Bootstrapping Dependency Grammars from Sentence Fragments via Austere Models

Valentin I. Spitkovsky

with Daniel Jurafsky (Stanford University)
and Hiyan Alshawi (Google Inc.)
Why do unsupervised learning?

- one practical reason
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences (WSJ15’s 16K cover 160K tokens)
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use *less* than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences (WSJ15’s 16K cover 160K tokens)

- long sentences are hard — shorter inputs can be easier
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences (WSJ15’s 16K cover 160K tokens)

- long sentences are hard — shorter inputs can be easier:
  - better chances of guessing larger fractions of correct trees
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use *less* than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences (WSJ15’s 16K cover 160K tokens)

- long sentences are hard — shorter inputs can be easier:
  - better chances of guessing larger fractions of correct trees
  - preference for more local structures (Smith and Eisner, 2006)
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences (WSJ15’s 16K cover 160K tokens)

- long sentences are hard — shorter inputs can be easier:
  - better chances of guessing larger fractions of correct trees
  - preference for more local structures (Smith and Eisner, 2006)
  - faster training, etc.
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use *less* than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences (WSJ15’s 16K cover 160K tokens)

- long sentences are hard — shorter inputs can be easier:
  - better chances of guessing larger fractions of correct trees
  - preference for more local structures (Smith and Eisner, 2006)
  - faster training, etc. — a rich history going back to Elman (1993)
Why do unsupervised learning?

- one practical reason:
  - got lots of potentially useful data!
  - but more than would be feasible to annotate...

- yet grammar inducers use less than supervised parsers:
  - most systems train on WSJ10 (or, more recently, WSJ15)
  - WSJ10 has approximately 50K tokens (5% of WSJ’s 1M)
  - in just 7K sentences (WSJ15’s 16K cover 160K tokens)

- long sentences are hard — shorter inputs can be easier:
  - better chances of guessing larger fractions of correct trees
  - preference for more local structures (Smith and Eisner, 2006)
  - faster training, etc.
  — a rich history going back to Elman (1993)

- ... could we “start small” and use more data?
How have long inputs been handled previously?
How have long inputs been handled previously?

- very carefully...
How have long inputs been handled previously?

- very carefully...

- Viterbi training (tolerates bad independence assumptions of models)
How have long inputs been handled previously?

- very carefully...

- Viterbi training (tolerates bad independence assumptions of models)

- punctuation-induced constraints (partial bracketing: Pereira and Schabes, 1992)
How have long inputs been handled previously?

- very carefully...

- Viterbi training (tolerates bad independence assumptions of models)

- + punctuation-induced constraints (partial bracketing: Pereira and Schabes, 1992)

- = punctuation-constrained Viterbi training
Example:

Punctuation (Spitkovsky et al., 2011)
Example: Punctuation (Spitkovsky et al., 2011)

Although it probably has reduced the level of expenditures for some purchasers,
**Example:**

Although it probably has reduced the level of expenditures for some purchasers, utilization management —

Punctuation (Spitkovsky et al., 2011)
Example: Punctuation (Spitkovsky et al., 2011)

Although it probably has reduced the level of expenditures for some purchasers, utilization management — like most other cost containment strategies —
**Example:**

Punctuation (Spitkovsky et al., 2011)

Although it probably has reduced the level of expenditures for some purchasers, like most other cost containment strategies doesn’t appear to have altered the long-term rate of increase in health-care costs,
**Example:**

Punctuation (Spitkovsky et al., 2011)

\[
\text{[SBAR Although it probably has reduced the level of expenditures for some purchasers], [NP utilization management] — [PP like most other cost containment strategies] — [VP doesn’t appear to have altered the long-term rate of increase in health-care costs], [NP the Institute of Medicine]},
\]
**Example**: Punctuation (Spitkovsky et al., 2011)

Although it probably has reduced the level of expenditures for some purchasers, utilization management — like most other cost containment strategies — doesn’t appear to have altered the long-term rate of increase in health-care costs, the Institute of Medicine, an affiliate of the National Academy of Sciences,
Example: Punctuation (Spitkovsky et al., 2011)

