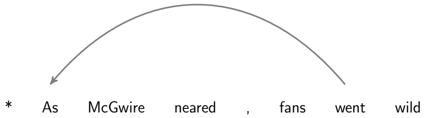
Representation

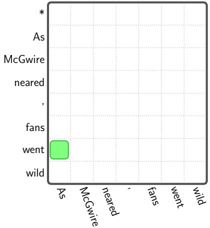


Representation

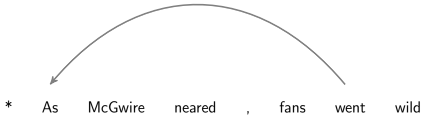


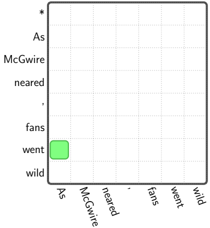
First-Order Feature Calculation





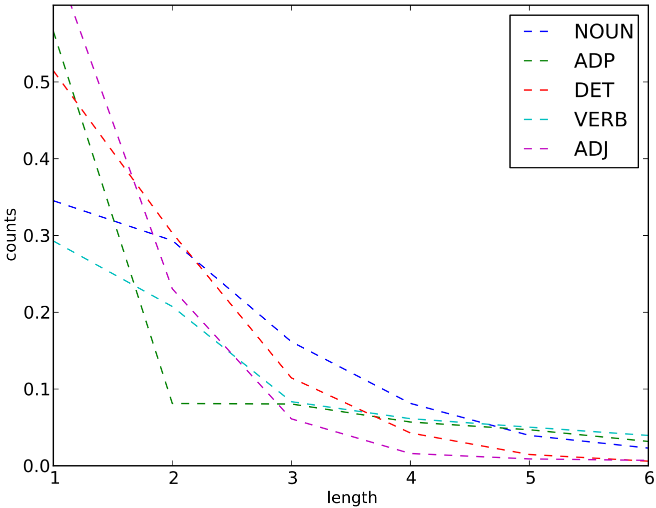
First-Order Feature Calculation



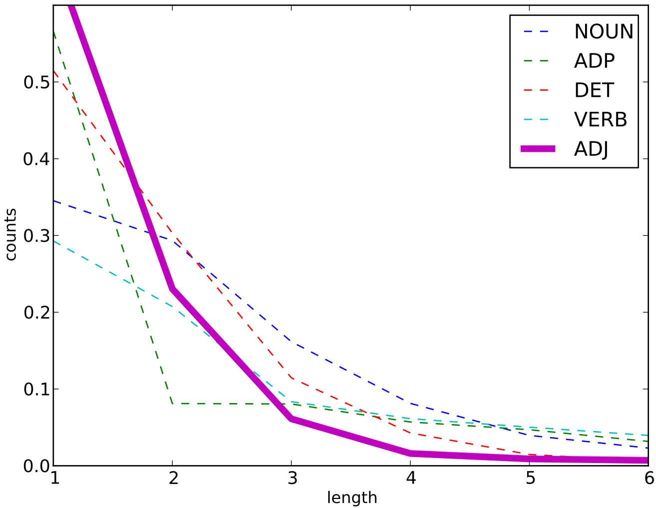


[went]	[VBD]	[As]	[ADP]	[went]
[VERB]	[As]	[IN]	[went, VBD]	[As, ADP]
[went, As]	[VBD, ADP]	[went, VERB]	[As, IN]	[went, As]
[VERB, IN]	[VBD, As, ADP]	[went, As, ADP]	[went, VBD, ADP]	[went, VBD, As]
[ADJ, *, ADP]	[VBD, *, ADP]	[VBD, ADJ, ADP]	[VBD, ADJ, *]	[NNS, *, ADP]
[NNS, VBD, ADP]	[NNS, VBD, *]	[ADJ, ADP, NNP]	[VBD, ADP, NNP]	[VBD, ADJ, NNP]
[NNS, ADP, NNP]	[NNS, VBD, NNP]	[went, left, 5]	[VBD, left, 5]	[As, left, 5]
[ADP, left, 5]	[VERB, As, IN]	[went, As, IN]	[went, VERB, IN]	[went, VERB, As]
[JJ, *, IN]	[VERB, *, IN]	[VERB, JJ, IN]	[VERB, JJ, *]	[NOUN, *, IN]
[NOUN, VERB, IN]	[NOUN, VERB, *]	[JJ, IN, NOUN]	[VERB, IN, NOUN]	[VERB, JJ, NOUN]
[NOUN, IN, NOUN]	[NOUN, VERB, NOUN]	[went, left, 5]	[VERB, left, 5]	[As, left, 5]
[IN, left, 5]	[went, VBD, As, ADP]	[VBD, ADJ, *, ADP]	[NNS, VBD, *, ADP]	[VBD, ADJ, ADP, NNP]
[NNS, VBD, ADP, NNP]	[went, VBD, left, 5]	[As, ADP, left, 5]	[went, As, left, 5]	[VBD, ADP, left, 5]
[went, VERB, As, IN]	[VERB, JJ, *, IN]	[NOUN, VERB, *, IN]	[VERB, JJ, IN, NOUN]	[NOUN, VERB, IN, NOUN]
[went, VERB, left, 5]	[As, IN, left, 5]	[went, As, left, 5]	[VERB, IN, left, 5]	[VBD, As, ADP, left, 5]
[went, As, ADP, left, 5]	[went, VBD, ADP, left, 5]	[went, VBD, As, left, 5]	[ADJ, *, ADP, left, 5]	[VBD, *, ADP, left, 5]
[VBD, ADJ, ADP, left, 5]	[VBD, ADJ, *, left, 5]	[NNS, *, ADP, left, 5]	[NNS, VBD, ADP, left, 5]	[NNS, VBD, *, left, 5]
[ADJ, ADP, NNP, left, 5]	[VBD, ADP, NNP, left, 5]	[VBD, ADJ, NNP, left, 5]	[NNS, ADP, NNP, left, 5]	[NNS, VBD, NNP, left, 5]
[VERB, As, IN, left, 5]	[went, As, IN, left, 5]	[went, VERB, IN, left, 5]	[went, VERB, As, left, 5]	[JJ, *, IN, left, 5]
[VERB, *, IN, left, 5]	[VERB, JJ, IN, left, 5]	[VERB, JJ, *, left, 5]	[NOUN, *, IN, left, 5]	[NOUN, VERB, IN, left, 5]

