Learning Latent Linguistic Structure to Optimize End Tasks

David A. Smith
with Jason Naradowsky
and Xiaoye “Tiger” Wu

12 October 2012
Learning Latent Linguistic Structure to Optimize End Tasks

David A. Smith
with Jason Naradowsky
and Xiaoye “Tiger” Wu

12 October 2012
Why Use NLP?

2 oh, we need consecutive
1 we need 6 and 7
2 i have 8 9 4
2 why do n’t you get 7
Why Use NLP?

K    2    oh, we need consecutive
S    1    we need 6 and 7
S    2    i have 8 9 4
C    2    why don’t you get 7
Why Use NLP?

<table>
<thead>
<tr>
<th>K</th>
<th>str</th>
<th>2</th>
<th>oh, we need consecutive</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>hnd</td>
<td>2</td>
<td>we need 6 and 7</td>
</tr>
<tr>
<td>S</td>
<td>pup</td>
<td>2</td>
<td>i have 8 9 4</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>2</td>
<td>why don’t you get 7</td>
</tr>
</tbody>
</table>
## Why Use NLP?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>2</td>
<td>oh, we <strong>need</strong> consecutive</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>str</td>
<td>1</td>
<td>we <strong>need</strong> 6 and 7</td>
</tr>
<tr>
<td>S</td>
<td>hnd</td>
<td>2</td>
<td>i <strong>have</strong> 8 9 4</td>
</tr>
<tr>
<td>C</td>
<td>pup</td>
<td>2</td>
<td>why do n’t you <strong>get</strong> 7</td>
</tr>
</tbody>
</table>
Why Use NLP?

K 2 oh, we need consecutive
S str 1 we need 6 and 7
S hnd 2 i have 8 9 4
C pup 2 why don’t you get 7
Task-directed Learning

What does the cow say?
The cow says ‘moooo’
Language Learning

feedback

KB
say(cow, ‘moo’) = 0.87

REL:say
the cow says moo

utterance

∞

the cow says moo

I'm an infant scientist
How to Use NLP?

K 2 oh, we need consecutive
S str 1 we need 6 and 7
S hnd 2 i have 8 9 4
C pup 2 why do n’t you get 7
**How to Use NLP?**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>2</td>
<td>oh, we need consecutive</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>1</td>
<td>we need 6 and 7</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>2</td>
<td>i have 8 9 4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>why don’t you get 7</td>
<td></td>
</tr>
</tbody>
</table>

Can we parse these as coordinations?
How to Use NLP?

Do we need training data for tagging, parsing, entity detection, coref, etc.?

Can we parse these as coordinations?

| K | oh, we need consecutive |
| S | we need 6 and 7 |
| S | i have 8 9 4 |
| C | why don’t you get 7 |
How to Use NLP?

Isn’t predicting this, or other game states, what we care about?

Do we need training data for tagging, parsing, entity detection, coref, etc.?

Can we parse these as coordinations?

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>2</td>
<td>oh, we need consecutive</td>
</tr>
<tr>
<td>S</td>
<td>str</td>
<td>1</td>
</tr>
<tr>
<td>S</td>
<td>hnd</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>pup</td>
<td>2</td>
</tr>
</tbody>
</table>
The Story So Far

• Joint inference prevents propagating errors
• But...
• Out-of-domain data plays havoc with NLP
• Joint training requires *everything* to be annotated
Learning Latent Representations

- Require less training data for...
  - ...each stage in the pipeline
  - ...adapting to a new domain
- Efficient inference and learning for coupling structured observed and latent variables
- Features inspired by pipelined models
This Talk

• Motivation

• Latent representations from auxiliary tasks

• Latent representations from structured end tasks
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Start state:  \(([[], [1, \ldots, n], \{\})\)

Final state:  \((S, [], A)\)

Shift:  \((S, i|B, A) \Rightarrow (S|i, B, A)\)

Reduce:  \((S|i, B, A) \Rightarrow (S, B, A)\)

Right-Arc:  \((S|i, j|B, A) \Rightarrow (S|i|j, B, A \cup \{i \rightarrow j\})\)

Left-Arc:  \((S|i, j|B, A) \Rightarrow (S, j|B, A \cup \{i \leftarrow j\})\)
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack  Buffer  Arcs
[ ]_S [who, did, you, see]_B {}
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack | Buffer | Arcs
--- | --- | ---
[who]_S | [did, you, see]_B | {}
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack  Buffer  Arcs