Although it probably has reduced the level of expenditures for some purchasers, utilization management — like most other cost containment strategies — doesn’t appear to have altered the long-term rate of increase in health-care costs, the Institute of Medicine, an affiliate of the National Academy of Sciences, concluded after a two-year study.
**Example:**

Punctuation (Spitkovsky et al., 2011)

\[ \text{Although it probably has reduced the level of expenditures for some purchasers}, [\text{utilization management}] \quad \text{—} \quad [\text{like most other cost containment strategies}] \quad \text{—} \quad [\text{doesn’t appear to have altered the long-term rate of increase in health-care costs}], [\text{the Institute of Medicine}], [\text{an affiliate of the National Academy of Sciences}], [\text{concluded after a two-year study}]. \]

\[ \quad \text{... wouldn’t it be great if we could just break it up?} \]
How have long inputs been handled previously?

- splitting on punctuation
How have long inputs been handled previously?

- splitting on punctuation:
  - supervised parsing of long Chinese sentences (Li et al., 2005)
  - (Li et al., 2010)
How have long inputs been handled previously?

- splitting on punctuation:
  - *supervised* parsing of long Chinese sentences (Li et al., 2005) (Li et al., 2010)
  - *unsupervised* constituent parsing (Ponvert et al., 2011)
How have long inputs been handled previously?

- **splitting on punctuation:**
  - *supervised* parsing of long Chinese sentences (Li et al., 2005)
    (Li et al., 2010)
  - *unsupervised* constituent parsing (Ponvert et al., 2011)
  - *unsupervised* chunking (Ponvert et al., 2010) via Seginer’s (2007) CCL parser
What if we chopped up input at punctuation?

- impact on quantity of data
What if we chopped up input at punctuation?

- **impact on quantity of data (with a 15-token threshold):**
  - number of training inputs goes up to 34,856 (from 15,922)
What if we chopped up input at punctuation?

- **Impact on quantity of data (with a 15-token threshold):**
  - Number of training inputs goes up to 34,856 (from 15,922)
  - Number of tokens increases to 709,215 (from 163,715)
What if we chopped up input at punctuation?

- impact on *quantity* of data (with a 15-token threshold):
  - number of training inputs goes up to 34,856 (from 15,922)
  - number of tokens increases to 709,215 (from 163,715)
  - more and simpler word sequences incorporated earlier
What if we chopped up input at punctuation?

- **impact on quantity of data (with a 15-token threshold):**
  - number of training inputs goes up to 34,856 (from 15,922)
  - number of tokens increases to 709,215 (from 163,715)
  - more and simpler word sequences incorporated earlier
  - much *more dense* coverage of available data
What if we chopped up input at punctuation?

- impact on *quantity* of data (with a 15-token threshold):
  - number of training inputs goes up to 34,856 (from 15,922)
  - number of tokens increases to 709,215 (from 163,715)
  - more and simpler word sequences incorporated earlier
  - much *more dense* coverage of available data

- but, also impact on *quality* of data
What if we chopped up input at punctuation?

- **impact on quantity of data (with a 15-token threshold):**
  - number of training inputs goes up to 34,856 (from 15,922)
  - number of tokens increases to 709,215 (from 163,715)
  - more and simpler word sequences incorporated earlier
  - much more dense coverage of available data

- **but, also impact on quality of data:**
  - mostly phrases and clauses (75% agree with constituent boundaries)
What if we chopped up input at punctuation?

- **impact on quantity of data (with a 15-token threshold):**
  - number of training inputs goes up to 34,856 (from 15,922)
  - number of tokens increases to 709,215 (from 163,715)
  - more and simpler word sequences incorporated earlier
  - much more dense coverage of available data

- **but, also impact on quality of data:**
  - mostly phrases and clauses (75% agree with constituent boundaries)
  - many fewer complete sentences exhibiting full structure
What if we chopped up input at punctuation?

- **impact on quantity of data (with a 15-token threshold):**
  - number of training inputs goes up to 34,856 (from 15,922)
  - number of tokens increases to 709,215 (from 163,715)
  - more and simpler word sequences incorporated earlier
  - much *more dense* coverage of available data

- **but, also impact on quality of data:**
  - mostly phrases and clauses  (75% agree with constituent boundaries)
  - many fewer complete sentences exhibiting full structure
  - *even less representative* than short sentences
What if we chopped up input at punctuation?

