Learning to Parse Using a Tiny Corpus

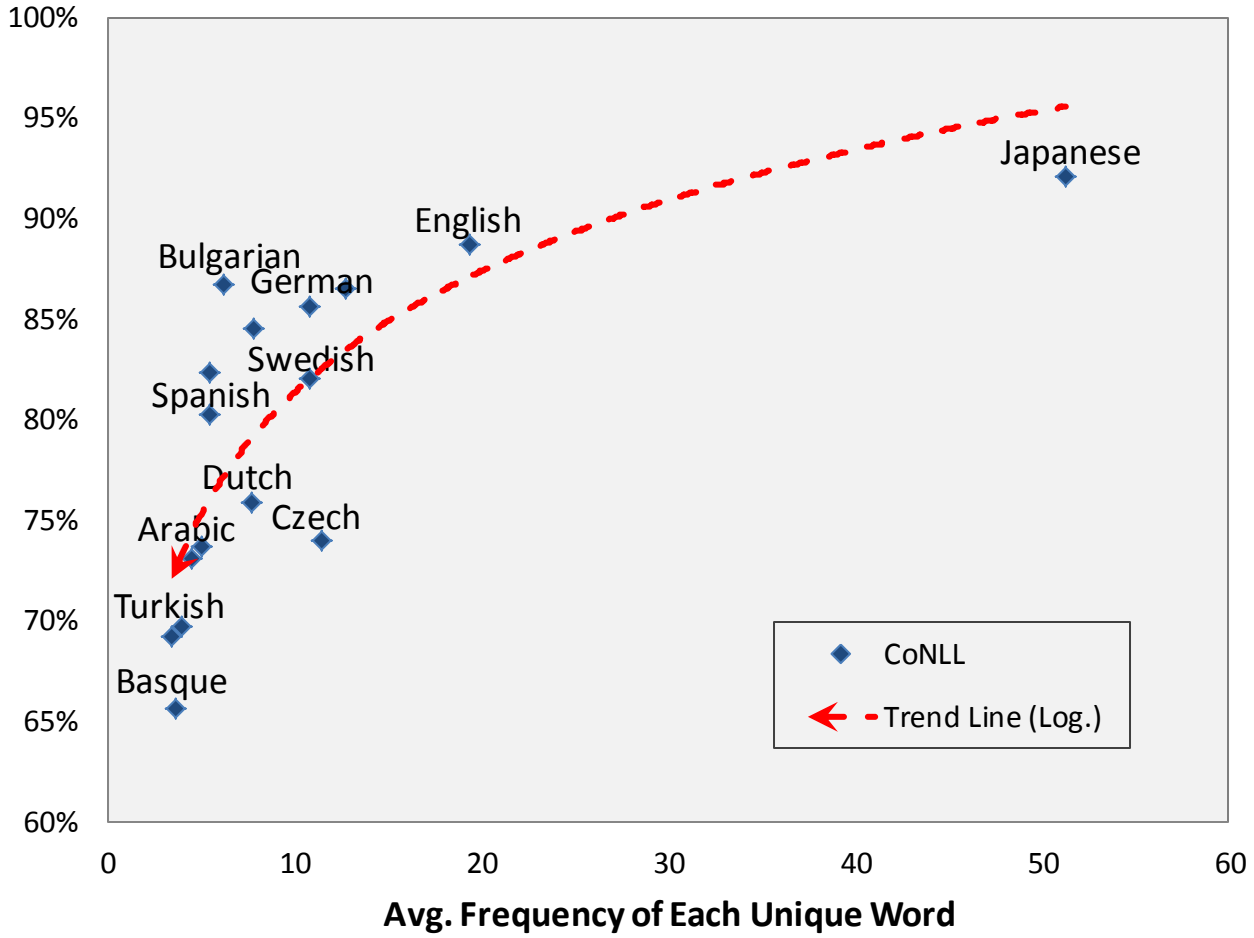
Tao Lei, Yu Xin

Regina Barzilay, Tommi Jaakkola

CSAIL, MIT

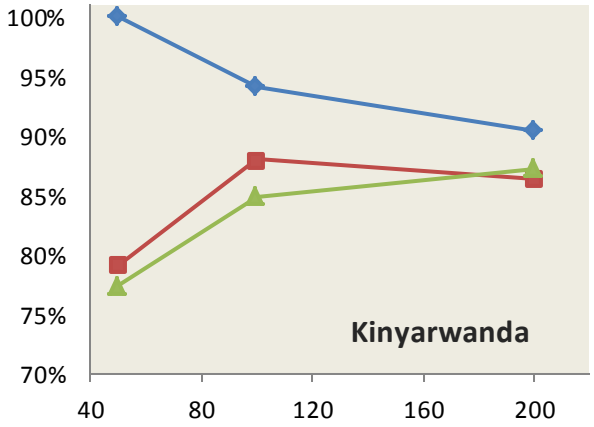
Sparsity Problem

- Data sparsity makes parsing harder
  - *due to less frequent/unseen words* and *dependency arcs* in data