[]$_S$  [did, you, see]$_B$  \{ who $\leftarrow^\text{OBJ}$ did \}

Left-arc

OBJ

who  did  you  see
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack

Buffer

Arcs

\[
\begin{align*}
\text{Stack} & : [\text{did}]_S \\
\text{Buffer} & : [\text{you, see}]_B \\
\text{Arcs} & : \{ \text{who} \xrightarrow{\text{OBJ}} \text{did} \}
\end{align*}
\]

Shift

\[
\begin{align*}
\text{OBJ} & \quad \text{SBJ} \\
\text{who} & \quad \text{did} \quad \text{you} \quad \text{see}
\end{align*}
\]
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack:
[did, you]_S

Buffer:
[see]_B

Arcs:
{ who \stackrel{OBJ}{\leftarrow} did, 
did \stackrel{SBJ}{\rightarrow} you }
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack
[did]_S

Buffer
[see]_B

Arcs
{ who ← OBJ did, did → SBJ you }

Reduce

who

did

you

see
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack
[did, see]_S

Buffer
[ ]_B

Arcs
{ who \(\xleftarrow{\text{OBJ}}\) did,
   did \(\xrightarrow{\text{SBJ}}\) you,
   did \(\xrightarrow{\text{VG}}\) see }

Right-arc
VG

\(\text{OBJ}\)

\(\text{SBJ}\)

who
did
you
see
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack
[did, you]_S

Buffer
[see]_B

Arcs
{ who \xrightarrow{OBJ} did, 
did \xrightarrow{SBJ} you }

Right-arc

SBJ

who \xrightarrow{OBJ} did

you \xrightarrow{SBJ} see

VG
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack

[did, you]_S

Buffer

[see]_B

Arcs

\{ who \xleftarrow{OBJ} did, \\
   did \xrightarrow{SBJ} you \}
Transition-Based Parsing

Arc-eager shift-reduce parsing (Nivre, 2003)

Stack
[did, you]S

Buffer
[s]

Arcs

Very fast linear-time performance
WSJ 23 (2k sentences) in 3 s

did → you

Right-arc

SBJ

Choose action w/best classifier score
100k - 1M features

VG
Better Features for Parsing

- Baseline
- 7-gram
- +7-gram
- +7-gram +ext
- +7-dep clusters
- 7-dep
- +7-dep
- +7-dep +ext

% Accuracy

Unlabeled
Classifying Actions

Stack
[did]_S

Buffer
[you, see]_B

Arcs
{ who OBJ did }

Shift

who

did

you

see
Classifying Actions

Stack
[did]_S

Buffer
[you, see]_B

Arcs
{ who ^OBJ did }

Output: Action [+ Label]
Classifying Actions

Parsing Methods
- Stack
- Buffer
- Arcs

Stack: \([\text{did}]_S\)
Buffer: \([\text{you, see}]_B\)
Arcs: \{\text{who} \stackrel{\text{OBJ}}{\leftarrow} \text{did}\}

Output: Action \([\text{+ Label}]\)

POS, Form in buffer; lookahead 3
Classifying Actions

Stack  
[did]_S

Buffer  
[you, see]_B

Arcs  
\{ who \overset{OBJ}{\leftarrow} did \}

POS, Form of top 2 stack items

POS, Form in buffer; lookahead 3

Shift

Output: Action [+ Label]
Classifying Actions

- **Stack**: [did] \_S
- **Buffer**: [you, see] \_B
- **Arcs**: \{ who \_OBJ did \}

**Output**: Action [+ Label]

- Dep. label of words in stack, buffer
- Shift
- POS, Form of top 2 stack items
- POS, Form in buffer; lookahead 3
- Dep. label of words in stack, buffer
- [did] \_S
- [you, see] \_B
- \{ who \_OBJ did \}

\[ \text{who} \text{ did} \]
\[ \text{you} \hspace{1cm} \text{see} \]
Classifying Actions

**Stack**

\[ \text{[did]}_S \]

**Buffer**

\[ \text{[you, see]}_B \]

