High-order Low-rank Tensors for Semantic Role Labeling

Yuan Zhang, Tao Lei, Regina Barzilay
Lluís Màrquez, Alessandro Moschitti

NAACL 2015

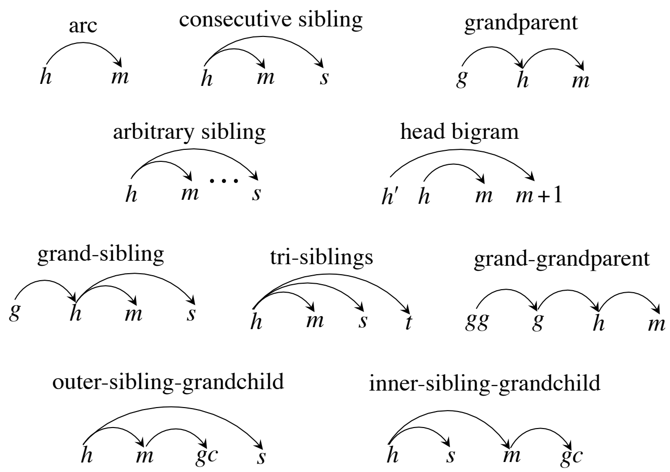
Structural Prediction

- Traditional structural prediction requires huge feature engineering

Example: syntactic dependency parsing

more than 10 groups of features

>100 feature templates



- Problem: feature sparsity, hard to generalize to unseen data

Structural Prediction

- Recent advance:

  learn low-dim. representations and their interactions (compositions) to achieve better generalization

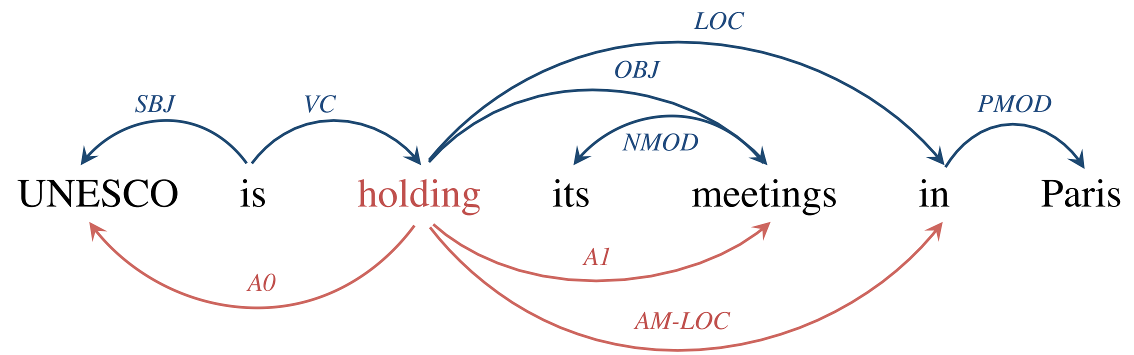
> ✦ Neural networks (Stenetorp 2013; Socher et al 2013; Chen and Manning 2014; Weiss et al 2015)
>
> ✦ Tensor factorization (Quattoni et al 2014; Lei et al 2014; Srikumar and Manning 2014)

- In this work,

  we extend our tensor factorization method to SRL

Feature Construction in SRL

- Features defined over tuples (*pred*, *arg*, *role*, *path*)
  - ✦ *pred* — predicate holding
  - ✦ *arg* — argument UNESCO
  - ✦ *role* — role label *A0*
  - ✦ *path* — syntactic path