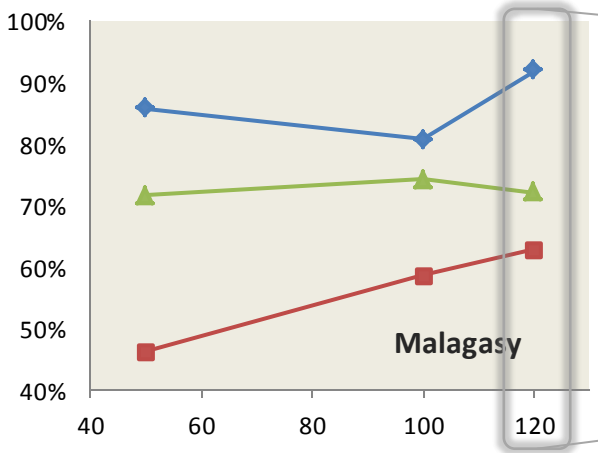


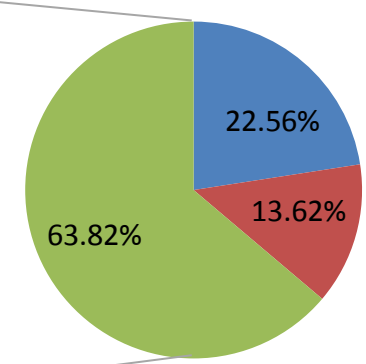
Sparsity Problem

- Prediction is worse when the arc is not seen in the training data



Seen Arc: See dependency arc in the training data
Seen Words: See the words in training but arc unseen
Unseen: At least one word not in training

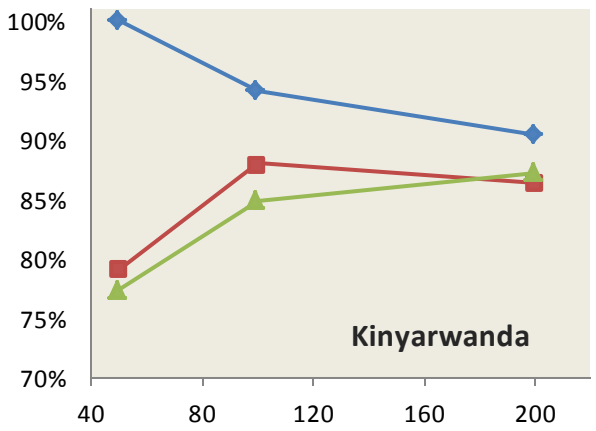


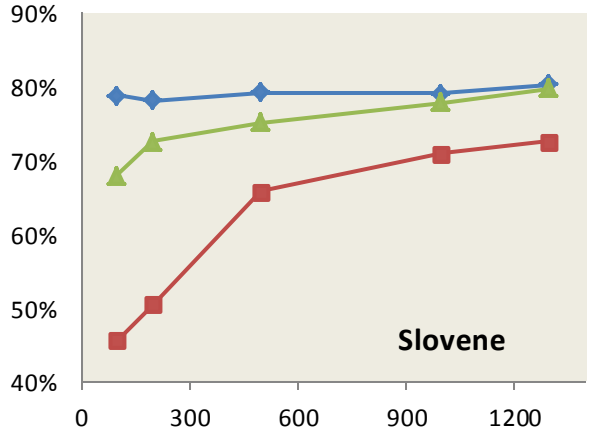


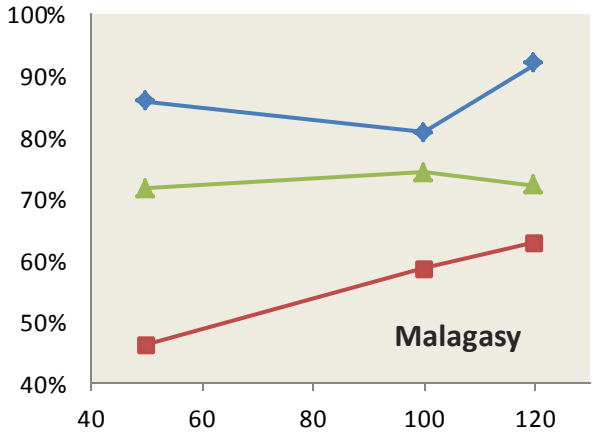
a large portion of dependency arcs in test is unseen