**Arcs**

\{ \text{who} \xleftarrow{\text{OBJ}} \text{did} \}

**Dep. label of words in stack, buffer**

- \text{POS, Form of top 2 stack items}
- \text{POS, Form in buffer; lookahead 3}
- \text{Dep. label of words in stack, buffer}

Output: Action \([+ \text{ Label}]\)

Can these be made less sparse?
Distributional Embeddings

Keyword in 7-gram context

among them the winner in Paris in becoming a Snively winner, I was but onto the 77 winner mark with Bishops

The recent Newbury winner has been installed to be the winner for the second

We've got the winner on the phone
Distributional Embedding

became the first female \textit{winner} of the prestigious Nobel Prize
Distributional Embedding

became the first female winner of the prestigious Nobel Prize
Distributional Embedding

Keyword in 7-dependency context

became the first female **winner** of the prestigious **Nobel Prize**
Distributional Embedding

Discriminative language model training

became the first female winner of the prestigious Nobel Prize capital
Distributional Embedding

Discriminative language model training

became the first female winner of the prestigious Nobel Prize

capital

Learn to predict observed word
Distributional Embedding

Discriminative language model training

became the first female winner of the prestigious Nobel Prize
Distributional Embedding

Discriminative language model training

became the first female **winner** of the prestigious Nobel Prize

Lookup in codebook

50 dim. 50 dim. 50 dim. 50 dim. 50 dim. 50 dim. 50 dim.
Distributional Embedding

Discriminative language model training

began the first female winner of the prestigious Nobel Prize

Lookup in codebook

50 dim. 50 dim. 50 dim. 50 dim. 50 dim. 50 dim. 50 dim.

3-layer perceptron

Correct word?
Distributional Embedding

Discriminative language model training

broke the first female winner of the prestigious Nobel Prize

Lookup in codebook

Correct word?
Distributional Embedding

Discriminative language model training

Words = points in 50d space

Backprop updates codes

Correct word?
Better Features for Parsing

- Baseline
- 7-gram
- +7-gram
- +7-gram + ext
- +7-dep clusters
- 7-dep
- +7-dep
- +7-dep + ext

% Accuracy

Unlabeled

84.17
Better Features for Parsing

- Baseline
- 7-gram
- +7-gram
- 7-dep
- +7-dep
- +7-dep +ext

% Accuracy

Unlabeled

84.17
82.95
Better Features for Parsing

- Baseline
- 7-gram
- +7-gram
- +7-gram +ext
- +7-dep clusters
- 7-dep
- +7-dep
- +7-dep +ext

% Accuracy

<table>
<thead>
<tr>
<th>Unlabeled</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>84.17</td>
</tr>
<tr>
<td>7-gram</td>
<td>82.95</td>
</tr>
<tr>
<td>+7-gram</td>
<td>83.33</td>
</tr>
<tr>
<td>+7-dep clusters</td>
<td>80.00</td>
</tr>
<tr>
<td>7-dep</td>
<td>82.50</td>
</tr>
<tr>
<td>+7-dep</td>
<td>83.00</td>
</tr>
<tr>
<td>+7-dep +ext</td>
<td>83.50</td>
</tr>
</tbody>
</table>
Better Features for Parsing

- Baseline
- 7-gram
- +7-gram
- +7-gram +ext
- +7-dep clusters
- 7-dep
- +7-dep
- +7-dep +ext

% Accuracy

Unlabeled
Better Features for Parsing

- Baseline
- 7-gram
- +7-gram
- 7-dep
- +7-dep
- 7-gram +ext
- +7-gram +ext
- +7-dep clusters
- 7-dep
- +7-dep
- +7-dep +ext

% Accuracy

Unlabeled

- 84.17
- 82.95
- 83.33
- 84.66
- 89.49
Better Features for Parsing

<table>
<thead>
<tr>
<th>Features</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>84.17</td>
</tr>
<tr>
<td>7-gram</td>
<td>82.95</td>
</tr>
<tr>
<td>+7-gram</td>
<td>83.33</td>
</tr>
<tr>
<td>+7-gram +ext</td>
<td>84.66</td>
</tr>
<tr>
<td>+7-dep clusters</td>
<td><strong>89.49</strong></td>
</tr>
<tr>
<td>7-dep</td>
<td></td>
</tr>
<tr>
<td>+7-dep</td>
<td></td>
</tr>
<tr>
<td>+7-dep +ext</td>
<td><strong>89.82</strong></td>
</tr>
</tbody>
</table>
Better Features for Parsing