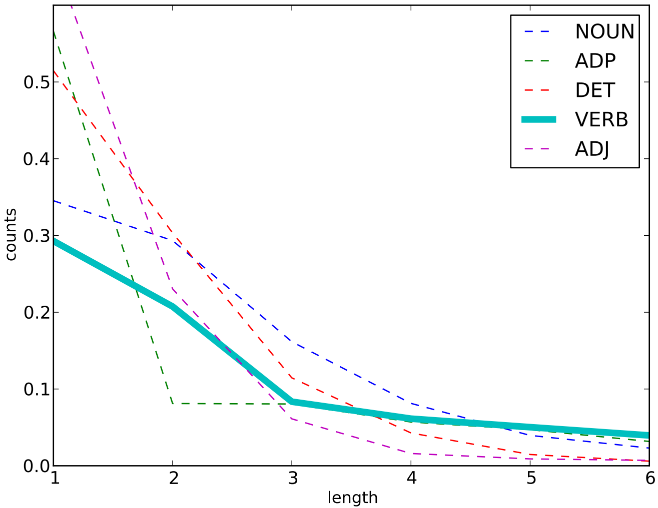
Arc Length By Part-of-Speech



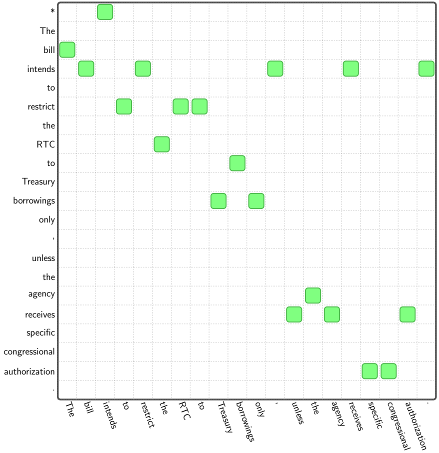
Arc Length By Part-of-Speech



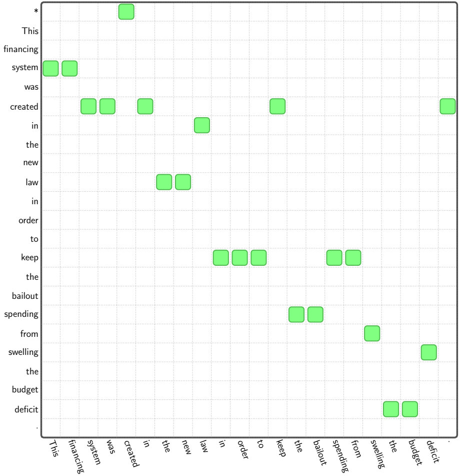
Arc Length By Part-of-Speech



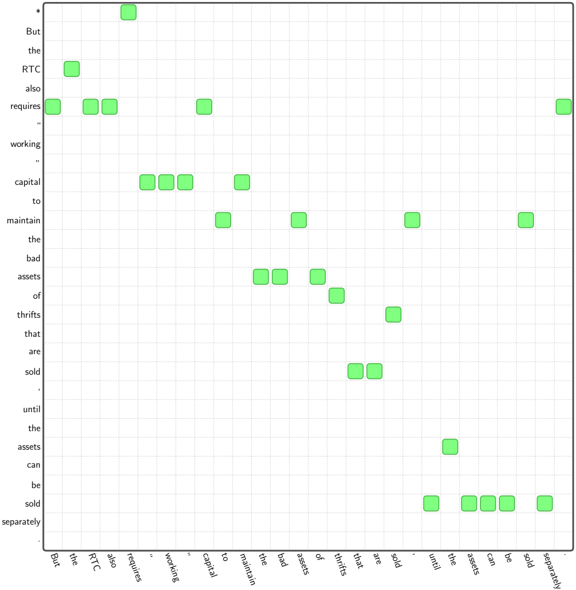
Arc Length Examples



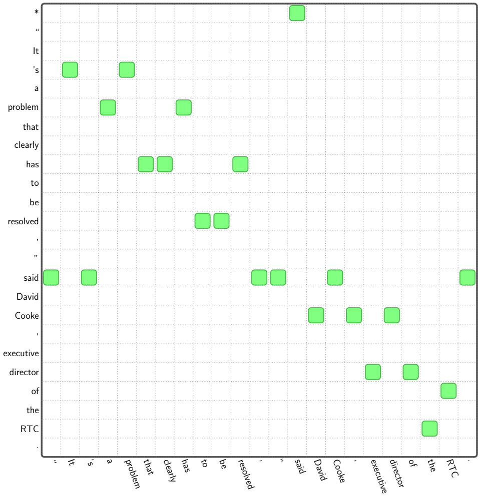
Arc Length Examples



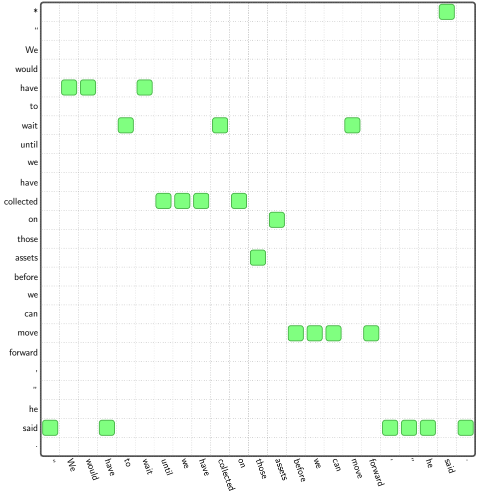
Arc Length Examples



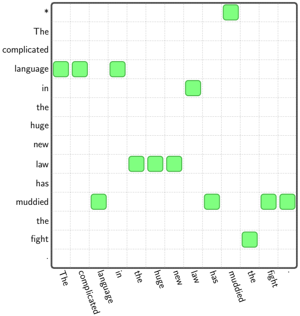
Arc Length Examples



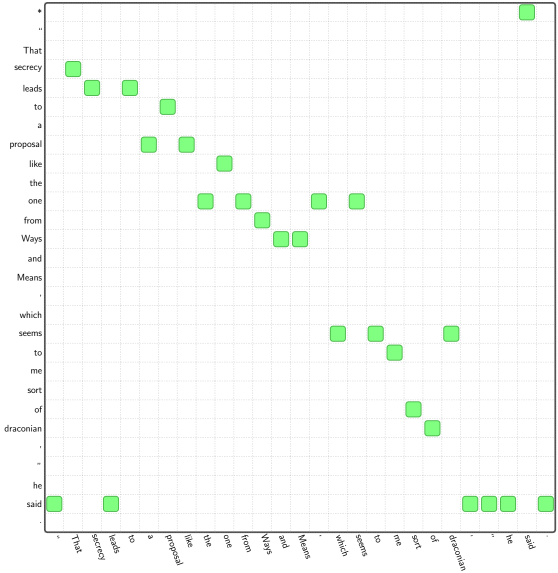
Arc Length Examples



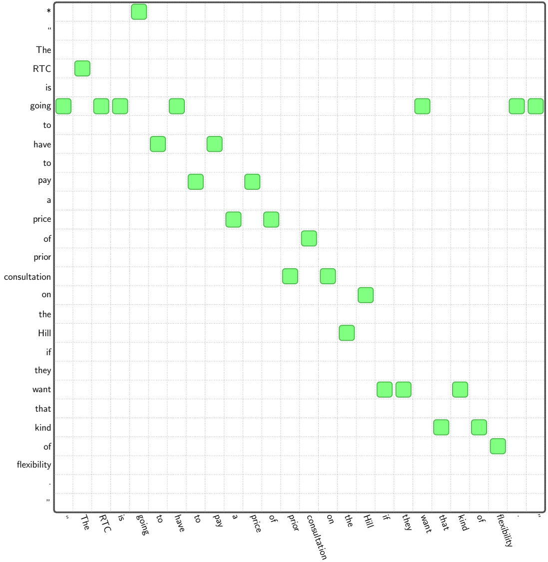
Arc Length Examples



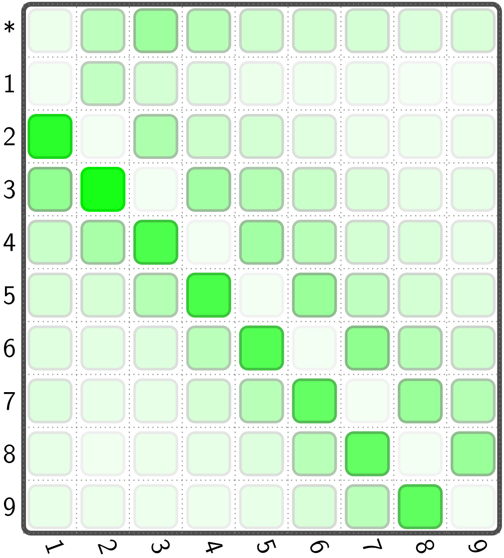
Arc Length Examples



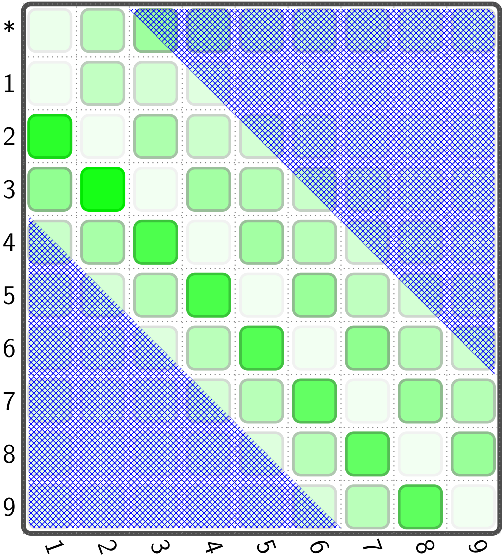
Arc Length Examples



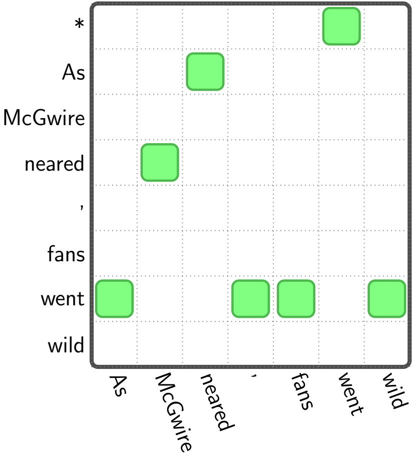
Arc Length Heat Map



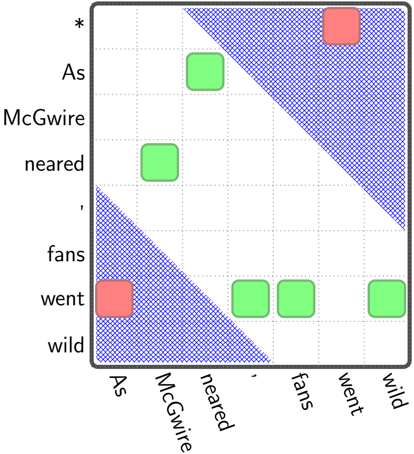
Arc Length Heat Map



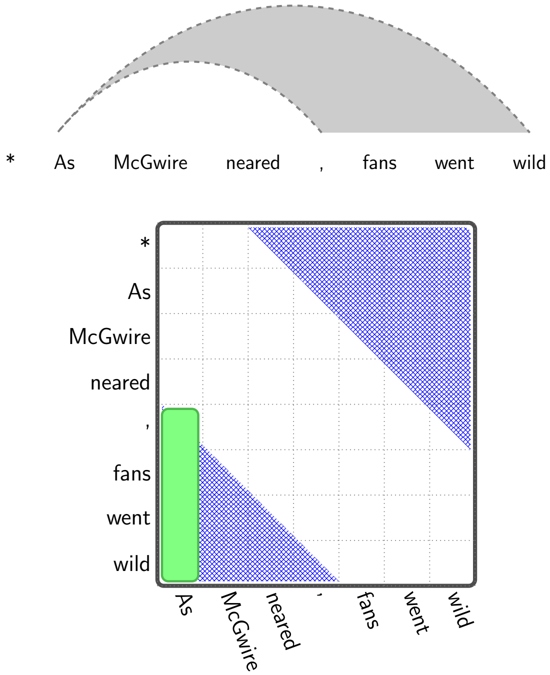
Banded Matrix



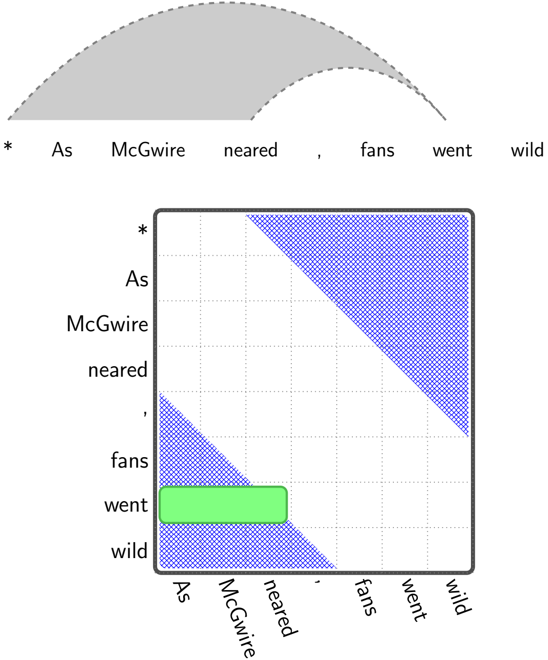
Banded Matrix



Outer Arc

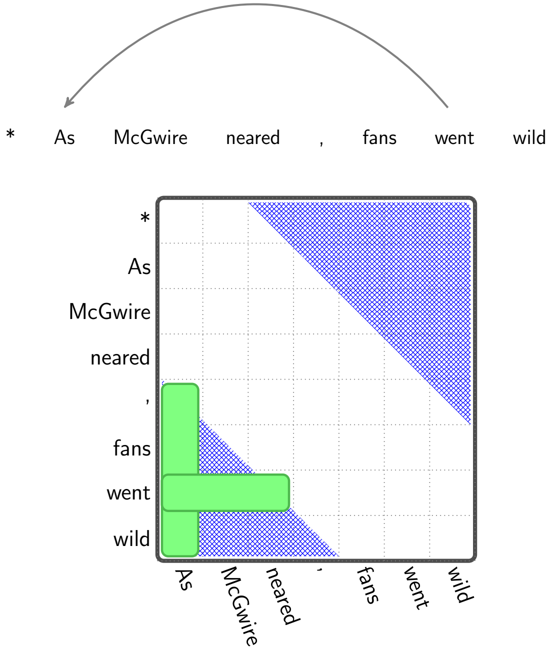


Outer Arc



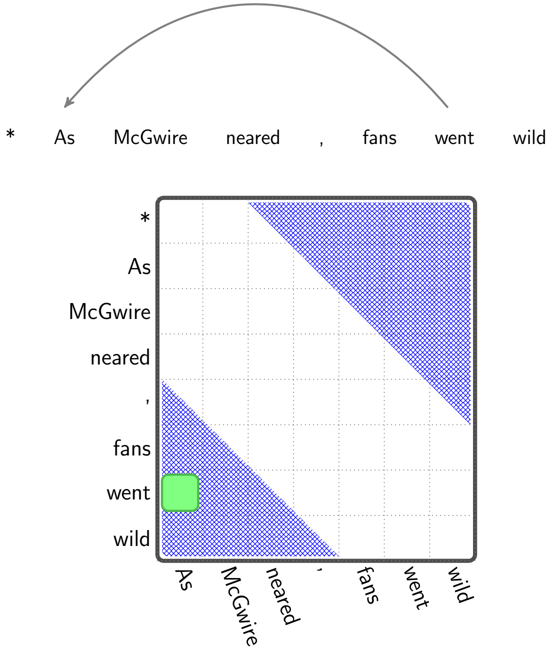
* As McGwire neared , fans went wild
*
As
McGwire
neared
,
fans
went
wild
As McGwire neared , fans went wild

Outer Arc

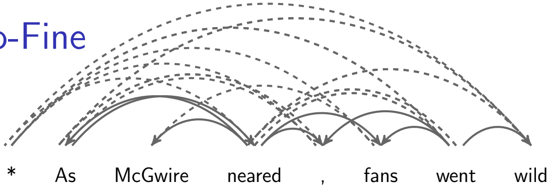


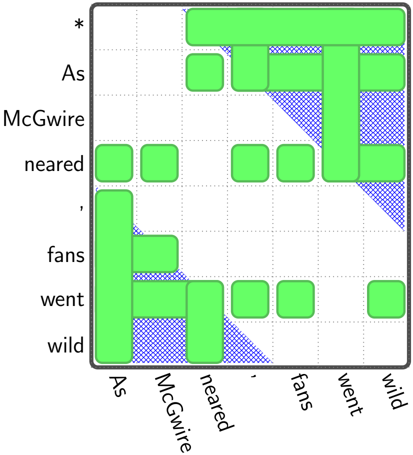
*
As
McGwire
neared
,
fans
went
wild
*
As
McGwire
neared
,
fans
went
wild
As
McGwire
neared
,
fans
went
wild