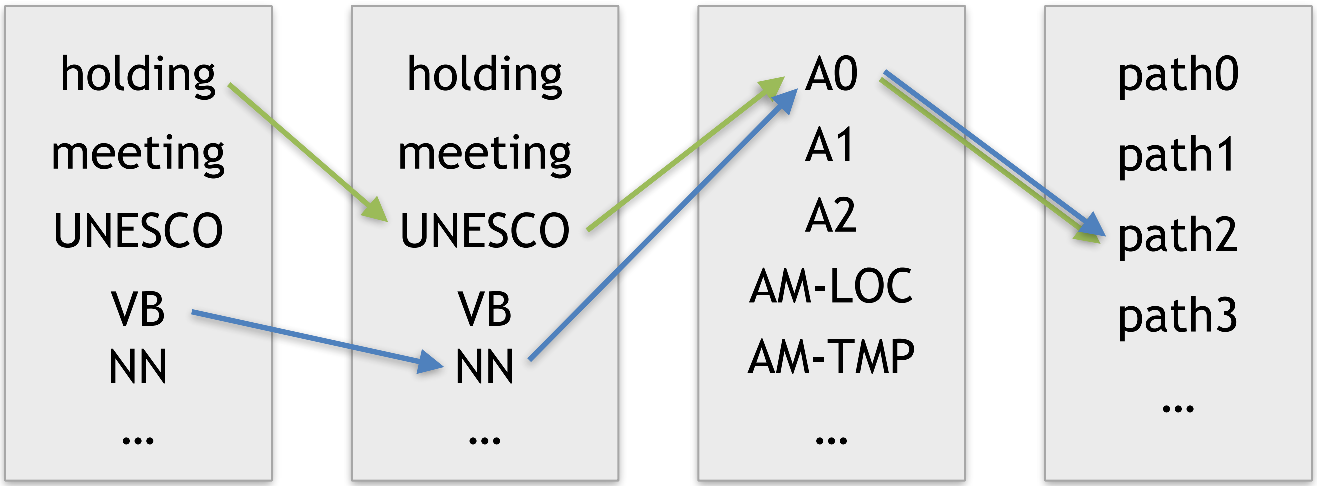


Example sentence

Feature Construction in SRL

- Features defined over tuples (*pred*, *arg*, *role*, *path*)

Selecting 1 up to 4 of them to construct features:



Each combination defines a feature

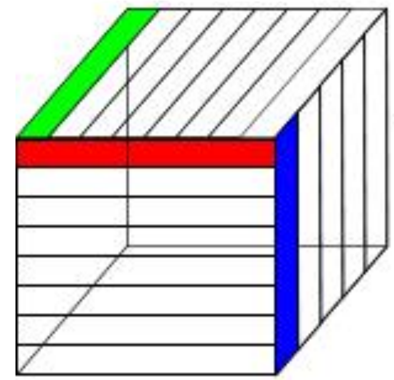
Needs to learn the corresponding feature weights (i.e. parameters)

A Tensor View of the Parameters

- Parameters of feature combinations indexed by a 4-way tensor:

A [*pred* , *arg* , *role* , *path*]

pred	*arg*	*role*	*path*
holding	holding	A0	path0
meeting	meeting	A1	path1
UNESCO	UNESCO	A2	path2
VB	VB	AM-LOC	path3
NN	NN	AM-TMP	…
…	…	…	