- Baseline
- 7-gram
- +7-gram
- +7-gram +ext
- +7-dep clusters
- 7-dep
- +7-dep
- +7-dep +ext

Accuracy:
- Unlabeled:
  - Baseline: 84.17%
  - 7-gram: 82.95%
  - +7-gram: 83.33%
  - +7-gram +ext: 84.66%
  - +7-dep clusters: 89.49%
  - 7-dep: 89.82%
  - +7-dep: 89.74%
Better Features for Parsing

% Accuracy

- Baseline
- 7-gram
- +7-gram
- +7-gram +ext
- +7-dep clusters
- 7-dep
- +7-dep
- +7-dep +ext

Unlabeled

Accuracy:
- Baseline: 84.17%
- 7-gram: 82.95%
- +7-gram: 83.33%
- +7-gram +ext: 84.66%
- +7-dep clusters: 89.49%
- 7-dep: 89.82%
- +7-dep: 89.74%
- +7-dep +ext: 90.28%
Better Features for Parsing

% Accuracy

- Baseline
- +7-dep

ROOT | 1 | 2 | 3-6 | 7+
---|---|---|----|---
Baseline | 87.22 | 82 | 79.44 | 56.06
+7-dep | 92.18 | 86.71 | 72.12
Unsupervised Domain Adaptation

Baseline W embed B embed W+B embed

%Accuracy

WSJ

84.17 89.82 87.93 89.44

Brown

81.58 86.67 86.74 87.05
This Talk

- Motivation
- Latent representations from auxiliary tasks
- Latent representations from structured end tasks
Pierre Vinken, 61 years old, will join the board as a nonexecutive director.

Vinken’s joining the board as director is imminent.
Pierre Vinken, 61 years old, will join the board as a nonexecutive director.

Vinken’s joining the board as director is imminent.

<table>
<thead>
<tr>
<th>PRED</th>
<th>ARG0</th>
<th>ARG1</th>
<th>ARG-PRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>join</td>
<td>Vinken</td>
<td>board</td>
<td>director</td>
</tr>
</tbody>
</table>
Pierre Vinken, 61 years old, will join the board as a nonexecutive director.

Vinken’s joining the board as director is imminent.

<table>
<thead>
<tr>
<th>PRED</th>
<th>ARG0</th>
<th>ARG1</th>
<th>ARG-PRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>join</td>
<td>Vinken</td>
<td>board</td>
<td>director</td>
</tr>
</tbody>
</table>
Coupling Structured Variables

• Outputs are complex combinatorial objects
  ✤ E.g., sequences, trees, graphs, strings

• Intermediate representations are complex combinatorial objects
  ✤ E.g., syntax helps predict semantics

• Developed efficient inference and learning
Composable Syntax

- Semantic Role Labeling
- Relation Extraction
  - employs(ORGANIZATION, PERSON)
  - Dependency and constituency models
- Named-entity recognition
  - Couple constituency model with semi-CRF
  - No grammar composition
Global Constraints

• Dependency trees
  ✤ Cf. Projective & nonprojective (MST) parsing
• (Nested) bracketings
  ✤ Cf. CKY & semi-CRFs
• Matchings and alignments
  ✤ Weighted bipartite matching & network flow; ITG
• Intersecting specialized grammars (CFG, TAG, etc.)
• Other finite-state, context-free, etc., constraints...
• Compare TurboParsers and Dual Decomposition
Semantic Role Labeling

a.) Syntactic Combinatorial Constraint

b.) Syntactic Layer

c.) Argument Prediction

d.) Sense Prediction
Semantic Role Labeling

- Baseline
- Oracle syntax
- Hidden syntax

<table>
<thead>
<tr>
<th>Language</th>
<th>Baseline</th>
<th>Oracle syntax</th>
<th>Hidden syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Czech</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>German</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Labeled F1: 45 - 100
Hidden Syntax

$\text{Alles bleibt voller Widersprüche.}$

Everything remains full of contradictions.
Everything remains full of contradictions.
Hidden Syntax

Everything remains full of contradictions.
Hidden Syntax

Everything remains full of contradictions.

Heading sentence with punctuation doesn’t affect SRL.
There remains only the hope of a renewed Big Bang.
There remains only the hope of a renewed Big Bang.