Outer Arc


* As McGwire neared , fans went wild
*
As
McGwire
neared
,
fans
went
wild
As McGwire neared , fans went wild

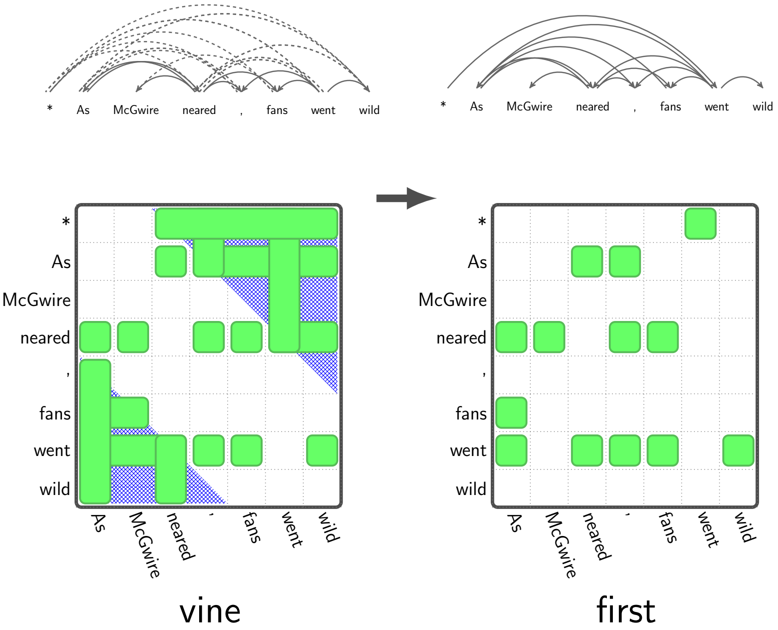
Coarse-to-Fine



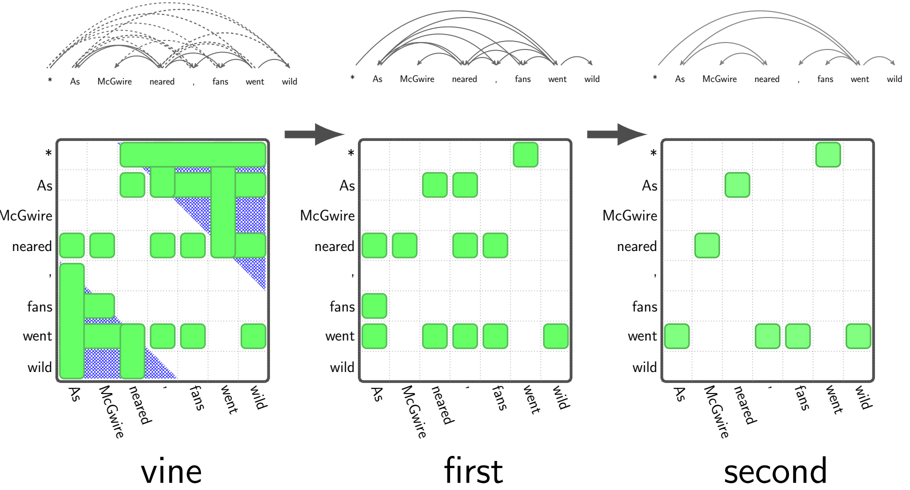


vine

Coarse-to-Fine



Coarse-to-Fine



dynamic programs for parsing

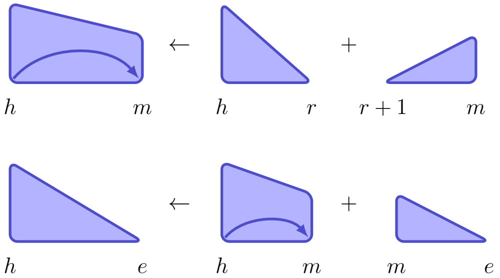
Inference Questions

questions:

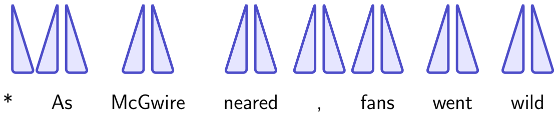
- How do we reduce inference time to $O(n)$?
- How do we decide which arcs to prune?

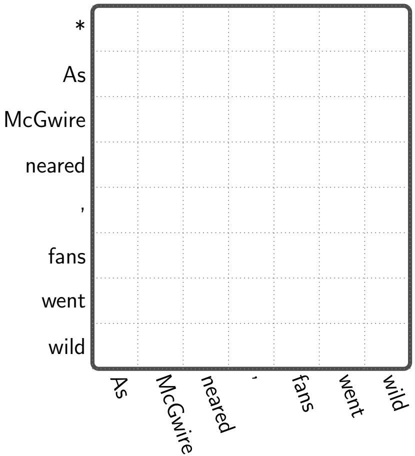
Vine Parsing (Eisner and Smith, 2005)