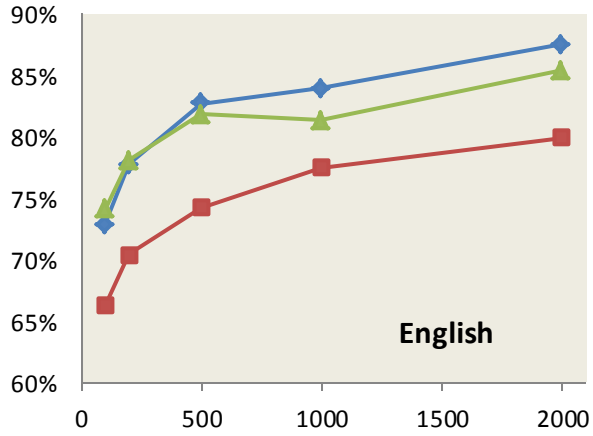
Sparsity Problem

- Prediction is worse when the arc is not seen in the training data



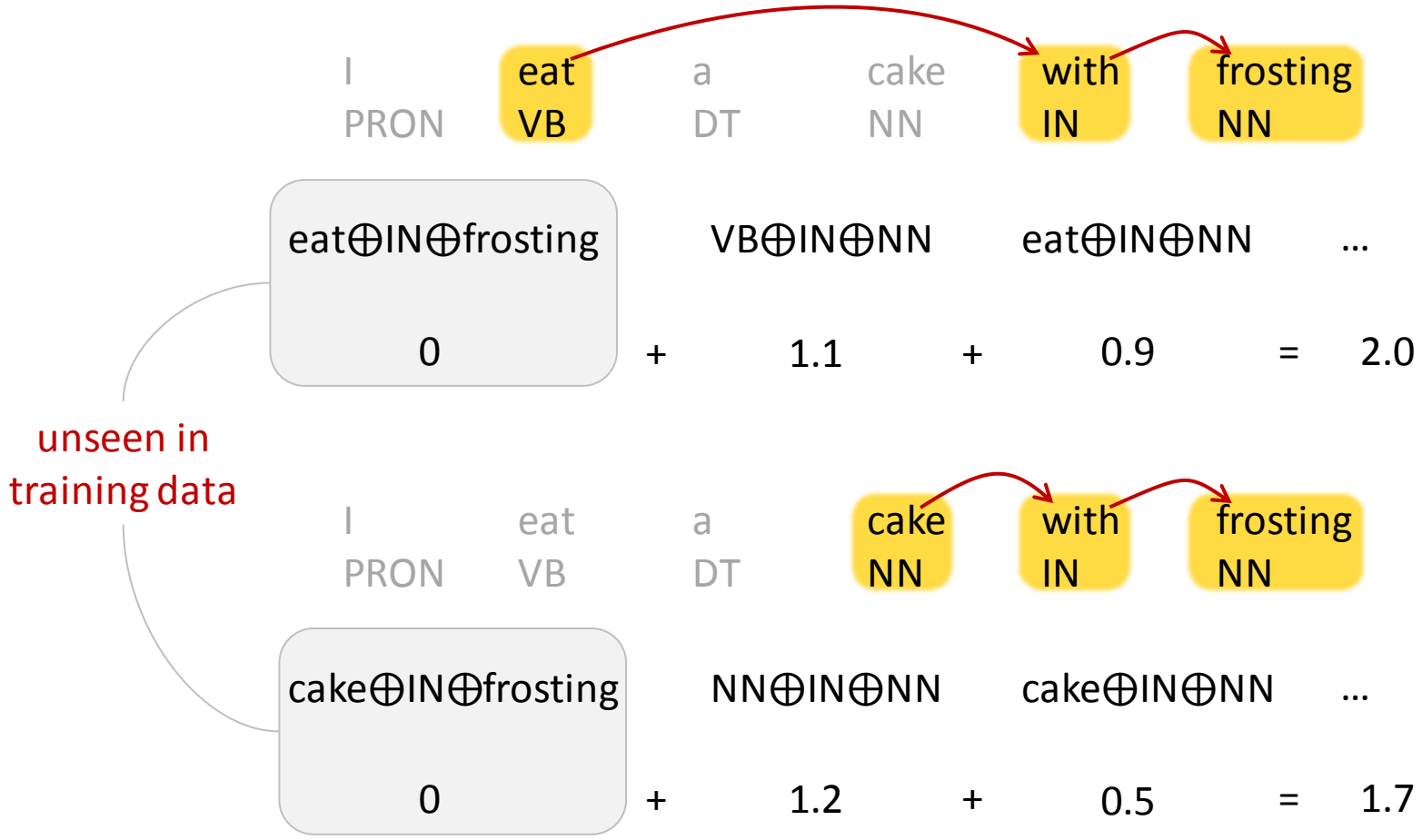






Sparsity Problem

- Feature weights are zeros when the features are not seen



Opportunity and Challenge

To deal with sparsity problem, we will

- Make the model flexible to add various rich features
  - For example, words, coarse-to-fine POS tags and word embeddings
  - Adjust complexity based on how much training data it has

- Model interactions between feature weights
  - E.g. propagating weights from seen features to unseen features

Motivating Example: Matrix Completion

- Learn a matrix (or high-order tensor) that has a lot of unseen entries
  - Example: image

Input
image with missing values

Output
re-constructed image

  - Example: Netflix problem