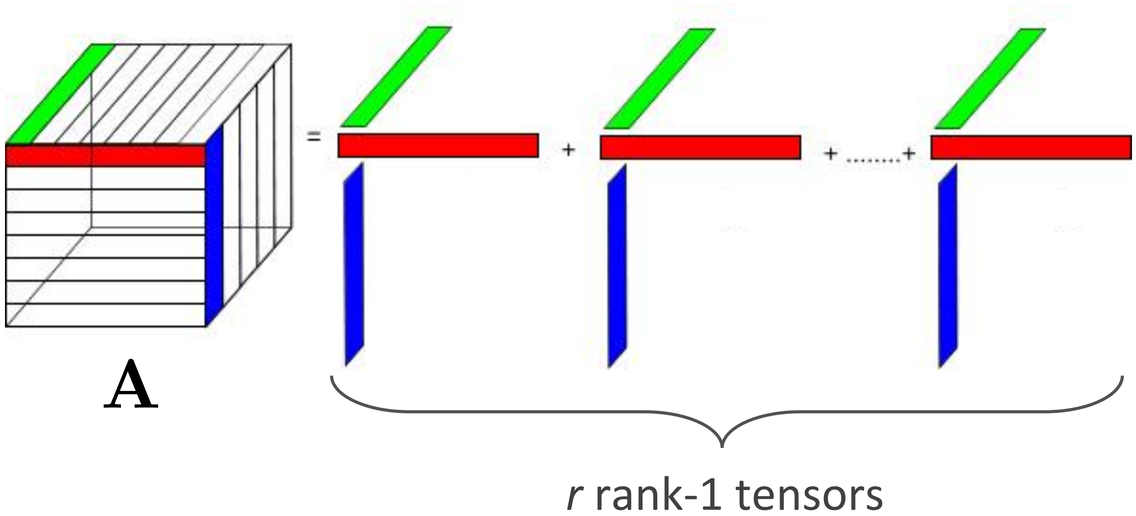


Entries of A stores the feature weights

Avoid Explosion via Low-rank

- Learn a low-rank factorization of A, optimized for parsing

$$A = \sum_{i=1}^{r} P(i) \otimes Q(i) \otimes R(i) \otimes S(i)$$



** here we use 3-way tensor for better visualization*

Online Learning

- Adopt standard mar-margin framework

$$\forall \mathbf{z}_{\text{sem}} \in \mathcal{Z}(\hat{\mathbf{x}}, \hat{\mathbf{y}}_{\text{syn}}) :$$

$$S_{sem}(\hat{\mathbf{x}}, \hat{\mathbf{y}}_{\text{syn}}, \hat{\mathbf{z}}_{\text{sem}}) \geq S_{sem}(\hat{\mathbf{x}}, \hat{\mathbf{y}}_{\text{syn}}, \mathbf{z}_{\text{sem}}) + cost(\hat{\mathbf{z}}_{\text{sem}}, \mathbf{z}_{\text{sem}})$$

score of gold structure *score of pred. structure* *margin*

Optimize parameters to satisfy this as much as possible

- Jointly update all parameter matrices via a new modified version of passive-aggressive algorithm

$$\Delta \boldsymbol{\theta} = \max \left\{ C, \frac{loss(\boldsymbol{\theta})}{\|g\boldsymbol{\theta}\|^2} \right\} g\boldsymbol{\theta}$$

Tensor Initialization

- Performance can be impacted by initial values of P,Q,R,S

- Basic initialization steps:

(i) learn a traditional model, obtain sparse subset of parameter values

(ii) store the values as a sparse tensor *T*

(iii) find a low-rank approximation of T

$$\min_{P,Q,R,S} \|T - \sum_i P(i) \otimes Q(i) \otimes R(i) \otimes S(i)\|_2^2$$

Tensor Initialization

- Performance can be impacted by initial values of P,Q,R,S

- Basic initialization steps:

(i) learn a traditional model, obtain sparse subset of parameter values

(ii) store the values as a sparse tensor *T*

(iii) find a low-rank approximation of T

$$\min_{P,Q,R,S} \|T - \sum_i P(i) \otimes Q(i) \otimes R(i) \otimes S(i)\|^2$$

In our previous work (Lei et al 2014), we use SVD initialization, which doesn't apply here

Iterative Power Method for Initialization

- Approximately find one component — P(i), Q(i), R(i) and S(i) using an iterative algorithm, one by one

Randomly initialize four unit vectors p, q, r and s
$T' = T - \sum_j P(j) \otimes Q(j) \otimes R(j) \otimes S(j)$
repeat
    $p = \langle T', -, q, r, s \rangle$ and normalize it
    $q = \langle T', p, -, r, s \rangle$ and normalize it
    $r = \langle T', p, q, -, s \rangle$ and normalize it
    $s = \langle T', p, q, r, - \rangle$
    $norm = \|s\|_2^2$
until $norm$ converges
$P(i) = p$ and $Q(i) = q$
$R(i) = r$ and $S(i) = s$