It’s useful to have NPs consistent inside and outside of PPs.
Relation Extraction

Figure 1: Latent Dependency coupling for the RE task.

The D-C\textsc{ONNECT} factor expresses ternary connection relations because the shared head word of the proposed relation is unknown. As is convention, variables are represented by circles, factors by rectangles.

We introduce six model scenarios.

- **Baseline**, simply the arc-factored model consisting only of $\text{Rel}$ and corresponding $\text{Label}$ variables for each entity. Features on the relation factors, which are common to all model configurations, are combinations of lexical information (i.e., the words that form the entity, the pos-tags of the entities, etc.) as well as the distance between the relation. This is a lightweight model and generally does not attempt to exhaustively leverage all possible proven sources of useful features (Zhou et al., 2005) towards a higher absolute score, but rather to serve as a point of comparison to the models which rely on syntactic information.

- **Baseline-Ent**, a variant of Baseline with additional features which include combinations of mention type, entity type, and entity sub-type.

- **Oracle D-Parse**, in which we also instantiate a full set of latent dependency syntax variables, and connect them to the baseline model using D-C\textsc{ONNECT} factors. Syntax variables are clamped to their true values.

- **Oracle C-Parse**, the constituency syntax analogue of Oracle D-Parse.

- **Hidden D-Parse**, which is an extension of Oracle D-Parse in which we connect all syntax variables to a DEP-T\textsc{REE} factor, syntax variables are unobserved, and are learned jointly with the end task. The features for latent syntax are a subset of those used in dependency parsing (McDonald et al., 2005).

- **Hidden C-Parse**, the constituency syntax analogue of Hidden D-Parse. The feature set is similar but bigrams are taken over the words defining the constituent span, rather than the words defining the head/modifier relation.

Coordination factor features for the syntactically-informed models are particularly important. This became evident in initial experiments where the baseline was often able to outperform the hidden syntactic model. However, inclusion of entity and mention label features into the connection factors provides the model with greater reasoning over when to coordinate or ignore the relation predictions with the underlying syntax. These are a proper subset of the Baseline-Ent features.

### 3.3 Data

We evaluate these models using the 2005 Automatic Content Extraction (ACE) data set (Walker, 2006), using the English (dual-annotated) and Chinese (solely annotator #1 data set) sections. Each corpus is annotated with entity mentions—tagged as PER, ORG, LOC, or MISC—and, where applicable, what type of relation exists between them (e.g., coarse: PHYS; fine: Located). But like most corpora available for the task, the burden of acquiring corresponding syntactic annotation is left to the researcher. In this situation it is common to turn to existing pre-trained parsing models.

We generate our data by first splitting the raw text paragraphs into sentences. Chinese sentences...
Relation Extraction

- Baseline
- Parser Deps.
- Hidden Deps.
- Parser Constit.
- Hidden Constit.

Labeled F1
In Summary

- Representations trained on more appropriate auxiliary tasks support better learning
- Models with latent syntax models near performance of those with oracle syntax
- Assertion (see papers): Latent structured variables without asymptotic slowdown
Thank you
Hidden Distribution

For the coordinating factors we use subsets of combinations of word, part-of-speech, and capitalization features taken between head and argument, and concatenate these with the distance and direction between the predicate and argument. We do not find the performance of the system to be as sensitive to which features are present in the coordinating factors as we did in the RE task.

4.0.3 Data

Figure 4.1. Examining the learned hidden dependency representation for SRL. In this Japanese example, the syntactic dependency arcs derived from gold standard syntactic annotations (left) are entirely disjoint from the correct predicate/arguments pairs (shown in the heatmaps by the squares outlined in black), and the observed syntax model fails to recover any of the correct predictions. In contrast, the hidden model structure (right) learns a representation that closely parallels the desired end task predictions, helping it recover three of the four correct SRL predictions, and providing some evidence towards the fourth. The dependency tree corresponding to the hidden structure is derived by edge-factored decoding: dependency variables whose beliefs $> 0.5$ are classified as true (though some arcs not relevant to the SRL predictions are omitted for clarity).

We evaluate our SRL model using the data set developed for the CoNLL 2009 shared task competition [10], which features seven languages and provides an ideal opportunity to measure the ability of the hidden structure to generalize across languages of disparate origin and varied characteristics. It also provides the opportunity to observe a variety of...