    Users give only a few movie ratings. Predict unseen ratings

Motivating Example

- In our case: learn a parameter matrix (or tensor) with unseen weights

Dependency Arc

$$\{\ \text{eat} \oplus \text{apple}, \text{eat} \oplus \text{NN}, \text{VB} \oplus \text{apple}, \text{VB} \oplus \text{NN}\ \}$$

Feature Strings

	eat	apple	banana	VB	NN
eat		1			1
apple					
banana					
VB		1			1
NN					

Feature Matrix

$\otimes$

…	**2**	…	…	**4**
…	0	0	…	…
…	0	0	…	
…	**1**	0.9	…	**5**
…	0.1	0.1	…	…

Parameter Matrix

$= 12$

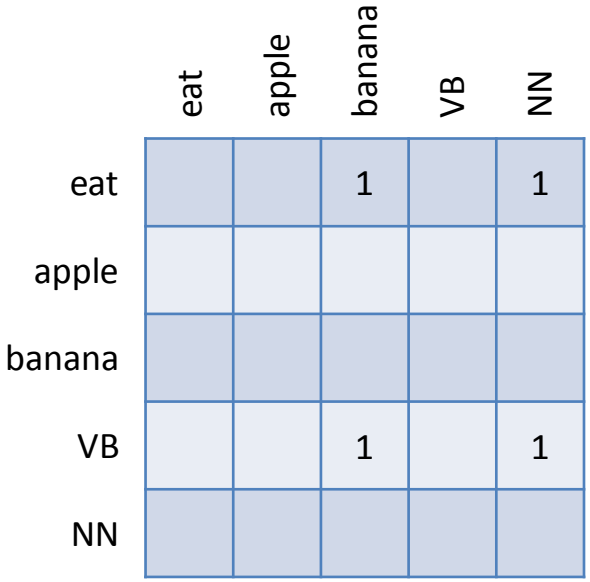
Motivating Example

- In our case: learn a parameter matrix (or tensor) with unseen weights

Dependency Arc

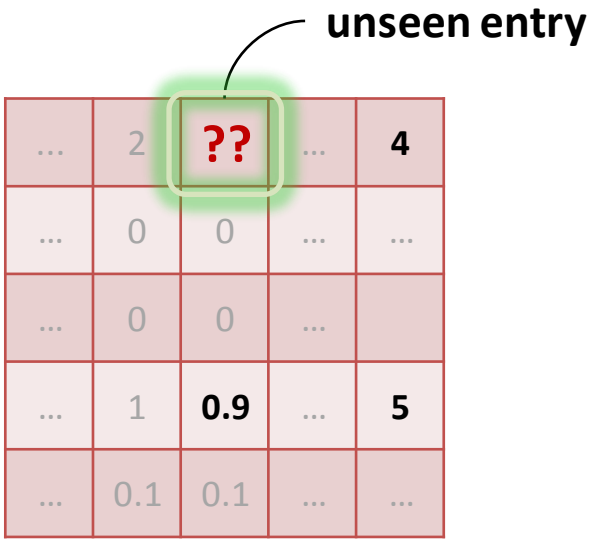
$$\{ \text{eat} \oplus \text{banana}, \text{eat} \oplus \text{NN}, \text{VB} \oplus \text{banana}, \text{VB} \oplus \text{NN} \}$$

