Learning to Parse Using a Tiny Corpus

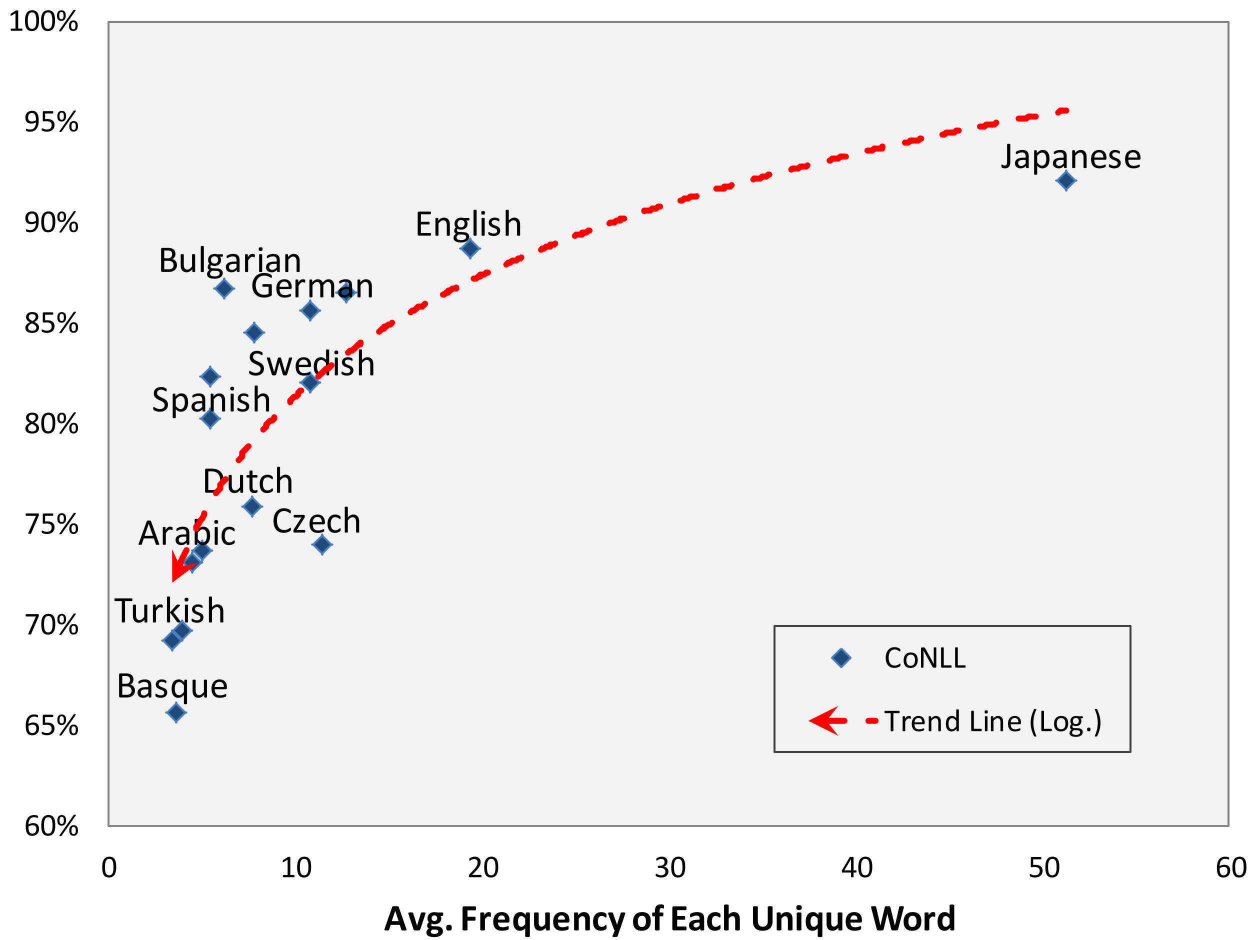
Tao Lei, Yu Xin

Regina Barzilay, Tommi Jaakkola

CSAIL, MIT

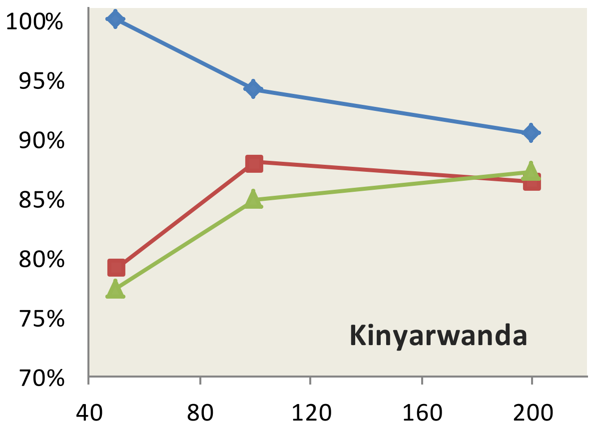
Sparsity Problem

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