Eisner First-Order Rules

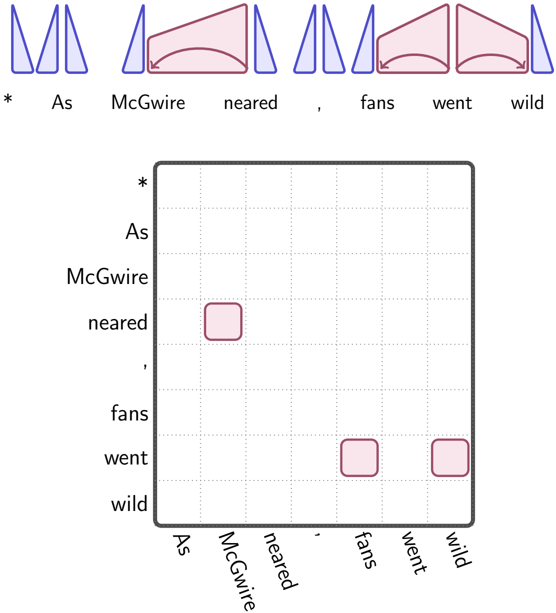


First-Order Parsing

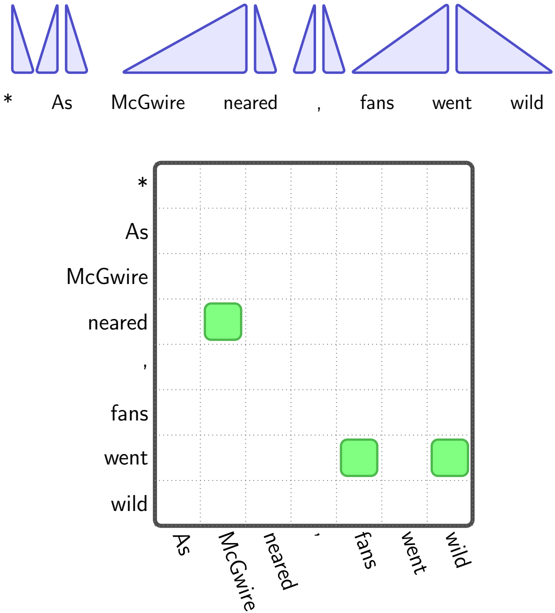




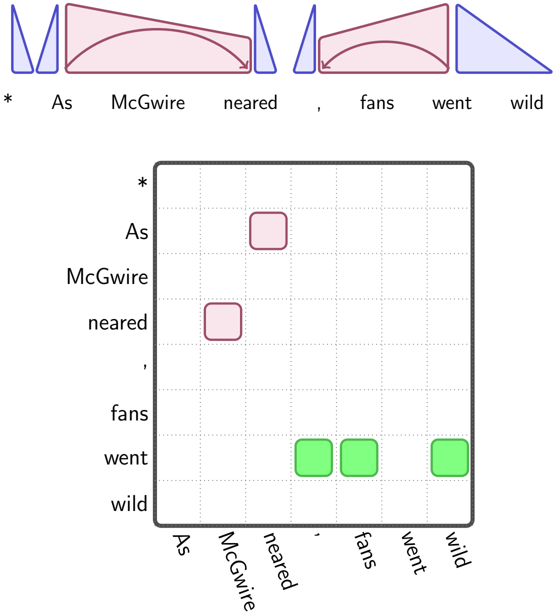
First-Order Parsing



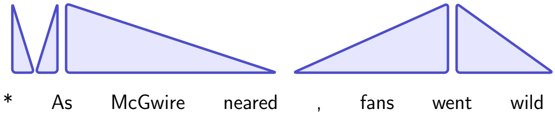
First-Order Parsing

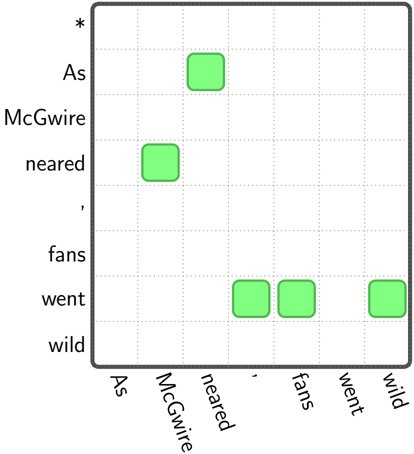


First-Order Parsing

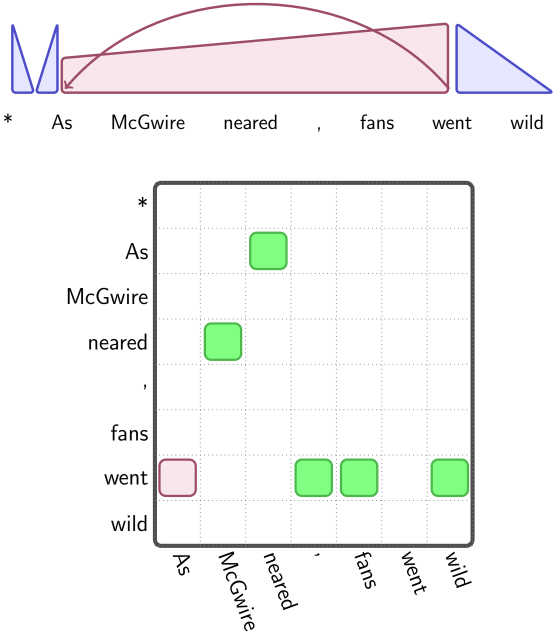


First-Order Parsing

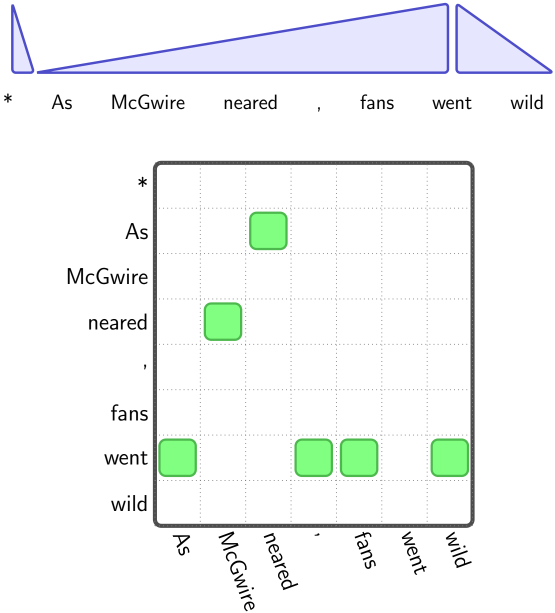




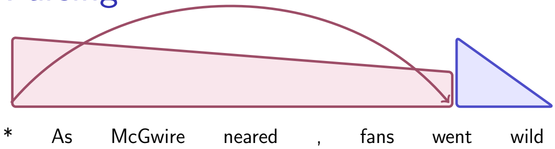
First-Order Parsing

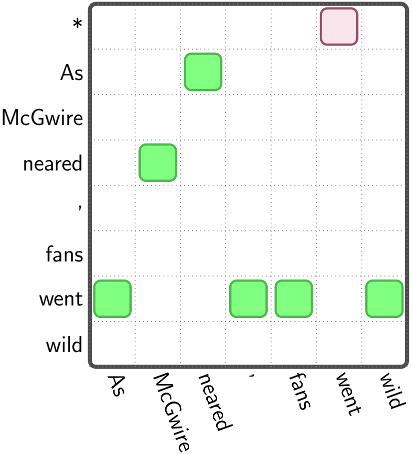


First-Order Parsing

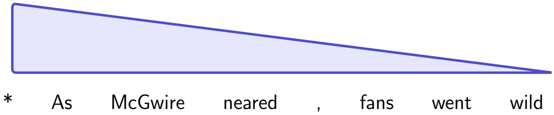


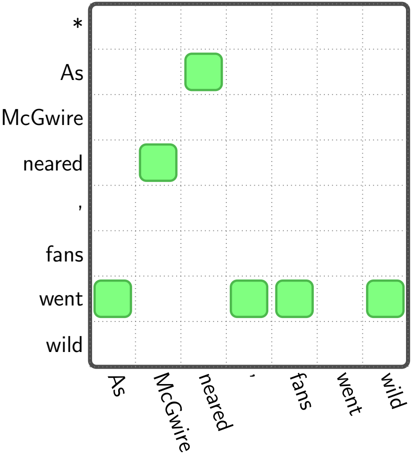
First-Order Parsing



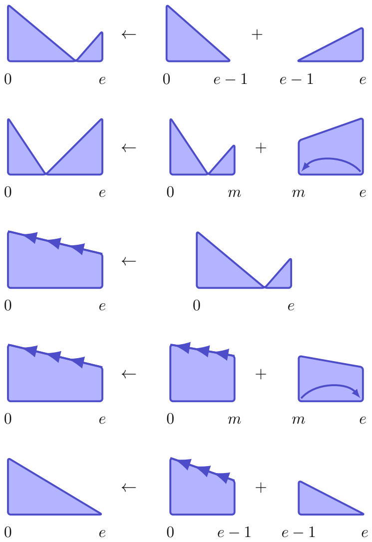


First-Order Parsing

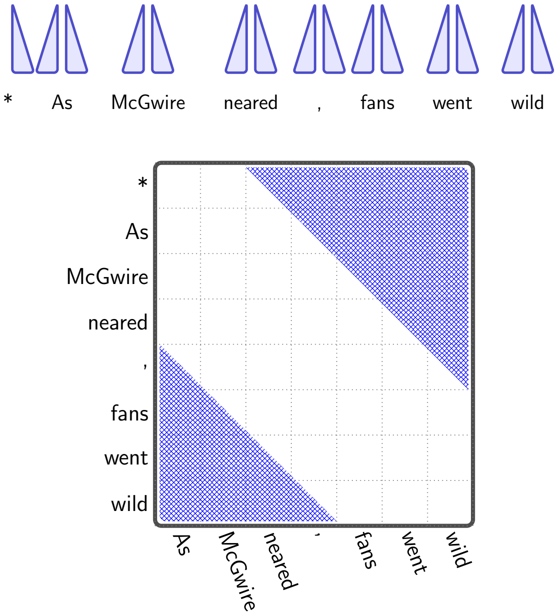




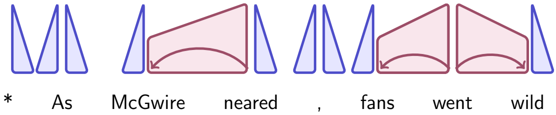
Vine Parsing Rules

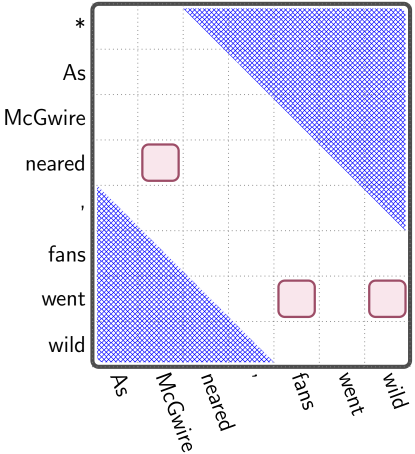


Vine Parsing

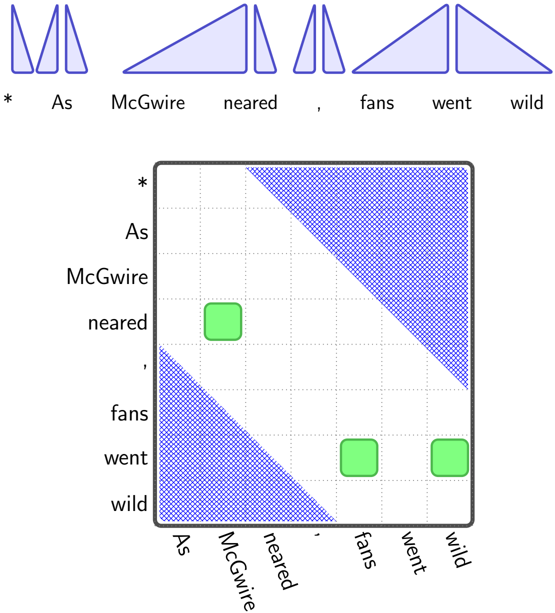


Vine Parsing

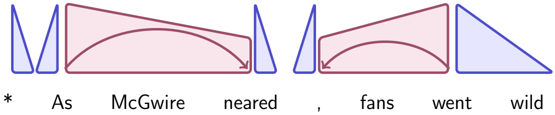


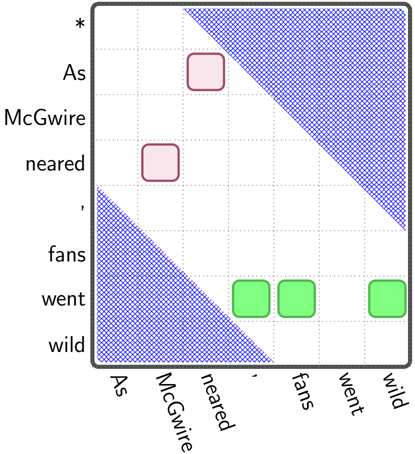


Vine Parsing

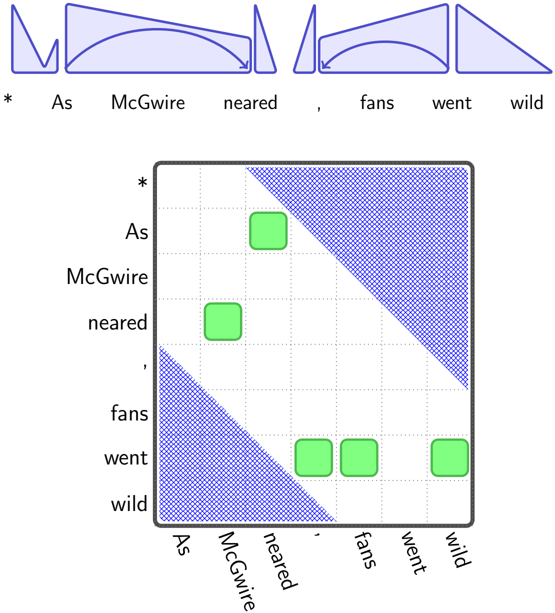


Vine Parsing

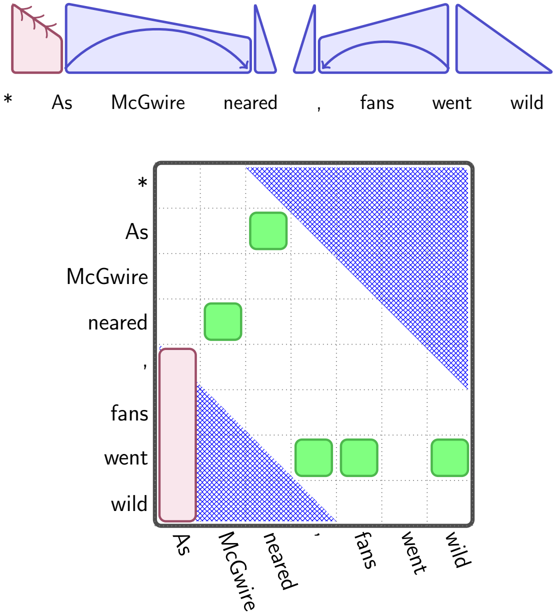




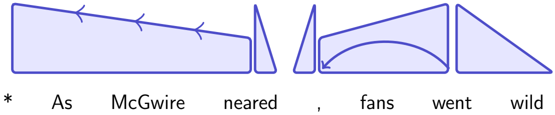
Vine Parsing

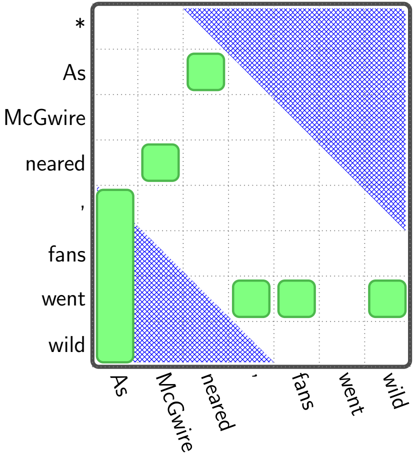


Vine Parsing

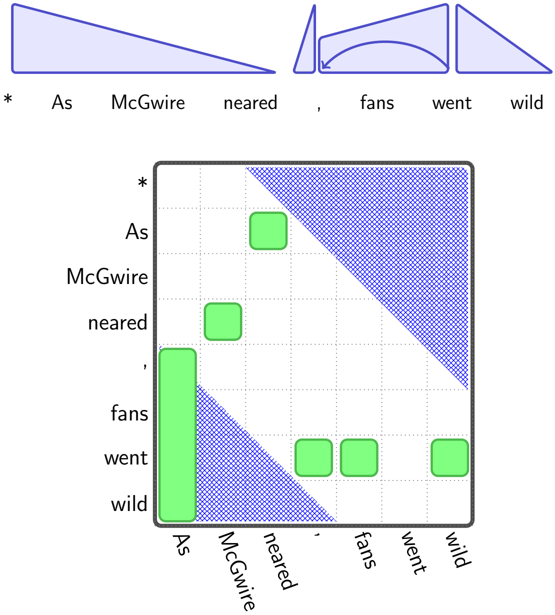


Vine Parsing

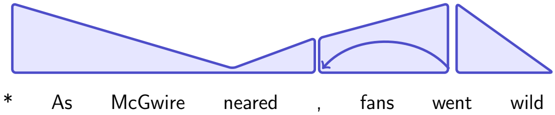