Feature Strings



Feature Matrix

$\otimes$



Parameter Matrix

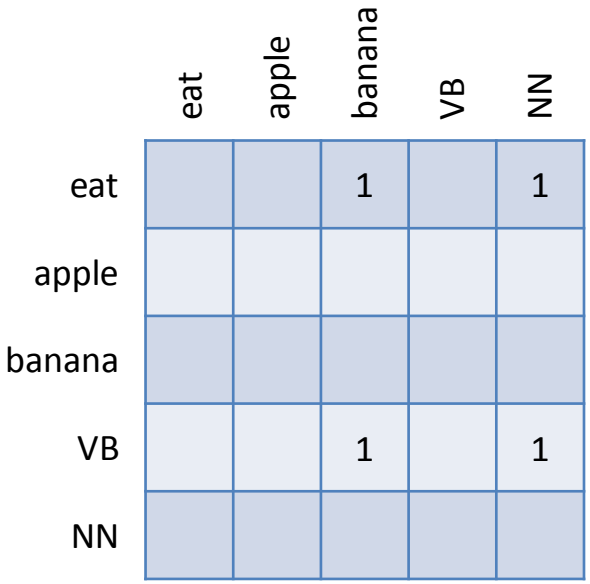
Motivating Example

- In our case: learn a parameter matrix (or tensor) with unseen weights

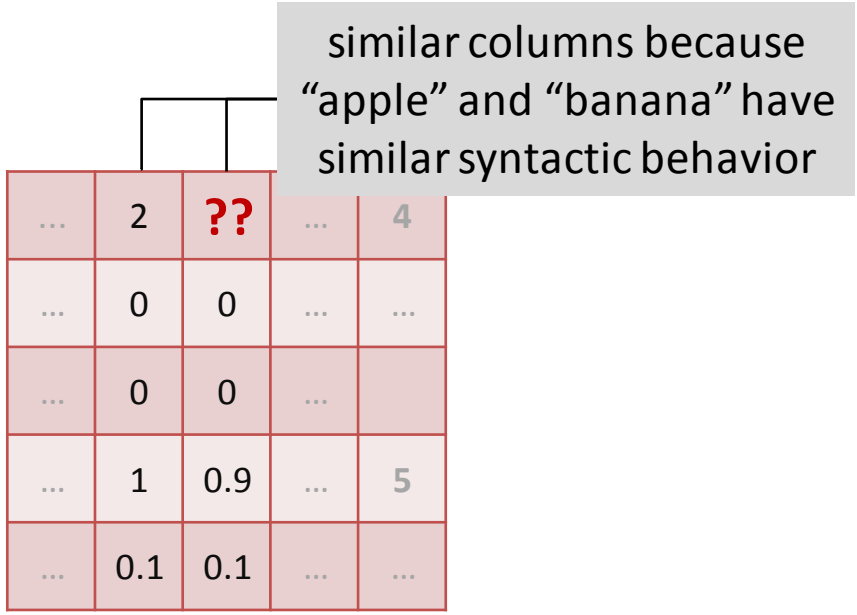
Dependency Arc

{ eat⊕banana, eat⊕NN, VB⊕banana, VB⊕NN }

Feature Strings



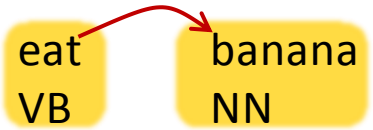
Feature Matrix



Parameter Matrix

Motivating Example

- In our case: learn a parameter matrix (or tensor) with unseen weights



Dependency Arc

$\{ \text{eat} \oplus \text{banana}, \text{eat} \oplus \text{NN}, \text{VB} \oplus \text{banana}, \text{VB} \oplus \text{NN} \}$