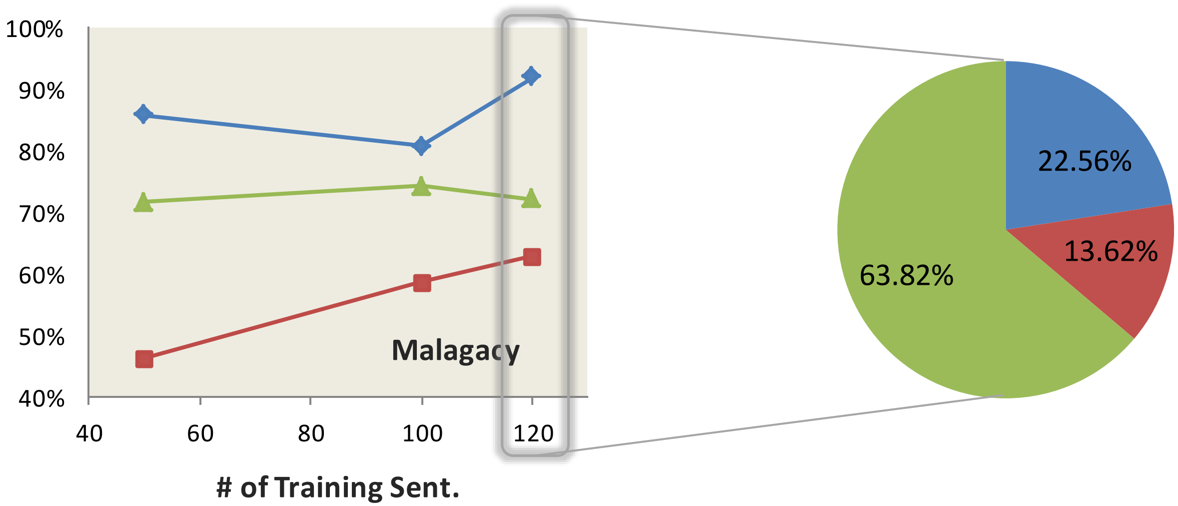


Sparsity Problem

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



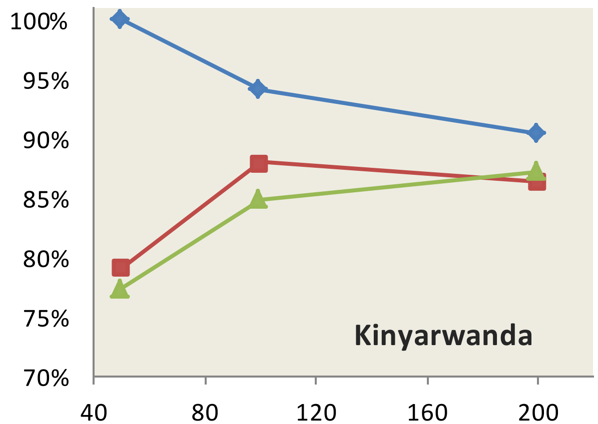
Seen Arc:	See dependency in the training data
Seen Words:	Only see the words in training
Unseen:	At least one word not in training