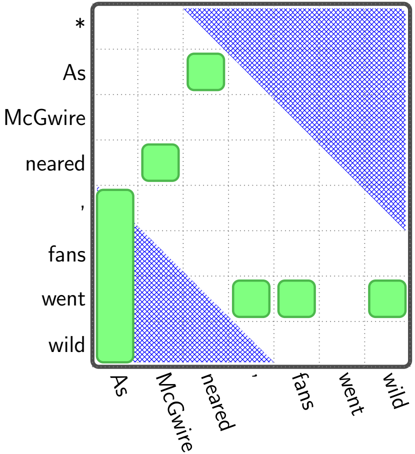


Vine Parsing

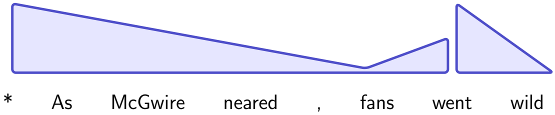


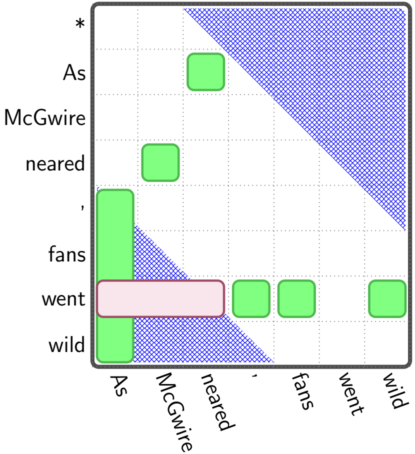
Vine Parsing



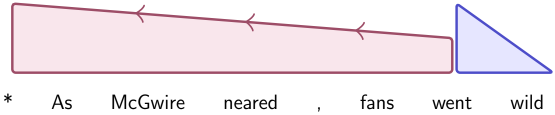


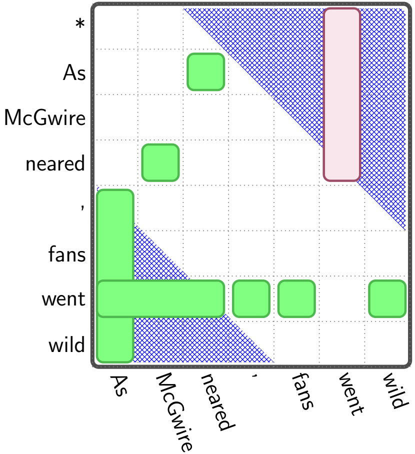
Vine Parsing



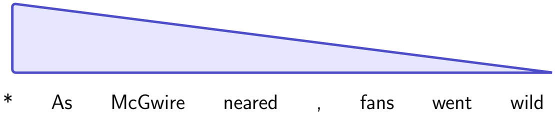


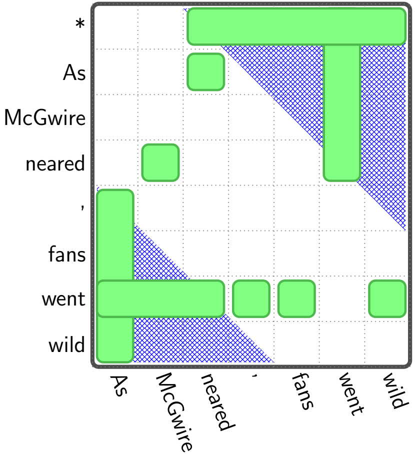
Vine Parsing





Vine Parsing





Arc Pruning

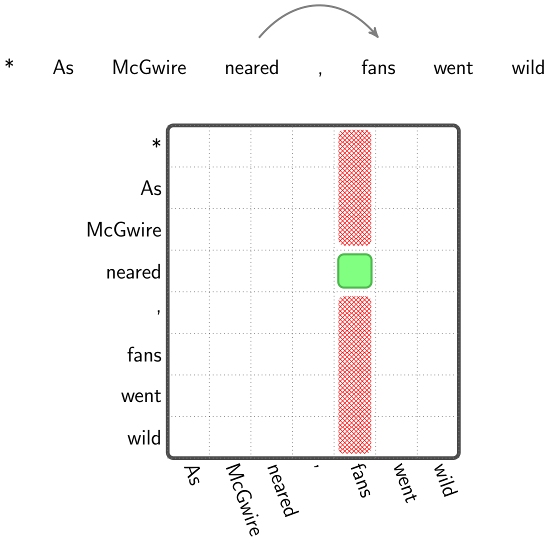
- Prune arcs based on max-marginals.

$$\text{maxmarginal}(\text{a}) = \max_{y: a \in y} (y \cdot w)$$

- Can compute using inside-outside algorithm.
- Generic algorithm using hypergraph parsing.

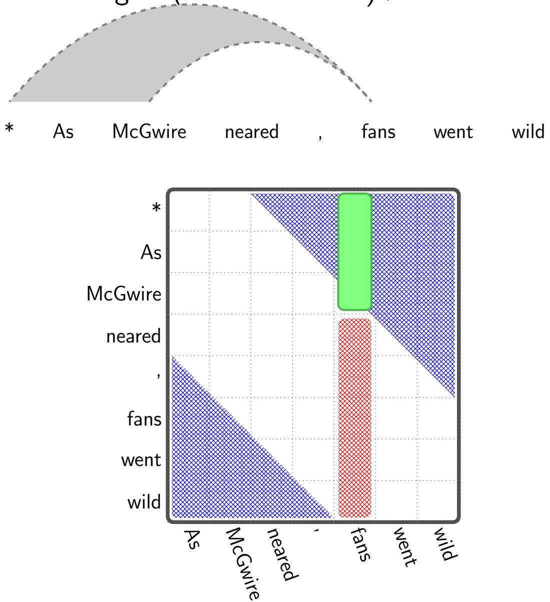
Max-Marginals for First-Order Arcs

$$\text{maxmarginal}(\texttt{neared} \rightarrow \texttt{fans}) > \text{threshold ?}$$



Max-Marginals for Outer Arcs

$$\text{maxmarginal}(\texttt{LEFT} \rightarrow \texttt{fans}) > \text{threshold ?}$$



pruning and training

Max-Marginal Pruning

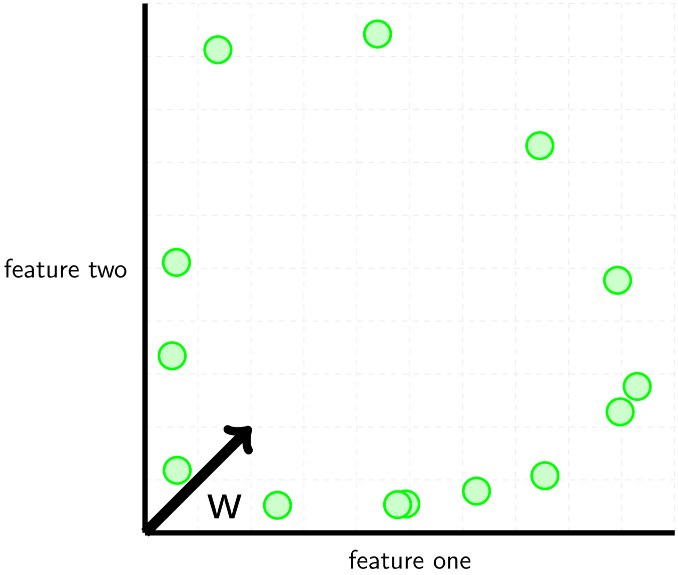
goal: Define a threshold on max-marginal score.

- Validation parameter α trades off between speed and accuracy.

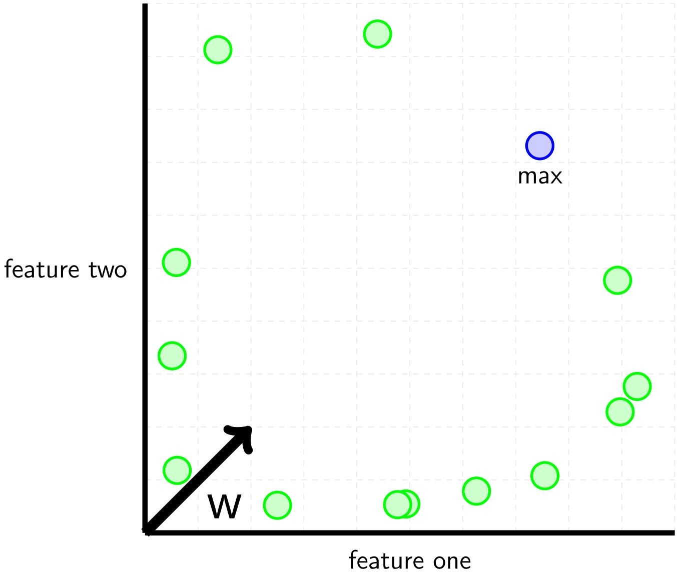
$$t_\alpha(w) \quad = \quad \alpha \max_y (y \cdot w) + (1-\alpha)\frac{1}{|A|}\sum_{a \in A} \text{maxmarginal}(a, w)$$

- Highest scoring parse upper bounds any max-marginal.
- Assume average of max-marginals is lower than gold.