Feature Strings

	eat	apple	banana	VB	NN
eat			1		1
apple					
banana					
VB			1		1
NN					

Feature Matrix

$\otimes$

…	2	**2**	…	**4**
…	0	0	…	…
…	0	0	…	
…	1	**0.9**	…	**5**
…	0.1	0.1	…	…

Parameter Matrix

$= 11.9$

Formulation

- Recall first-order decoding objective:

$$\begin{aligned} \widetilde{y_i} &= \underset{y_i \in T(x_i)}{\operatorname{argmax}} S(y_i) \\ &= \underset{y_i \in T(x_i)}{\operatorname{argmax}} \sum_{(h,c) \in y_i} \mathrm{s}(h,c) \end{aligned}$$

- Define score as matrix (tensor) inner product:

$$s(h,c) = \boldsymbol{\theta} \otimes \phi(h,c)$$

$$\mathrm{s}(h,c) = {w_h}^T \boldsymbol{A}\, w_c + \boldsymbol{\eta}^{\mathrm{T}} d(h,c)$$

Model Parameters: $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{\eta}\}$

Feature Matrix/Vector: $w_h w_c^{\mathrm{T}}$ $d(h,c)$

Formulation

- Minimize the loss of training data:

$$\min_{A,\eta} \mathcal{L}(D; A, \eta) = \frac{1}{N} \ell(x_i, \hat{y}_i)$$

$$\text{s.t.} \quad \|A\|_* + \lambda \|\eta\| \leq C$$

Force A to be low-rank using nuclear norm constraint

  - online learning algorithm available

    (Jaggi & Sulovsky, 2010) (Hazan, 2008)

Initialize $Z^{(1)} := v_0 v_0^T$ for arbitrary unit vector v.

For $k = 1$ to T do

  Compute $v_k := \text{ApproxEV}\left(-\nabla f(Z^{(k)}), \frac{C_f}{k^2}\right)$.

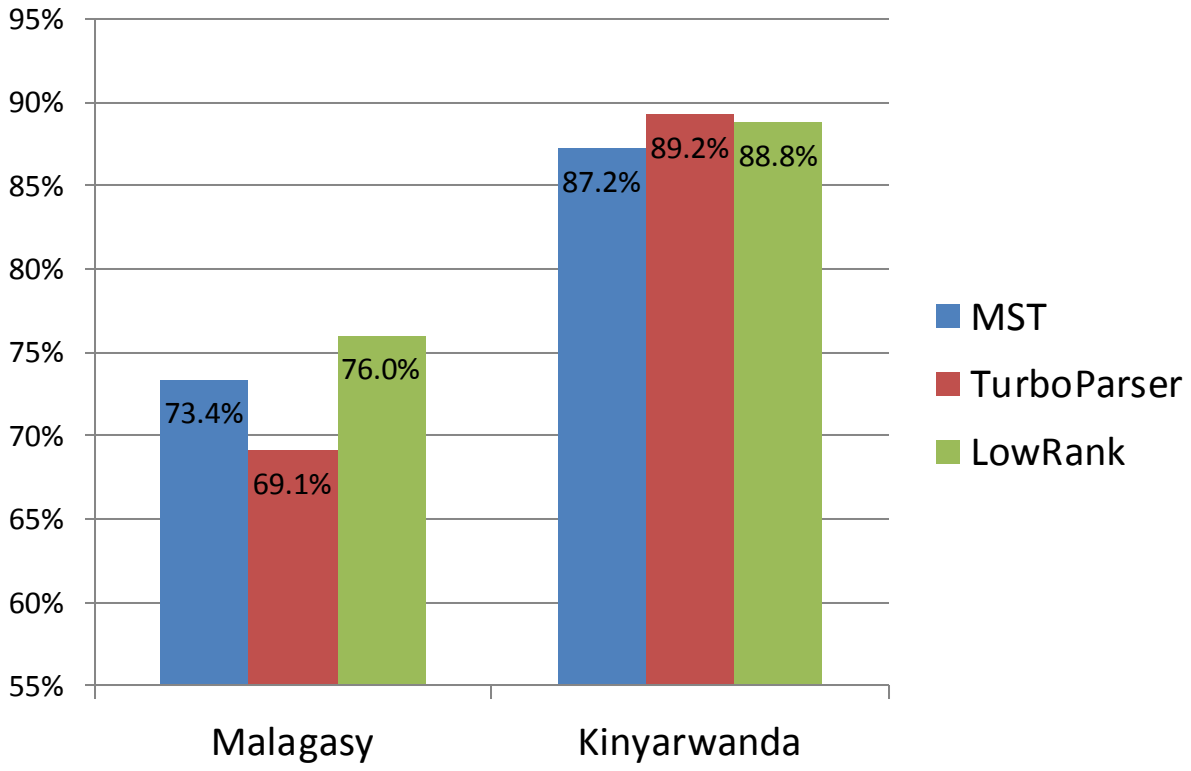
  Set $\alpha_k := \frac{2}{k}$.