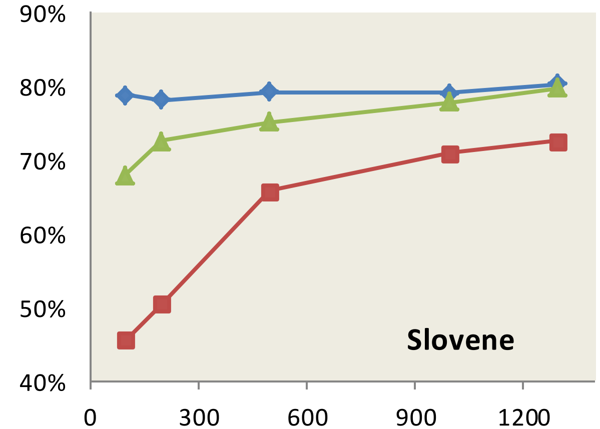


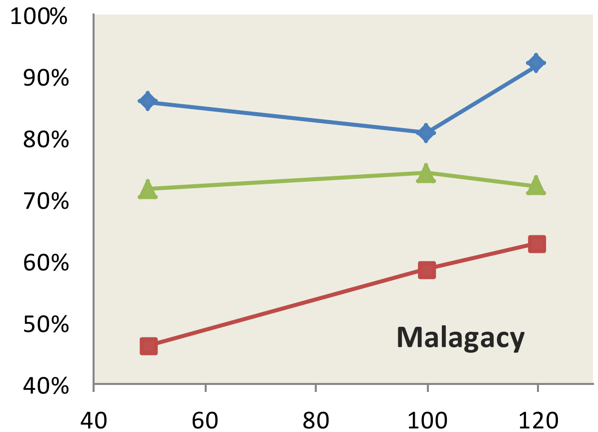
a large portion of dependency arcs in test is unseen

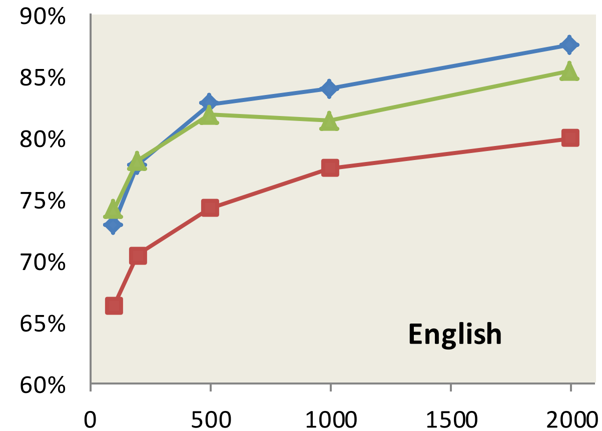
Sparsity Problem

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









Sparsity Problem

- Reason: feature weights are simply zero when the features are not seen

I PRON | eat VB | a DT | cake NN | with IN | frosting NN

eat$\oplus$IN$\oplus$frosting | VB$\oplus$IN$\oplus$NN | eat$\oplus$IN$\oplus$NN | …

0 + 1.1 + 0.9 = 2.0

unseen in training data

I PRON | eat VB | a DT | cake NN | with IN | frosting NN

cake$\oplus$IN$\oplus$frosting | NN$\oplus$IN$\oplus$NN | cake$\oplus$IN$\oplus$NN | …

0 + 1.2 + 0.5 = 1.7

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
  - Propagate 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