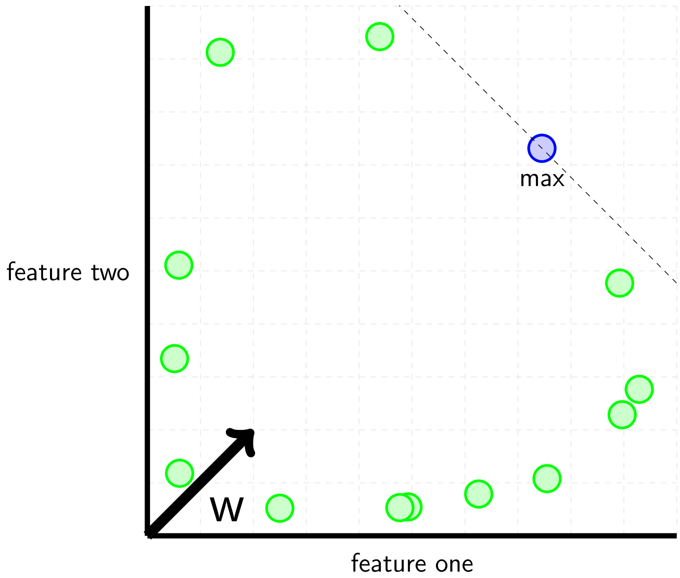
Pruning Threshold



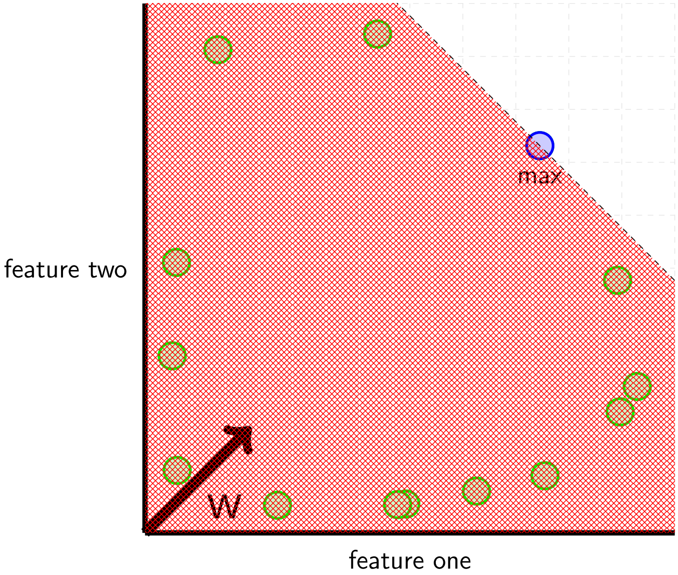
Pruning Threshold



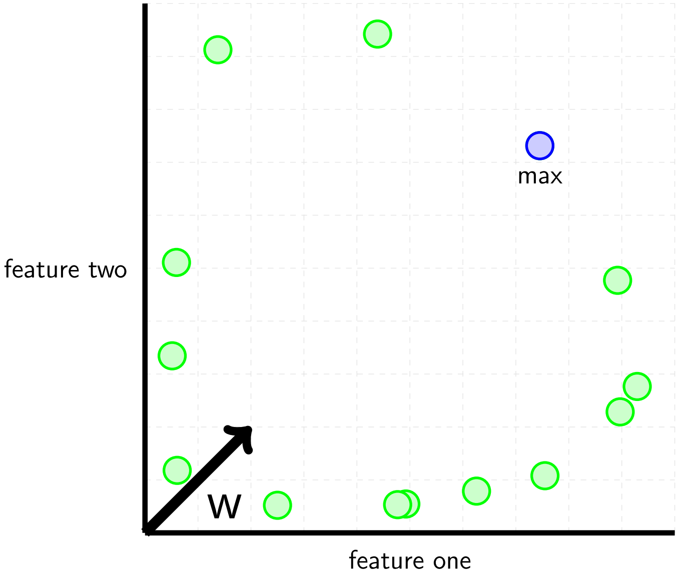
Pruning Threshold



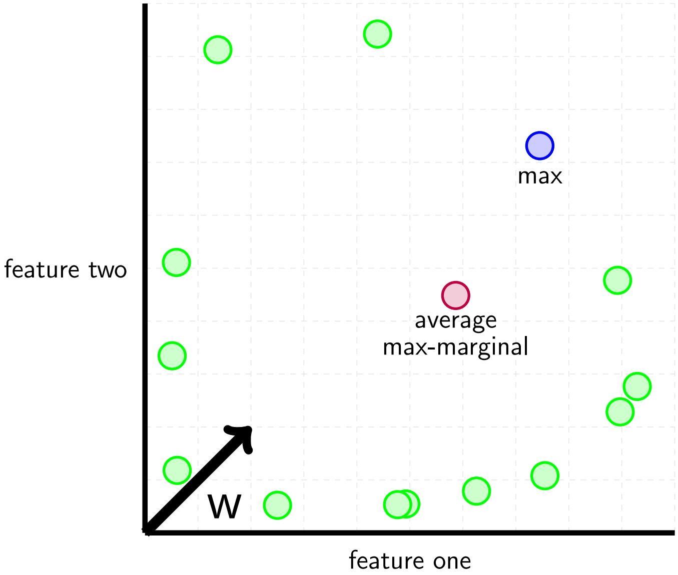
Pruning Threshold



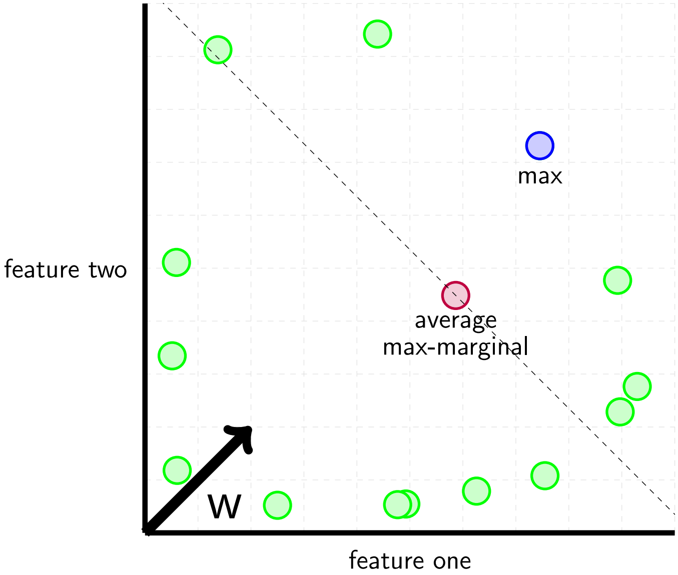
Pruning Threshold



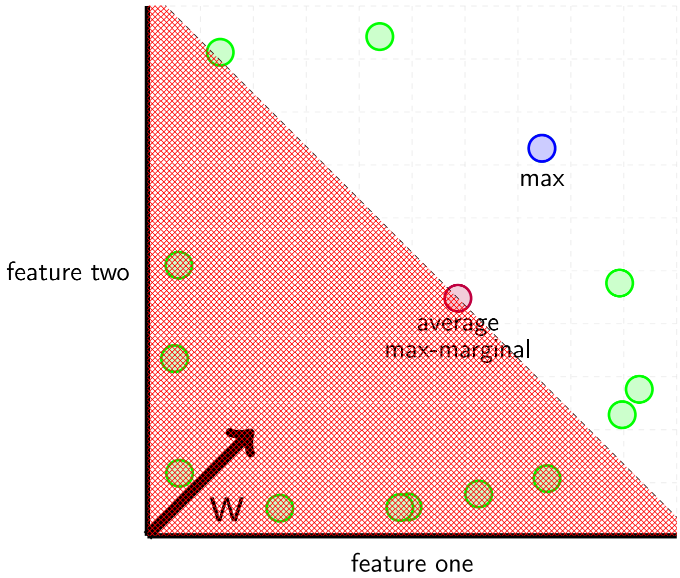
Pruning Threshold



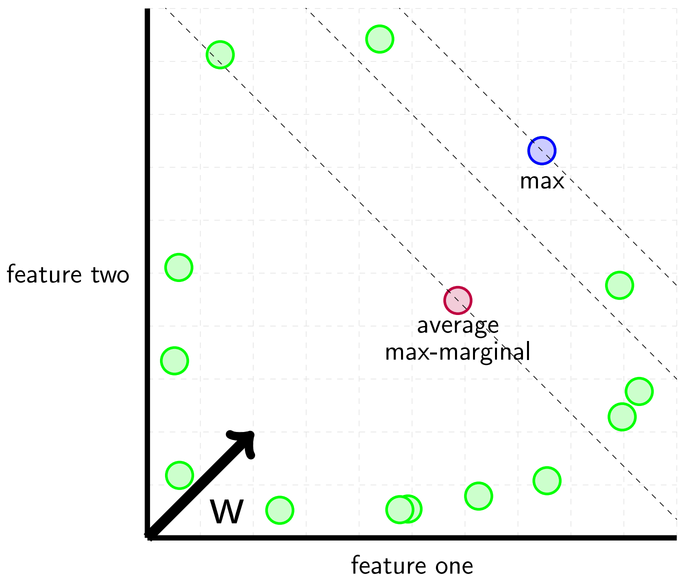
Pruning Threshold



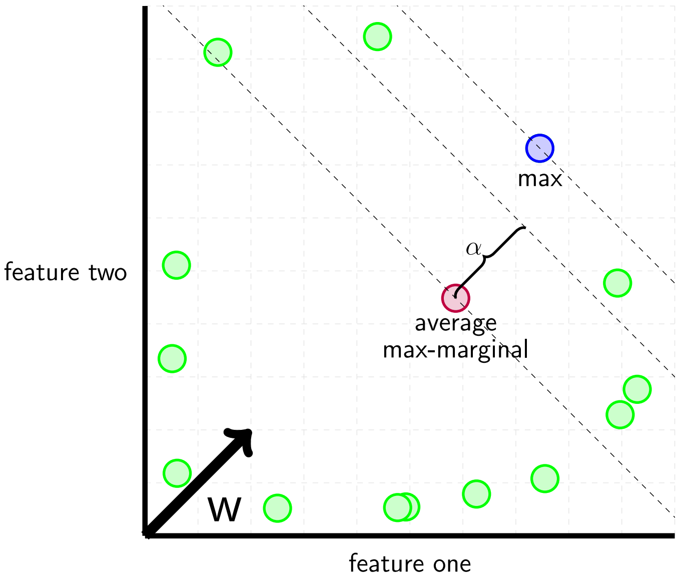
Pruning Threshold



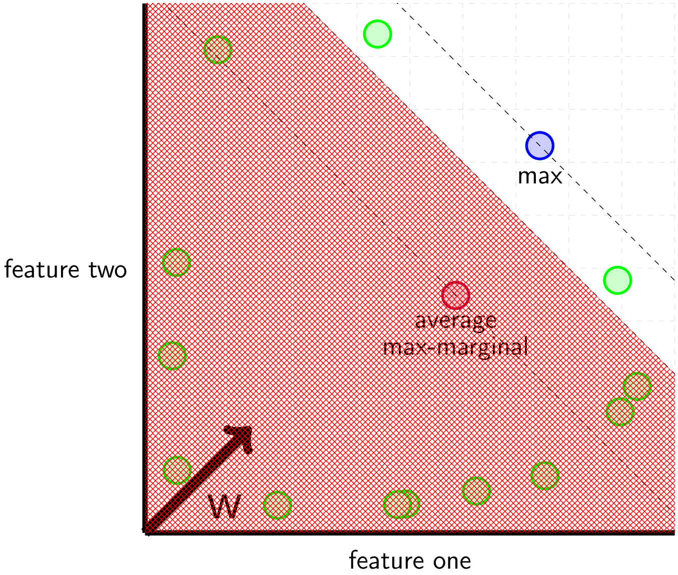
Pruning Threshold



Pruning Threshold



Pruning Threshold



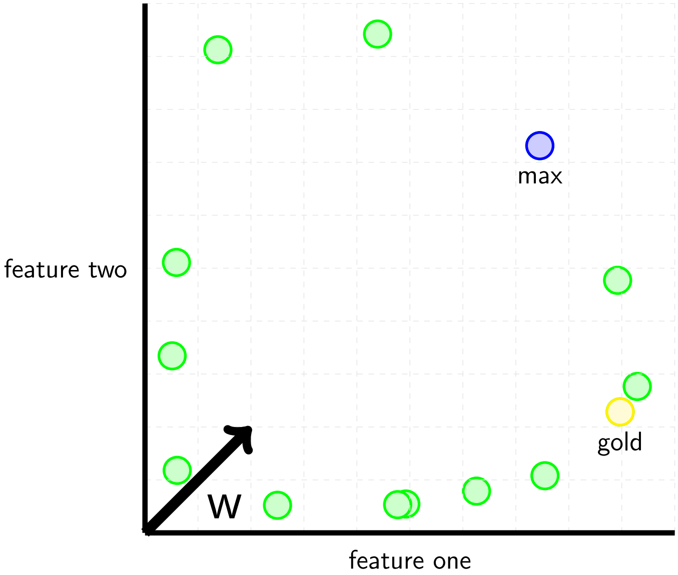
Structured Cascade Training (Weiss and Taskar, 2011)

- Train a linear model with a loss function for pruning.
- Regularized risk minimization with loss based on threshold

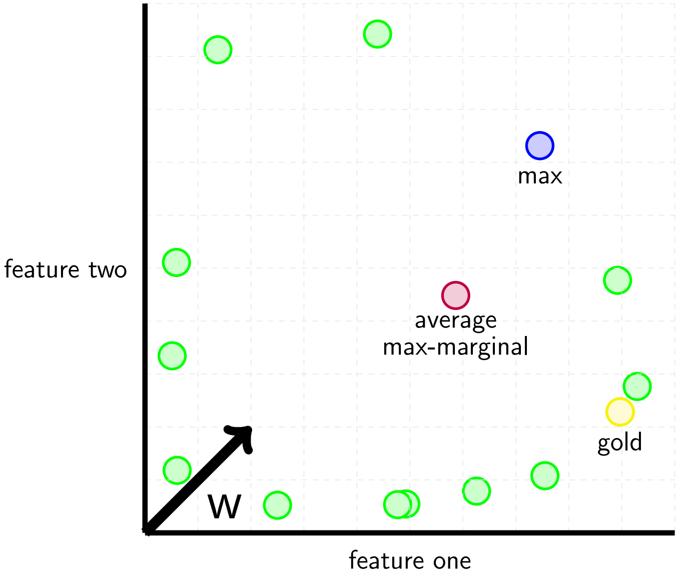
$$\min_{w} \lambda \|w\|^2 + \frac{1}{P} \sum_{p=1}^{P} [1 - y^{(p)} \cdot w + t_{\alpha}^{(p)}(w)]_+$$

- Can use a simple variant of perceptron/pegasos to train.