Optimize one vector while fixing the other three

Experimental Setup

- **Decoding**: weighted bipartite assignment (Lluís et al. 2013)
- **Dataset**: CoNLL-2009 joint syntactic and semantic parsing
- **Features:**

  a traditional set of 14 templates (Johansson, 2009)

  + our tensor component

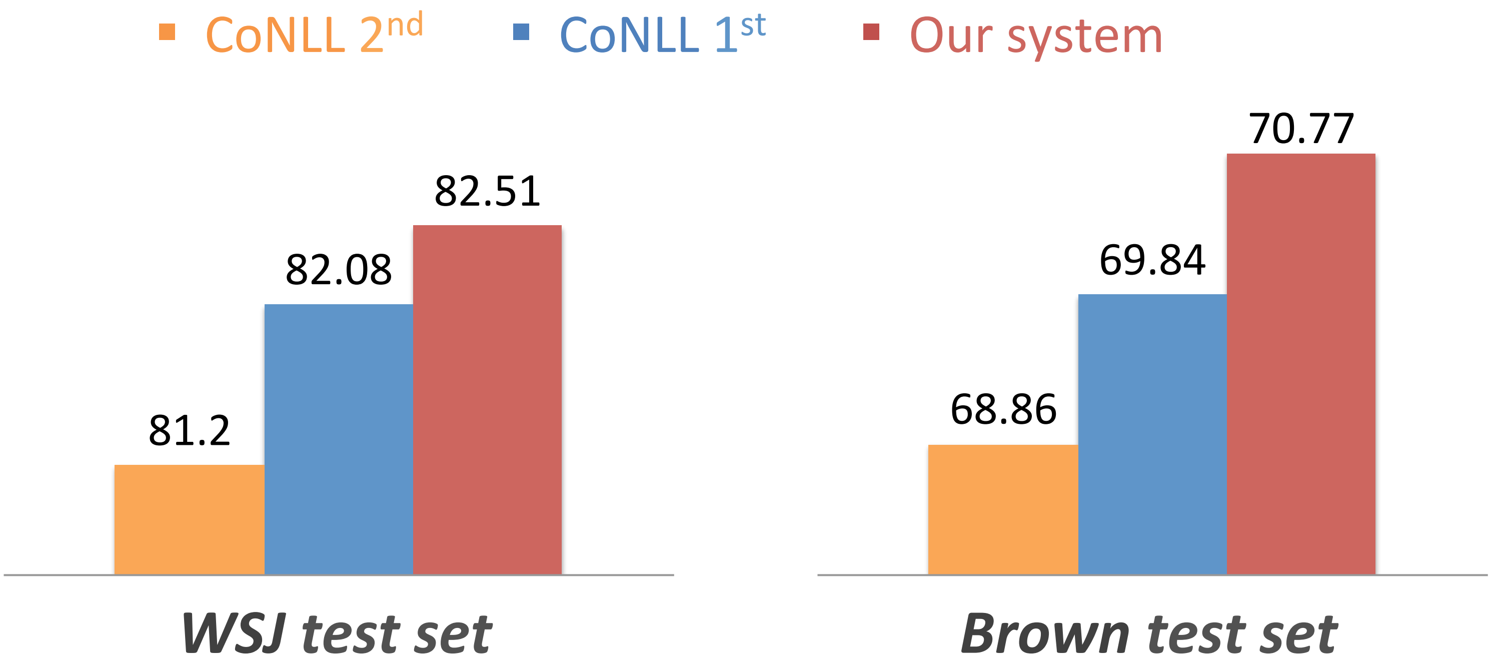
- **Baselines:**

  best systems participated CoNLL-2009 and their improved versions

  (Che et al., 2009; Zhao et al., 2009; Bjorkelund et al., 2010; Roth and Woodsend, 2014)

All explored much richer feature sets, language-specific tuning and system combination