  Set $Z^{(k+1)} := Z^{(k)} + \alpha_k \left(v_k v_k^T - Z^{(k)}\right)$.

End for

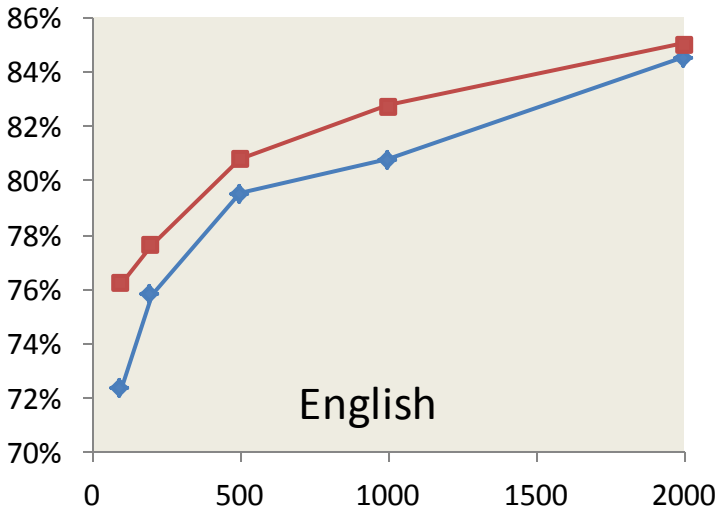
Results

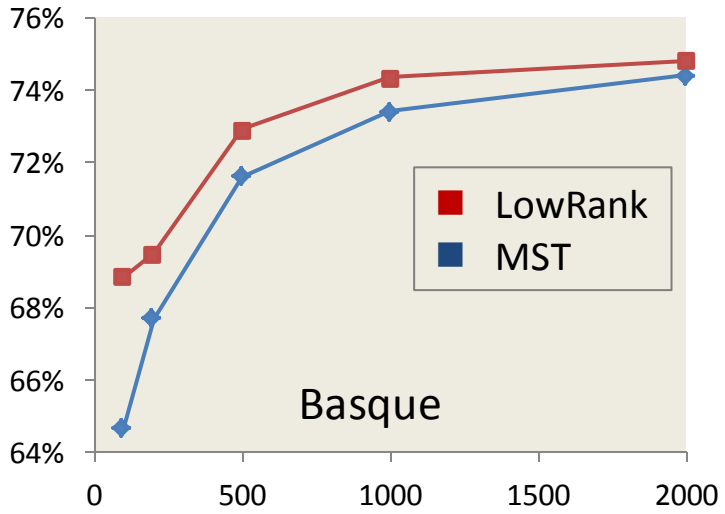
- Models trained with gold POS tags
- Evaluated without "#" tag at the end of sentences