- **impact on quantity of data** (with a 15-token threshold):
  - number of training inputs goes up to 34,856 (from 15,922)
  - number of tokens increases to 709,215 (from 163,715)
  - more and simpler word sequences incorporated earlier
  - much *more dense* coverage of available data

- but, also impact on **quality of data**:
  - mostly phrases and clauses  (75% agree with constituent boundaries)
  - many fewer complete sentences exhibiting full structure
  - even *less representative* than short sentences

- however, there is an appropriate model family (DBMs)
Class-based, head-outward generation  

(Alshawi, 1996)
Class-based, head-outward generation

\[ ch \]
\[ ce \]
\[ dir = R \]
\[ adj = T \]
Class-based, head-outward generation

\[ \text{dir} = \mathbb{R} \quad \text{adj} = \mathbb{T} \]
Class-based, head-outward generation

\( \text{dir} = R \quad \text{adj} = T \)

(Alshawi, 1996)
Class-based, head-outward generation

(Alshawi, 1996)

\[ dir = R \]

\[ adj = F \]
Class-based, head-outward generation

\[ \text{dir} = R \]

\[ \text{adj} = F \]

(Alshawi, 1996)
Class-based, head-outward generation

\[
\begin{align*}
\text{dir} & = R \\
\text{adj} & = F
\end{align*}
\]

(Alshawi, 1996)
Class-based, head-outward generation

\[ \text{dir} = R \quad \text{adj} = F \]
Class-based, head-outward generation

\[ \text{dir} = R \]

\[ \text{adj} = F \]

(Alshawi, 1996)
Class-based, head-outward generation

\[ \text{dir} = R \]

\[ \text{adj} = F \]

\[ P_{\text{ROOT}}(c_h \mid \text{comp}) \]
Class-based, head-outward generation

\( \text{dir} = R \quad \text{adj} = F \)

\[ \mathbb{P}_{\text{ROOT}}(c_h \mid \text{comp}) \quad \mathbb{P}_{\text{ATTACH}}(c_d \mid c_h, \text{dir}, \text{cross}) \]
Class-based, head-outward generation (Alshawi, 1996)

\[
\begin{align*}
\mathbb{P}_{\text{ROOT}}(c_h \mid \text{comp}) \quad & \quad \mathbb{P}_{\text{ATTACH}}(c_d \mid c_h, \text{dir}, \text{cross}) \quad \mathbb{P}_{\text{STOP}}(\mid \text{dir}, \text{adj}, c_e, \text{comp})
\end{align*}
\]

\[ \text{dir} = R \quad \text{adj} = F \]
Example (cont’d): DBMs (Spitkovsky et al., 2012)
### Example (cont’d):

DBMs (Spitkovsky et al., 2012)

<table>
<thead>
<tr>
<th>complete</th>
<th>length &amp; type</th>
<th>left &amp; right</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>IN</td>
</tr>
<tr>
<td>51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>incomplete</td>
<td>SBAR</td>
<td>IN</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NP</td>
<td>NN</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PP</td>
<td>IN</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VP</td>
<td>VBZ</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NP</td>
<td>DT</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NP</td>
<td>DT</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VP</td>
<td>VBD</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example (cont’d):  

DBMs (Spitkovsky et al., 2012)

<table>
<thead>
<tr>
<th>Length &amp; Type</th>
<th>Left &amp; Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>S</td>
</tr>
<tr>
<td>12</td>
<td>SBAR</td>
</tr>
<tr>
<td>2</td>
<td>NP</td>
</tr>
<tr>
<td>6</td>
<td>PP</td>
</tr>
<tr>
<td>14</td>
<td>VP</td>
</tr>
<tr>
<td>4</td>
<td>NP</td>
</tr>
<tr>
<td>8</td>
<td>NP</td>
</tr>
<tr>
<td>5</td>
<td>VP</td>
</tr>
</tbody>
</table>

Incomplete Fragments / Austere Models

Spitkovsky et al. (Stanford & Google)
Example (cont’d):

DBMs (Spitkovsky et al., 2012)