Motivating Example

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

eat apple
VB NN

Dependency Arc

{ eat⊕apple, eat⊕NN, VB⊕apple, VB⊕NN }

Feature Strings

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

Feature Matrix

$\otimes$

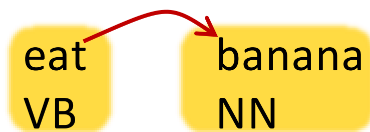
…	**2**	…	…	**4**
…	0	0	…	…
…	0	0	…	
…	**1**	0.9	…	**5**
…	0.1	0.1	…	…

Parameter Matrix

$= 12$

Motivating Example

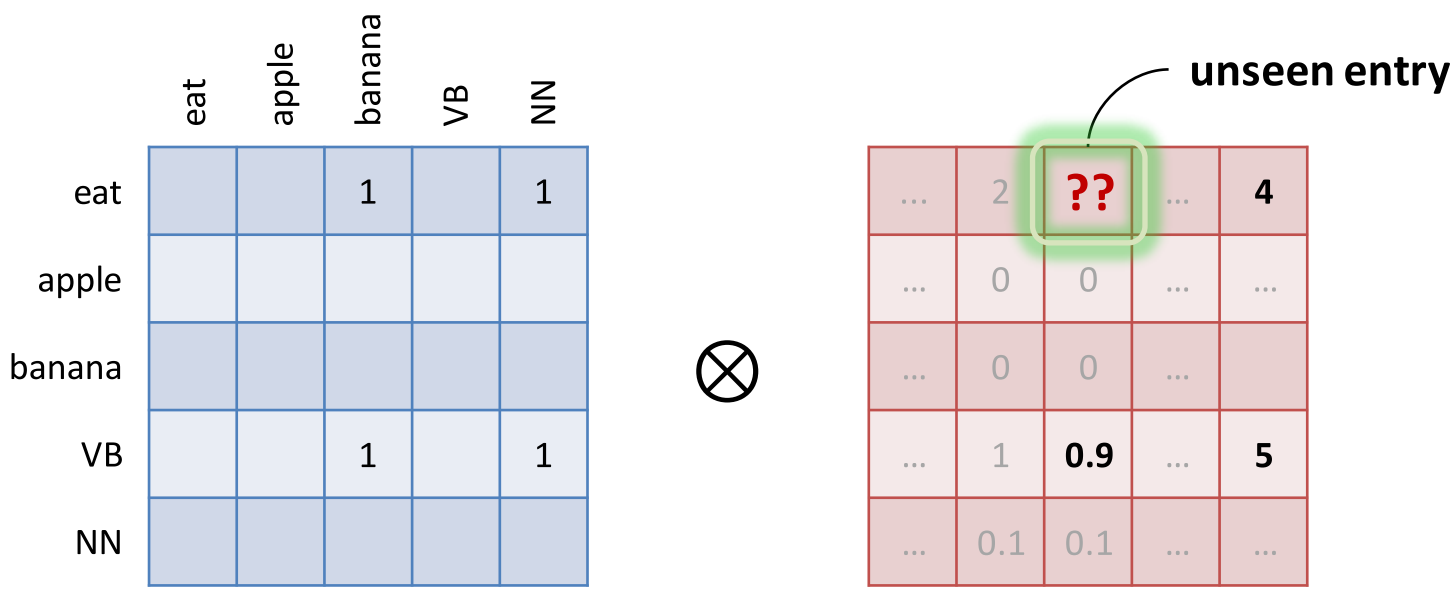
- In our case: learn a parameter matrix (or tensor) with unseen weights