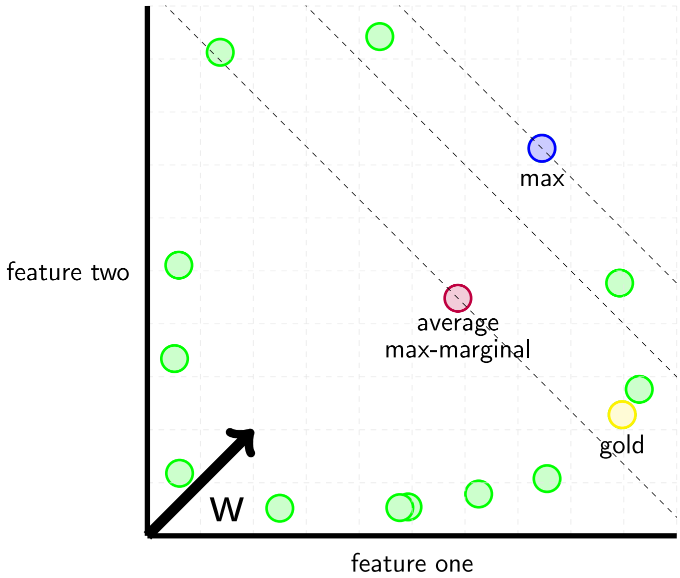
Structured Cascade Training



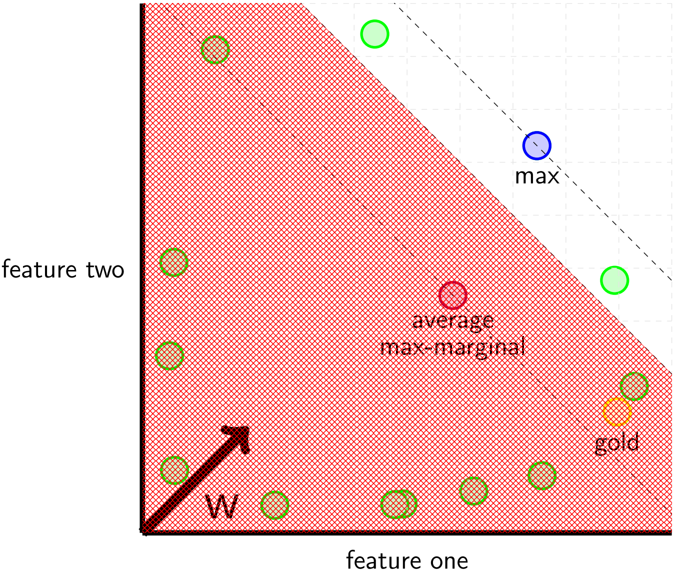
Structured Cascade Training



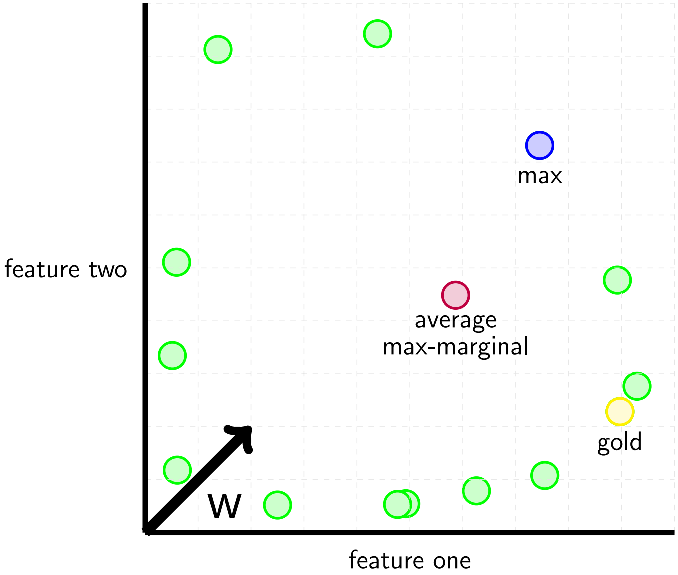
Structured Cascade Training



Structured Cascade Training

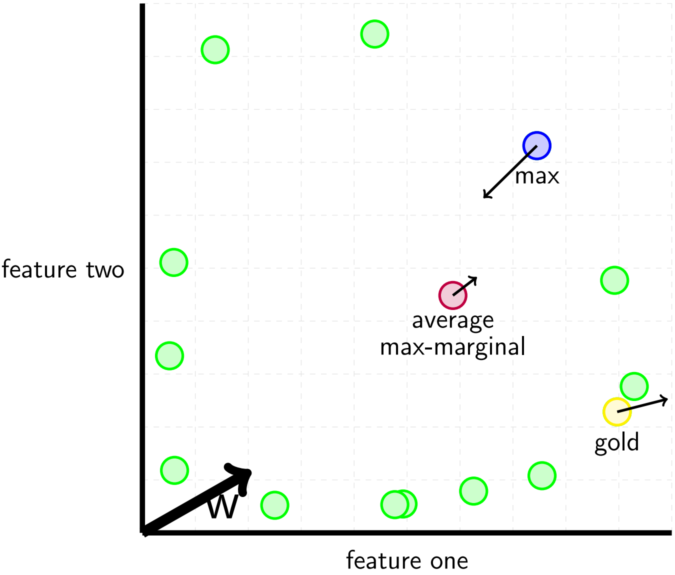


Structured Cascade Training

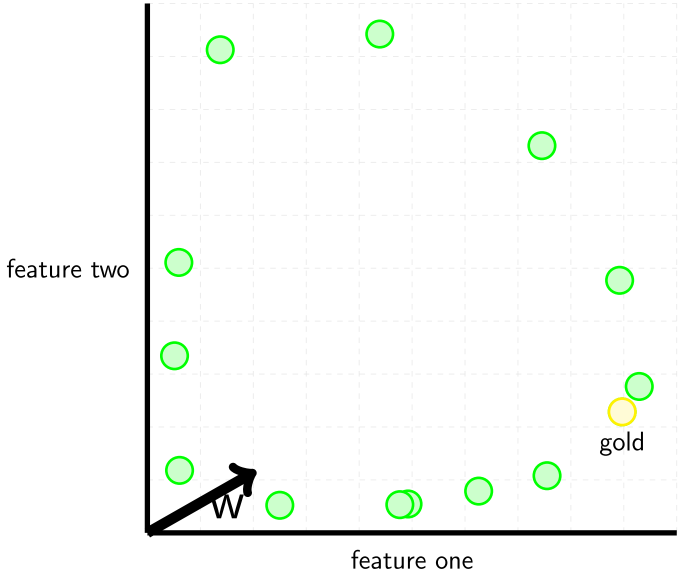


Structured Cascade Training

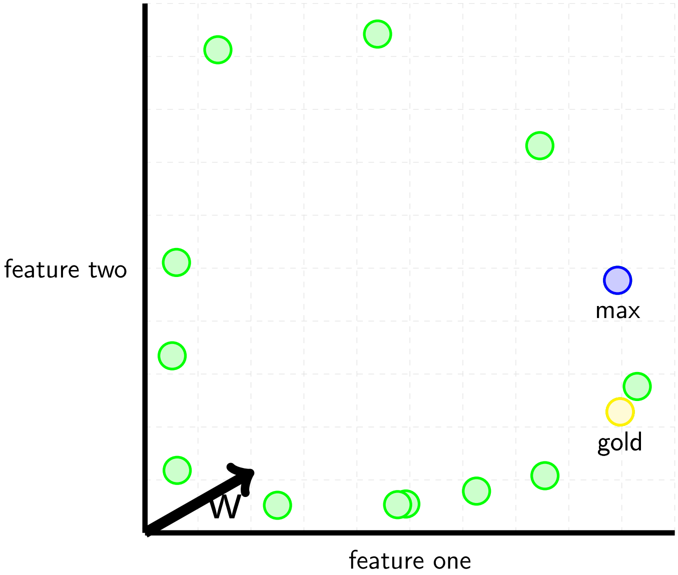
Structured Cascade Training



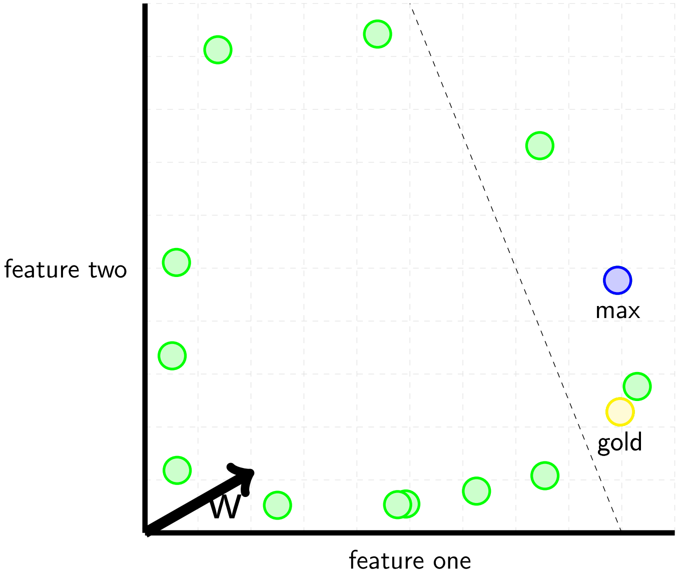
Structured Cascade Training



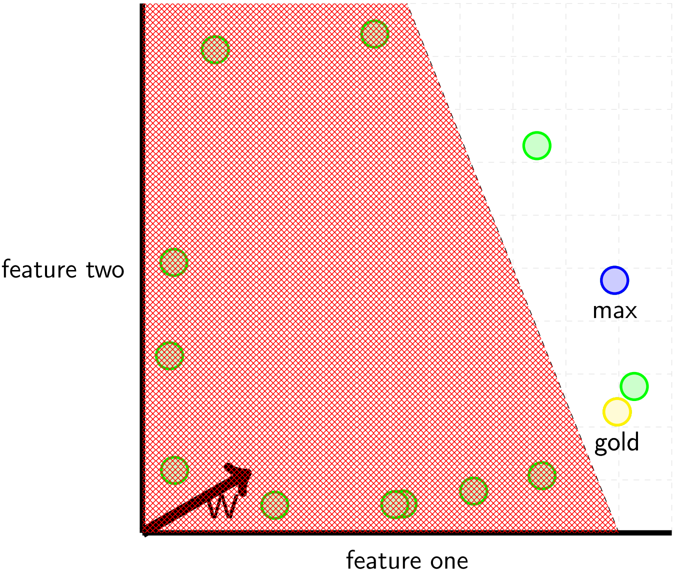
Structured Cascade Training



Structured Cascade Training



Structured Cascade Training



experiments

Implementation

Inference

- Experiments use a highly-optimized `C++` implementation.
- Baseline first-order parser processes 2000 tokens/sec.
- Hypergraph parsing framework with shared inference.

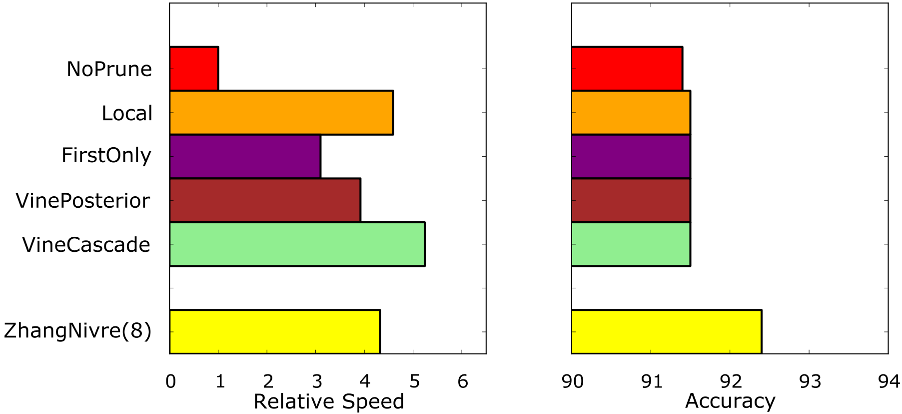
Model

- Final models trained with hamming-loss MIRA.
- Full collection of dependency parsing features (Koo, 2010).
- First-, second-, and third-order models match state-of-the-art.