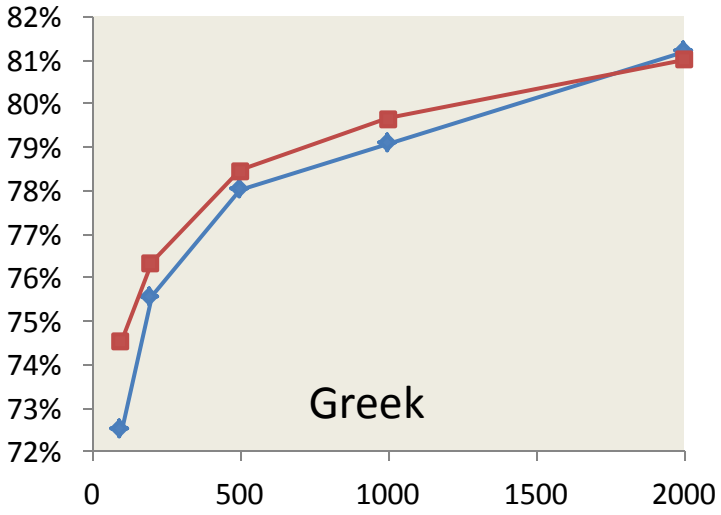


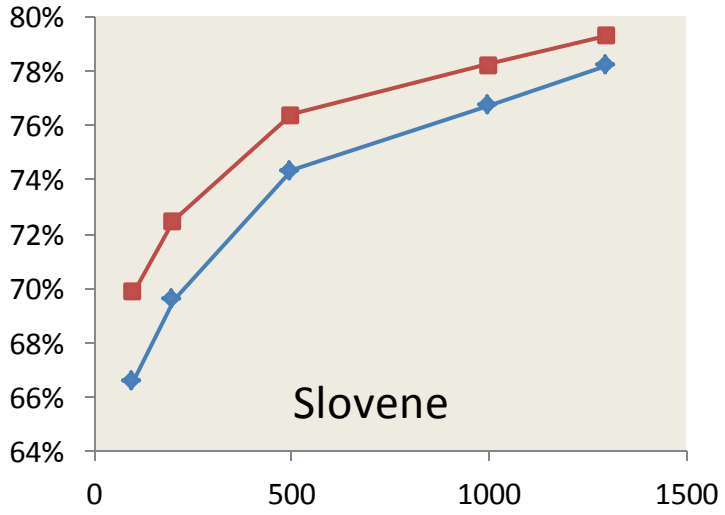
Results

- Results on CoNLL shared task (up to 2000 sentences)



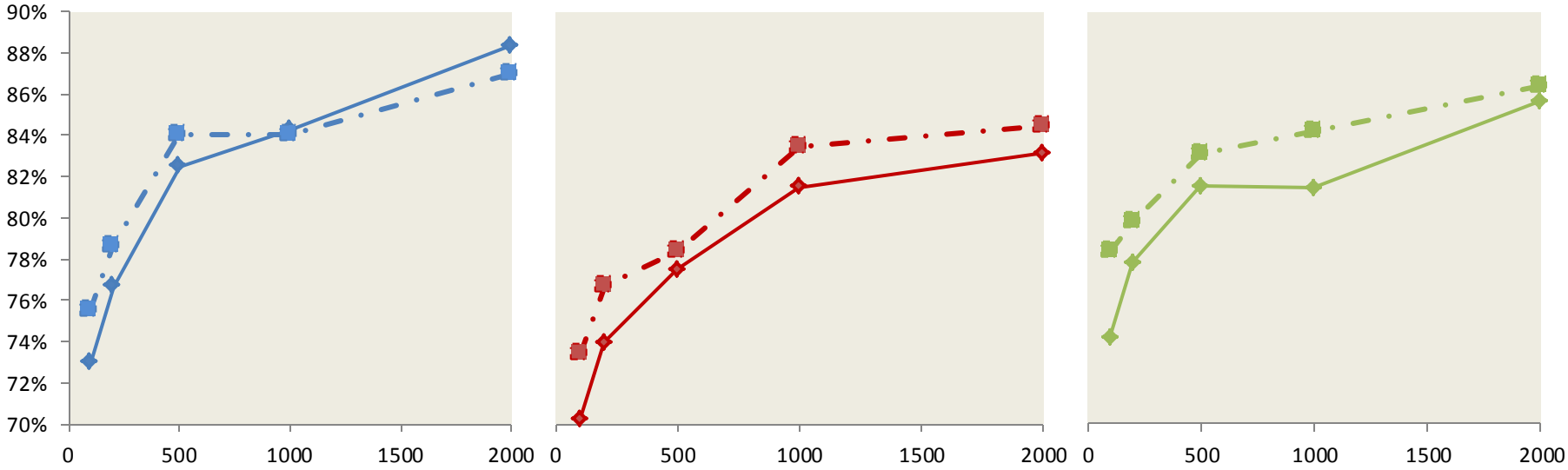






Results

- MST parser: solid lines
- Low-rank parser: doted lines



Seen Arc: See dependency in the training data

Seen Words: See the words in training but arc unseen

Unseen: At least one word not in training

Results

- Adding unsupervised word embeddings to English

	MST	MST+label	LowRank	LowRank+wv
100	72.5%	72.4%	76.3%	**76.6% (+0.3%)**
200	75.8%	75.8%	77.7%	**78.0% (+0.3%)**
500	79.4%	79.5%	80.8%	**81.4% (+0.6%)**
1000	80.9%	80.8%	**82.8%**	**82.8% (+0.0%)**
2000	83.7%	84.5%	85.1%	**85.8% (+0.7%)**

- **MST+label:** MST parser trained with labeled dependencies
- Other models are trained with only unlabeled dependencies.

Thanks

- Current implementation available at:

  http://people.csail.mit.edu/taolei/muri/lowrankparser.zip