<table>
<thead>
<tr>
<th></th>
<th>length &amp; type</th>
<th>left &amp; right</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>complete</strong></td>
<td>51 S</td>
<td>IN NN</td>
</tr>
<tr>
<td><strong>incomplete</strong></td>
<td>12 SBAR</td>
<td>IN NNS</td>
</tr>
<tr>
<td></td>
<td>2 NP</td>
<td>NN NN</td>
</tr>
<tr>
<td></td>
<td>6 PP</td>
<td>IN NNS</td>
</tr>
<tr>
<td></td>
<td>14 VP</td>
<td>VBZ NNS</td>
</tr>
<tr>
<td></td>
<td>4 NP</td>
<td>DT NNP</td>
</tr>
<tr>
<td></td>
<td>8 NP</td>
<td>DT NNPS</td>
</tr>
<tr>
<td></td>
<td>5 VP</td>
<td>VBD NN</td>
</tr>
</tbody>
</table>
**Example (cont’d):**

DBMs (Spitkovsky et al., 2012)

<table>
<thead>
<tr>
<th></th>
<th>length &amp; type</th>
<th>left &amp; right</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>complete</strong></td>
<td>51</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>NN</td>
</tr>
<tr>
<td><strong>incomplete</strong></td>
<td>12</td>
<td>SBAR</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>NNS</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>NN</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>PP</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>NNS</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>VP</td>
</tr>
<tr>
<td></td>
<td>VBZ</td>
<td>NNS</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>NNP</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>NNPS</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>VP</td>
</tr>
<tr>
<td></td>
<td>VBD</td>
<td>NN</td>
</tr>
</tbody>
</table>

DBM-3

**partial parse forests**

“easy-first” (Goldberg and Elhadad, 2010), optional soft EM
We tried; it works...

\(^1\text{nlp.stanford.edu/pubs/goldtags-data.tar.bz2: untagger model}\)
We tried; it works...

- experimental setup

\[1\text{nlp.stanford.edu/pubs/goldtags-data.tar.bz2:untagger.model}\]
We tried; it works...

- experimental setup:
  - context-sensitive unsupervised tags (no gold POS)\(^1\)

\(^1\)nlp.stanford.edu/pubs/goldtags-data.tar.bz2:unagger.model
We tried; it works...

- experimental setup:
  - context-sensitive unsupervised tags (no gold POS)\(^1\)
  - performance metric is directed dependency accuracy

\(^1\)nlp.stanford.edu/pubs/goldtags-data.tar.bz2:unagger.model
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)$^1$
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

---

$^1$nlp.stanford.edu/pubs/goldtags-data.tar.bz2:unagger.model
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)\(^1\)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline:** 59.7%

\(^1\)nlp.stanford.edu/pubs/goldtags-data.tar.bz2: untagger.model
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)\(^1\)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline:** 59.7%

  (Spitkovsky et al., 2011) 59.1 [EMNLP]  context-sensitive clusters
  (Spitkovsky et al., 2011) 58.4 [CoNLL]  punctuation constraints
  (Tu and Honavar, 2012) 57.0 [EMNLP-CoNLL]
  (Blunsom and Cohn, 2011) 55.7 [EMNLP]
  (Gillenwater et al., 2010) 53.3 [TechReport]
  (Bisk and Hockenmaier, 2012) 53.3 [AAAI]
  (Spitkovsky et al., 2010) 47.9 [CoNLL]  Viterbi training

\(^1\)nlp.stanford.edu/pubs/goldtags-data.tar.bz2:untagger.model
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)\(^1\)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline:** 59.7%
  - DBMs on whole inputs only

\(^1\) nlp.stanford.edu/pubs/goldtags-data.tar.bz2: untagger.model
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)\(^1\)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline:** 59.7%
  - DBMs on whole inputs only
  - staged training on WSJ15 → WSJ45

\(^1\)nlp.stanford.edu/pubs/goldtags-data.tar.bz2:unagger.model
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)$^1$
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline: 59.7%**
  - DBMs on whole inputs only
  - staged training on WSJ15 $\rightarrow$ WSJ45
  - strong punctuation-induced constraints for full data

---

$^1$nlp.stanford.edu/pubs/goldtags-data.tar.bz2: untagger.model
We tried; it works...