Dependency Arc

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

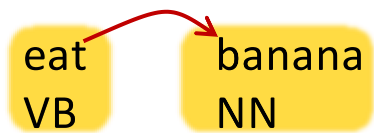
Feature Strings



Feature Matrix *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

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

Feature Matrix

$\otimes$

similar columns because "apple" and "banana" have similar syntactic behavior

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

Parameter Matrix

Motivating Example

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

eat VB → banana NN

Dependency Arc

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

Feature Strings

$$
\begin{array}{c|ccccc|}
 & \text{eat} & \text{apple} & \text{banana} & \text{VB} & \text{NN} \\
\hline
\text{eat} & & & 1 & & 1 \\
\text{apple} & & & & & \\
\text{banana} & & & & & \\
\text{VB} & & & 1 & & 1 \\
\text{NN} & & & & & \\
\hline
\end{array}
\otimes
\begin{array}{|ccccc|}
\hline
\cdots & 2 & \mathbf{2} & \cdots & \mathbf{4} \\
\cdots & 0 & 0 & \cdots & \cdots \\
\cdots & 0 & 0 & \cdots & \\
\cdots & 1 & \mathbf{0.9} & \cdots & \mathbf{5} \\
\cdots & 0.1 & 0.1 & \cdots & \cdots \\
\hline
\end{array}
= 11.9
$$

Feature Matrix *Parameter Matrix*

Formulation (Simplified)

- Recall first-order decoding objective:

$$\begin{aligned}\tilde{y}_i &= \operatorname*{argmax}_{y_i \in T(x_i)} S(y_i) \\ &= \operatorname*{argmax}_{y_i \in T(x_i)} \sum_{(h,c)\in y_i} \mathrm{s}(h,c)\end{aligned}$$

- Define score as matrix inner product:

$$s(h,c) = A \otimes \phi(h,c)$$

- Minimize the loss of training data:

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

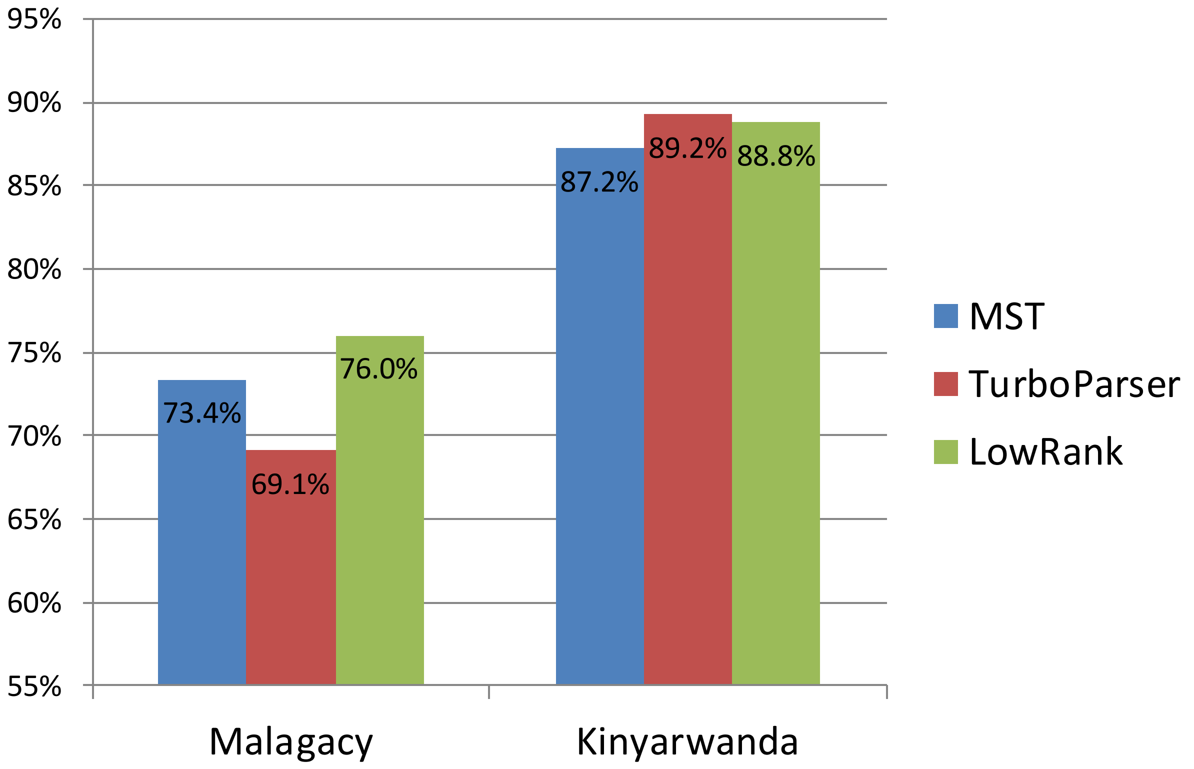
$$\text{s.t.} \quad \|A\|_* \leq C$$

Force A to be low-rank using nuclear norm constraint

  – online gradient descent algorithm available (Jaggi & Sulovsky, 2010) (Hazan, 2008)