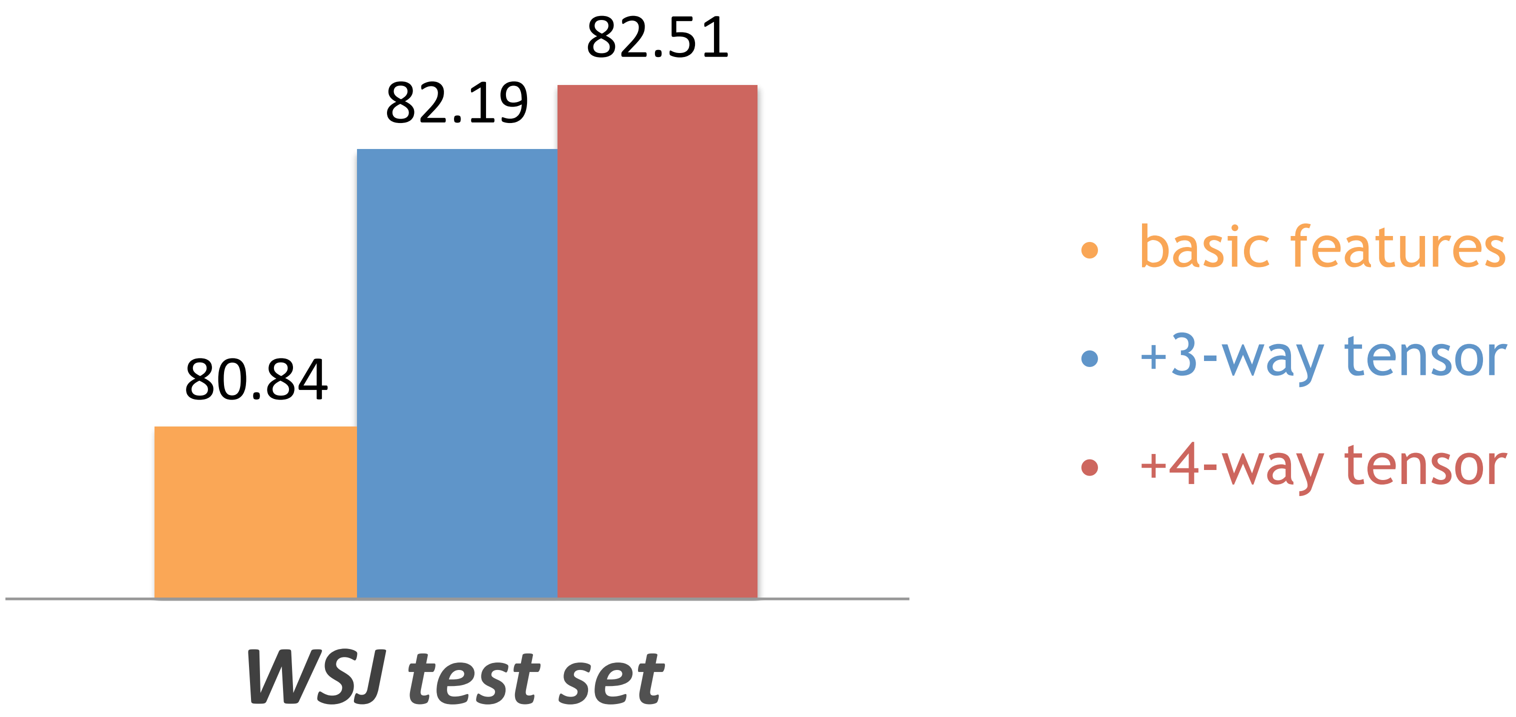
Result on English



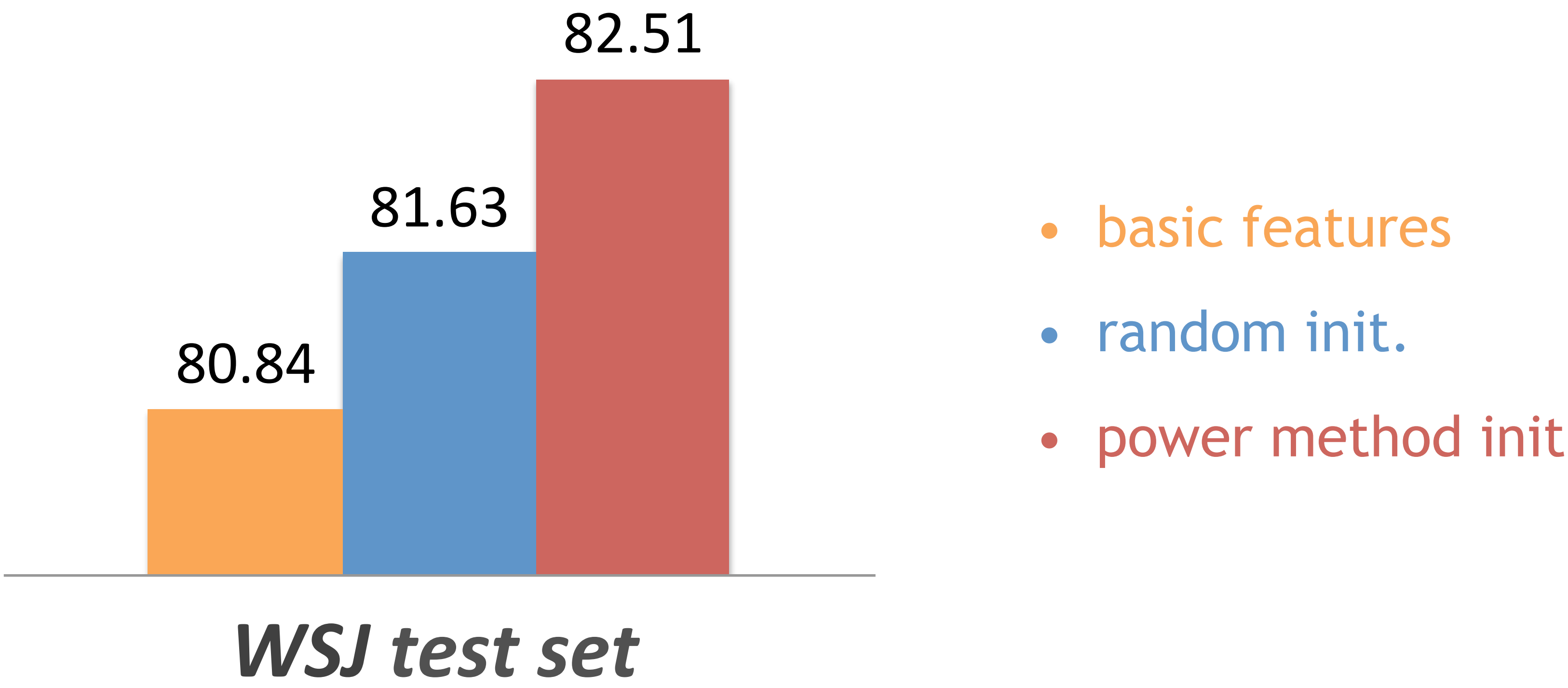
outperforms best single system (w/o reranking) with statistical significance

3-way vs. 4-way tensor

3-way tensor by merging “role” and “path” into one mode



Random vs. PM Initialization



Overall Improvement

Dataset	w/ tensor	w/o tensor
English	82.51	80.84
Catalan	74.67	71.86
Chinese	69.16	68.43
German	76.94	74.03
Spanish	75.58	72.85
Average	75.77	73.60

Adding tensor component leads to > 2% absolute gain in F-score

Thank you!

- RBG dependency parser
  https://github.com/taolei87/RBGParser
- Semantic role labeling parser
  https://github.com/taolei87/SRLParser

Overall Improvement

Dataset	w/ tensor	w/o tensor	CoNLL-1 (Zhao et al)
English	**82.51**	80.84	82.08
Catalan	74.67	71.86	**76.78**
Chinese	**69.16**	68.43	68.52
German	**76.94**	74.03	74.65
Spanish	75.58	72.85	**77.33**
Average	75.77	73.60	75.84