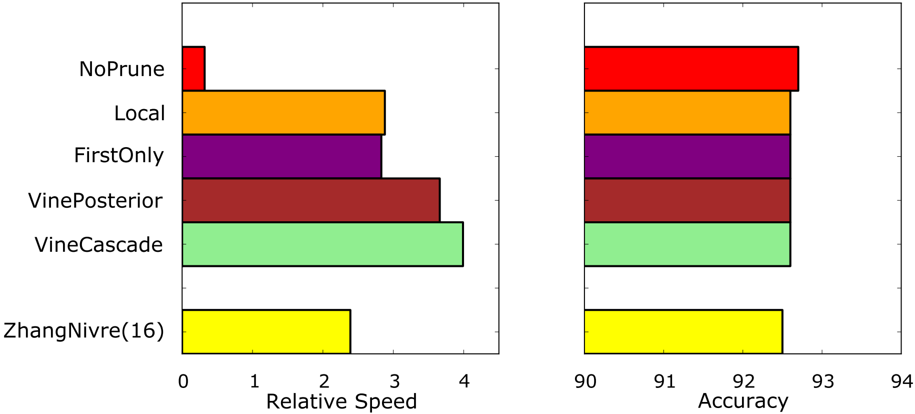
Baselines

NoPrune	exhaustive parsing model with no pruning
LocalShort	unstructured classifier over $O(n)$ short arcs (Bergsma and Cherry, 2010)
Local	unstructured classifier over $O(n^2)$ arcs (Bergsma and Cherry, 2010)
FirstOnly	structured first-order model in cascade (Koo, 2010)
VinePosterior	posterior pruning cascade trained with L-BFGS
ZhangNivre	reimplementation of state-of-the-art, k-best, transition-based parser (Zhang and Nivre, 2011).

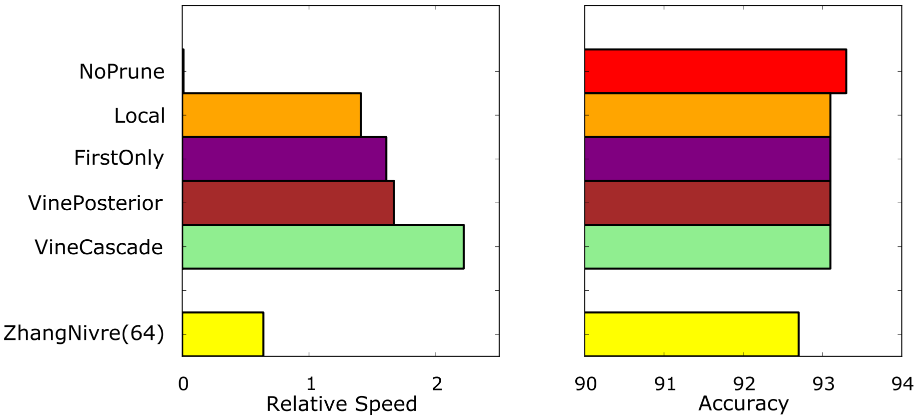
Speed/Accuracy Experiments: First-Order Parsing



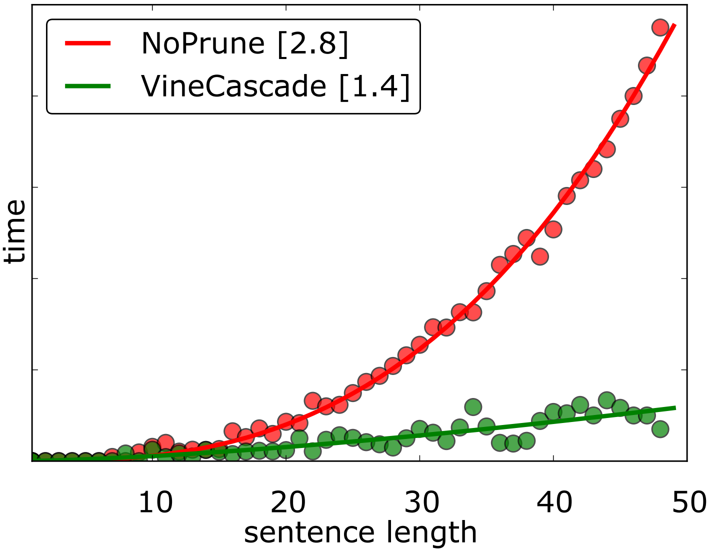
Speed/Accuracy Experiments: Second-Order Parsing



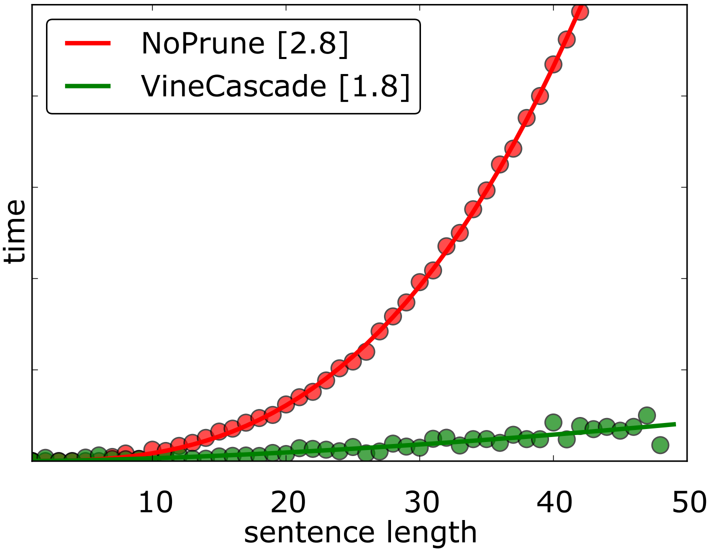
Speed/Accuracy Experiments: Third-Order Parsing



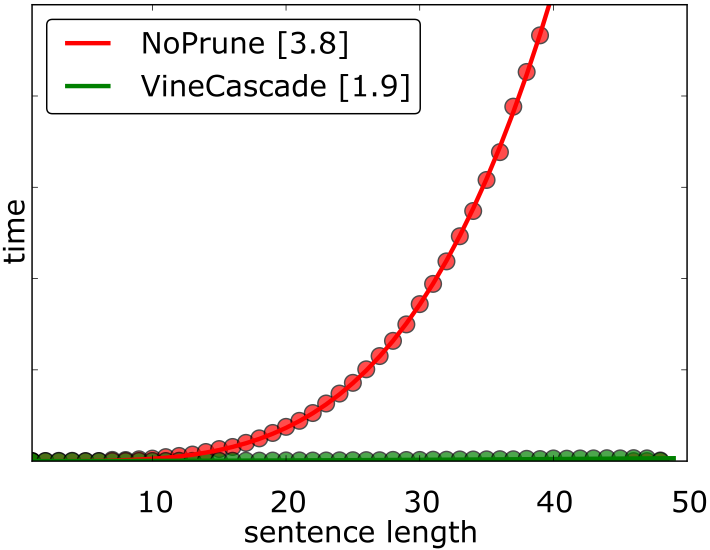
Empirical Complexity: First-Order Parsing



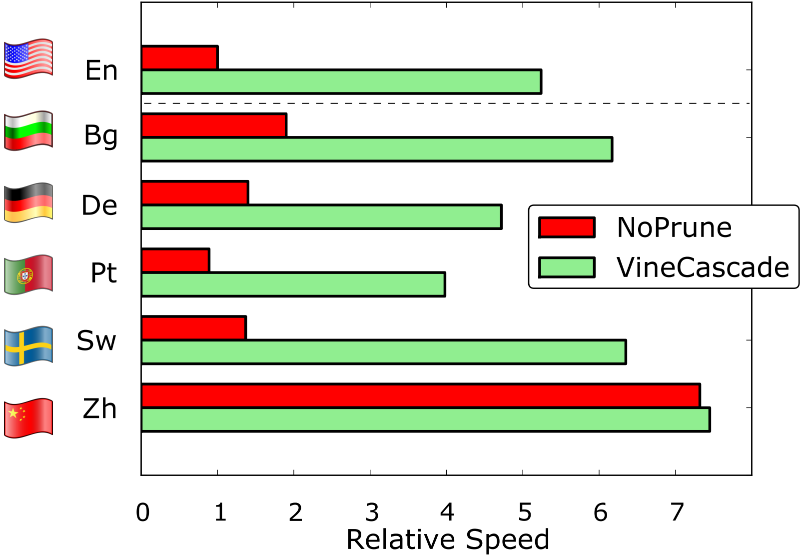
Empirical Complexity: Second-Order Parsing



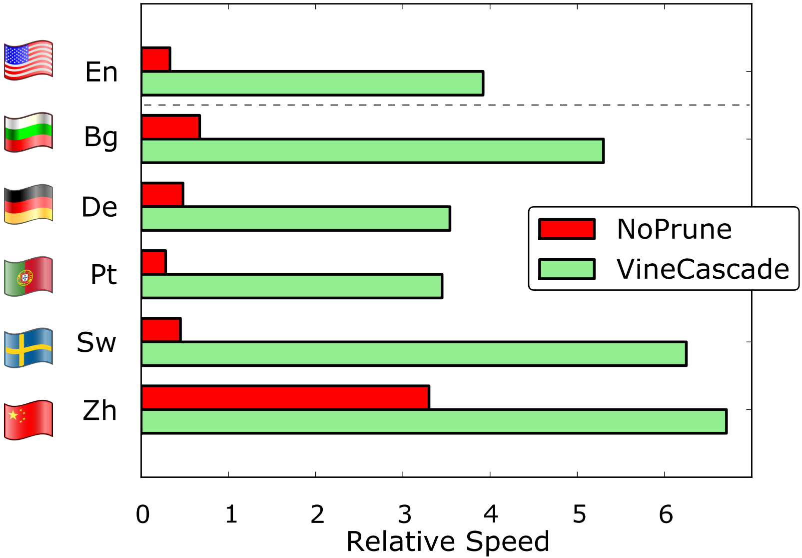
Empirical Complexity: Third-Order Parsing



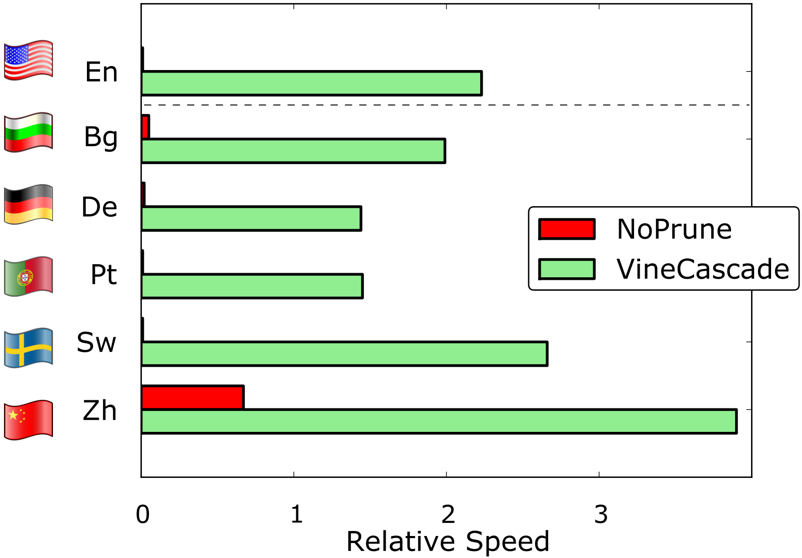
Multilingual Experiments: First-Order Parsing



Multilingual Experiments: Second-Order Parsing



Multilingual Experiments: Third-Order Parsing



Special thanks to:

Ryan McDonald, Hao Zhang, Michael Ringgaard, Terry Koo, Keith Hall, Kuzman Ganchev, Yoav Goldberg, Andre Martins, and the rest of the Google NLP team