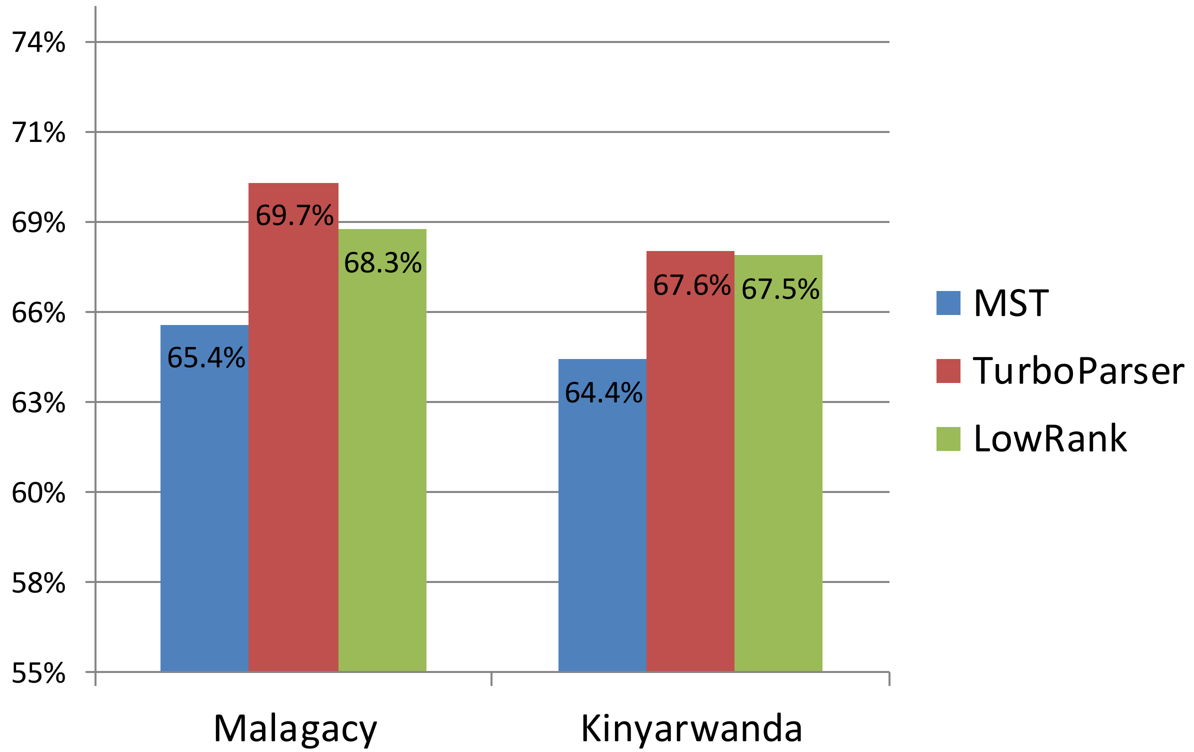
Results

- Models trained with gold POS tags



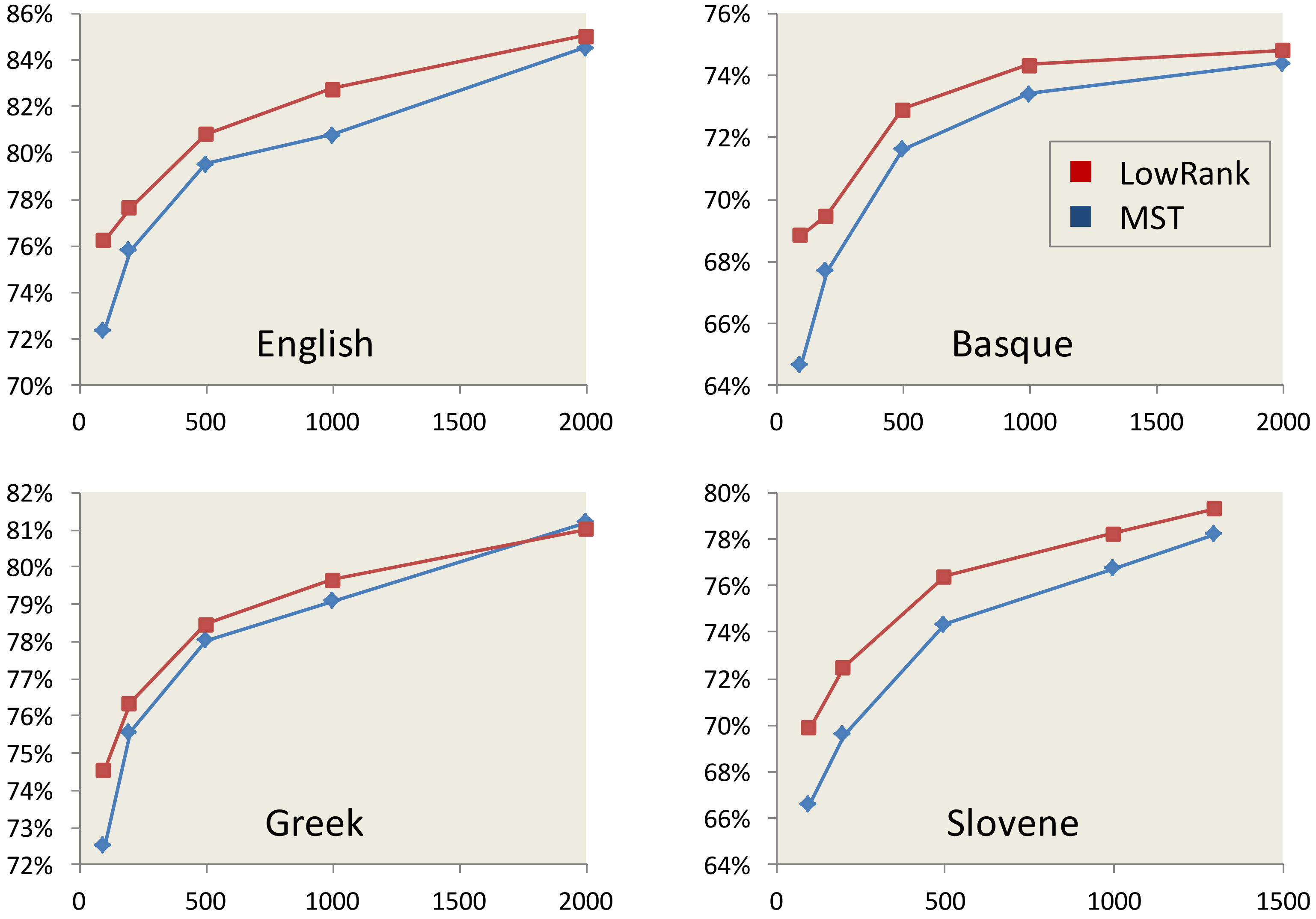
Results

- Models trained with auto POS tags by TurboTagger



Results

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



Results

- Adding unsupervised word embeddings to English

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