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
  - \(arg\) — argument
  - \(role\) — role label
  - \(path\) — syntactic path

Example sentence

UNESCO is holding its meetings in Paris

Example sentence
Feature Construction in SRL

• Features defined over tuples (\textit{pred}, \textit{arg}, \textit{role}, \textit{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:

\[ A \]

\[ \begin{align*}
\text{pred} & : \text{holding meeting UNESCO VB NN ...} \\
\text{arg} & : \text{holding meeting UNESCO VB NN ...} \\
\text{role} & : A0 A1 A2 AM-LOC AM-TMP ... \\
\text{path} & : \text{path0 path1 path2 path3 ...}
\end{align*} \]

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

\[ \forall z_{sem} \in Z(\hat{x}, y_{syn}) : \]
\[ S_{sem}(\hat{x}, y_{syn}, z_{sem}) \geq S_{sem}(\hat{x}, y_{syn}, z_{sem}) + \text{cost}(z_{sem}, z_{sem}) \]

*score of gold structure* \hspace{1cm} *score of pred. structure* \hspace{1cm} *margin*

Optimize parameters to satisfy this as much as possible

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

\[ \Delta \theta = \max \left\{ C, \frac{\text{loss}(\theta)}{\|g\theta\|^2} \right\} g\theta \]
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

1: Randomly initialize four unit vectors p, q, r and s
2: \( T' = T - \sum_j P(j) \otimes Q(j) \otimes R(j) \otimes S(j) \)
3: repeat
4: \( p = \langle T', -, q, r, s \rangle \) and normalize it
5: \( q = \langle T', p, -, r, s \rangle \) and normalize it
6: \( r = \langle T', p, q, -, s \rangle \) and normalize it
7: \( s = \langle T', p, q, r, - \rangle \)
8: \( \text{norm} = \|s\|_2^2 \)
9: until norm converges
10: \( P(i) = p \) and \( Q(i) = q \)
11: \( R(i) = r \) 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

- CoNLL 2\textsuperscript{nd}
- CoNLL 1\textsuperscript{st}
- Our system

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

- basic features
- +3-way tensor
- +4-way tensor

WSJ test set
Random vs. PM Initialization

- basic features
- random init.
- power method init

WSJ test set

80.84
81.63
82.51
Overall Improvement

<table>
<thead>
<tr>
<th>Dataset</th>
<th>w/ tensor</th>
<th>w/o tensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>82.51</td>
<td>80.84</td>
</tr>
<tr>
<td>Catalan</td>
<td>74.67</td>
<td>71.86</td>
</tr>
<tr>
<td>Chinese</td>
<td>69.16</td>
<td>68.43</td>
</tr>
<tr>
<td>German</td>
<td>76.94</td>
<td>74.03</td>
</tr>
<tr>
<td>Spanish</td>
<td>75.58</td>
<td>72.85</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>75.77</strong></td>
<td><strong>73.60</strong></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>w/ tensor</th>
<th>w/o tensor</th>
<th>CoNLL-1 (Zhao et al)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>82.51</td>
<td>80.84</td>
<td>82.08</td>
</tr>
<tr>
<td>Catalan</td>
<td>74.67</td>
<td>71.86</td>
<td>76.78</td>
</tr>
<tr>
<td>Chinese</td>
<td>69.16</td>
<td>68.43</td>
<td>68.52</td>
</tr>
<tr>
<td>German</td>
<td>76.94</td>
<td>74.03</td>
<td>74.65</td>
</tr>
<tr>
<td>Spanish</td>
<td>75.58</td>
<td>72.85</td>
<td>77.33</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>75.77</strong></td>
<td><strong>73.60</strong></td>
<td><strong>75.84</strong></td>
</tr>
</tbody>
</table>