**experimental setup:**
- context-sensitive unsupervised tags (no gold POS)
- performance metric is directed dependency accuracy
- evaluation on Section 23 (all sentences)

**state-of-the-art baseline: 59.7%**
- DBMs on whole inputs only
- staged training on WSJ15 → WSJ45
- strong punctuation-induced constraints for full data
- weaker constraints used in decoding for evaluation

---

1nlp.stanford.edu/pubs/goldtags-data.tar.bz2: untagger model
We tried; it works...

- experimental setup:
  - context-sensitive unsupervised tags (no gold POS)\(^1\)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- state-of-the-art baseline: 59.7%
  - DBMs on whole inputs only
  - staged training on WSJ15 $\rightarrow$ WSJ45
  - strong punctuation-induced constraints for full data
  - weaker constraints used in decoding for evaluation

- results with initially-split data — 60.2%

\(^{1}\text{nlp.stanford.edu/pubs/goldtags-data.tar.bz2:untagger.model}\)
We tried; it works…

- experimental setup:
  - context-sensitive unsupervised tags (no gold POS)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- state-of-the-art baseline: 59.7%
  - DBMs on whole inputs only
  - staged training on WSJ15 → WSJ45
  - strong punctuation-induced constraints for full data
  - weaker constraints used in decoding for evaluation

- results with initially-split data — 60.2%
  - can do better with simpler initial models — 61.2%
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline:** 59.7%
  - DBMs on whole inputs only
  - staged training on WSJ15 → WSJ45
  - strong punctuation-induced constraints for full data
  - weaker constraints used in decoding for evaluation

- **results with initially-split data** — 60.2% (3.5% exact)
  - can do better with simpler initial models — 61.2% (5.0%)

\[nlp.stanford.edu/pubs/goldtags-data.tar.bz2:unagger.model\]
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)\(^1\)
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline:** 59.7%
  - DBMs on whole inputs only
  - staged training on WSJ15 $\rightarrow$ WSJ45
  - strong punctuation-induced constraints for full data
  - weaker constraints used in decoding for evaluation

- **results with initially-split data** — 60.2% (3.5% exact)
  - can do better with simpler initial models — 61.2% (5.0%)
  - e.g., better not to model roots of incomplete fragments

\(^1\)nlp.stanford.edu/pubs/goldtags-data.tar.bz2:untagger.model
We tried; it works...

- **experimental setup:**
  - context-sensitive unsupervised tags (no gold POS)\textsuperscript{1}
  - performance metric is directed dependency accuracy
  - evaluation on Section 23 (all sentences)

- **state-of-the-art baseline:** 59.7%
  - DBMs on whole inputs only
  - staged training on WSJ15 → WSJ45
  - strong punctuation-induced constraints for full data
  - weaker constraints used in decoding for evaluation

- **results with initially-split data** — 60.2% (3.5% exact)
  - can do better with simpler initial models — 61.2% (5.0%)
  - e.g., better not to model roots of incomplete fragments
  - ... as well as non-adjacency for short inputs

\textsuperscript{1}nlp.stanford.edu/pubs/goldtags-data.tar.bz2:untagger.model
Summary

- instead of bootstrapping dependency grammar inducers from 16K short whole sentences (160K tokens)
Summary

- instead of bootstrapping dependency grammar inducers from 16K short whole sentences (160K tokens), we
  - start with 35K inter-punctuation fragments (709K tokens)
Summary

- instead of bootstrapping dependency grammar inducers from 16K short whole sentences (160K tokens), we
  - start with 35K inter-punctuation fragments (709K tokens)
  - using appropriate models that can handle incomplete data
instead of bootstrapping dependency grammar inducers from 16K short whole sentences (160K tokens), we

- start with 35K inter-punctuation fragments (709K tokens)
- using appropriate models that can handle incomplete data
- and improved state-of-the-art accuracy by more than 2%
Possible future directions?
Possible future directions?

- could we induce grammars from ungrammatical inputs?
Possible future directions?

- could we induce grammars from ungrammatical inputs?
  - perhaps sentence prefixes and suffixes?
Possible future directions?

- could we induce grammars from ungrammatical inputs?
  - perhaps sentence prefixes and suffixes?
  - could we go all the way down to $n$-grams?
Thanks!

Any questions?