Using Universal Linguistic Knowledge to Guide Grammar Induction

[Naseem et al., 2010]

Juri Alexander Opitz

June 30, 2016
“By a generative grammar I mean simply a system of rules that in some explicit and well-defined way assigns structural descriptions to sentences. Obviously, every speaker of a language has mastered and internalized a generative grammar (...) This is not to say that he is aware of the rules of the grammar or even that he can become aware of them.”

Overview

Introduction

The Model

Experiments

Conclusions

Outlook
Introduction
What Naseem et al. seek to accomplish

Guide (Dependency-) Grammar induction by (known) Linguistic Universals.
What is Grammar Induction?

- Automatic Learning of a Formal Grammar

1. receive observations
2. construct model which “explains” the observations
Why do we need Grammar Induction in NLP?

- Observations: spoken/written natural language
- Model: any kind of model which explains how the observations arised (by incorporating underlying deeper structures).
Example: Practical Use

- Observations: Texts (+Trees in supervised case).
- Model: Parser.
- Goal: Parse new Texts.
Why Grammar Induction for LRLs?

Successful parsers rely on manually annotated training material, which is:

- very costly (especially in this case: human needs to annotate data with trees)...
- typically constructed for each language.
Why Grammar Induction for LRLs?

Hence we need *Unsupervised* Grammar Induction for LLRs.
Common Problem with Unsupervised Learning

Models perform usually much worse than their supervised counterparts: They have no teacher and must learn on their own :-(

A possible Cure

Principal Idea of the paper: Exploit universal knowledge to *guide* the learning process.
## Linguistic Universals

| Root → Auxiliary | Noun → Adjective |
| Root → Verb     | Noun → Article   |
| Verb → Noun     | Noun → Noun      |
| Verb → Pronoun  | Noun → Numeral   |
| Verb → Adverb   | Preposition → Noun |
| Verb → Verb     | Adjective → Adverb |
| Auxiliary → Verb |                 |
Linguistic Universals - Example Parse

| Root → Auxiliary | Noun → Adjective |
| Root → Verb | Noun → Article |
| Verb → Noun | Noun → Noun |
| Verb → Pronoun | Noun → Numeral |
| Verb → Adverb | Preposition → Noun |
| Verb → Verb | Adjective → Adverb |
| Auxiliary → Verb |

Sentence: Nim Chimsky eats a ripe banana.

Noun Noun Verb Article Adjective Noun
Using Universal Linguistic Knowledge to Guide Grammar Induction

Introduction

Linguistic Universals - Example Parse

<table>
<thead>
<tr>
<th>Root → Auxiliary</th>
<th>Noun → Adjective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root → Verb</td>
<td>Noun → Article</td>
</tr>
<tr>
<td>Verb → Noun</td>
<td>Noun → Noun</td>
</tr>
<tr>
<td>Verb → Pronoun</td>
<td>Noun → Numerical</td>
</tr>
<tr>
<td>Verb → Adverb</td>
<td>Preposition → Noun</td>
</tr>
<tr>
<td>Verb → Verb</td>
<td>Adjective → Adverb</td>
</tr>
<tr>
<td>Auxiliary → Verb</td>
<td></td>
</tr>
</tbody>
</table>

Sentence: Nim Chimsky eats a ripe banana.

Noun Noun Verb Article Adjective Noun

a
|
banana--
|
root--eats-- ripe
|
Nim--Chimsky
Grammar induction & Low Resource Languages (LRLs)

Idea: With linguistic Universals we can guide grammar induction when we have few or no annotated data at all.
The Model, “explaining what we observe”.
Naseem et al. use a generative Bayesian Model to describe grammar generation when we observe words $x_1, x_2, ..., x_n$ and corresponding coarse symbols, i.e. PoS-Tags $s_1, s_2, ..., s_n$. 
Naseem et al. use hidden, refined symbols $z_1, z_2, ..., z_n$. For simplicity, we drop this here, i.e. $z_1, z_2, ..., z_n = s_1, s_2, ..., s_n$. 
Simplified Model: 2 Facets

1. Generative Process for Model parameters
2. Generative Process for Parses
Simplified Model: 2 Facets

1. For each coarse symbol $s$:
   - Draw a *word generation multinomial*.
   - For each possible context value $c$, draw also a *child symbol generation multinomial*.

2. For each Tree Node $i$ generated in context $c$ by parent symbol $s'$:
   - Draw coarse symbol $s_i$ from *child symbol generation multinomial* of parent
   - Draw word $x_i$ from *word generation multinomial*.
More formally:

1. For each coarse symbol $s$:
   - Draw $\Phi_s \sim Dir(\Phi_0)$.
   - For each possible context value $c$, draw $\theta_{sc} \sim Dir(\theta_0)$

2. For each Tree Node $i$ generated in context $c$ by parent symbol $s'$:
   - Draw coarse symbol $s_i \sim Mult(\theta_{s'})$
   - Draw word $x_i \sim Mult(\Phi_{s_i})$. 
The Dirichlet Distribution...

... is a distribution over multinomial distributions...
2 Parameters: K

K: How many discrete events do we have (e.g. number of words in vocab).
2 Parameters: Vector \( \mathbf{\alpha} \)

A K-dimensional “concentration parameter” Vector, all \( \alpha_i \) must be > 0 (e.g. counts of each word in text).
Example for K=3
Example for K=3

\[ \alpha = (6, 2, 2), (3, 7, 5), (6, 2, 6), (2, 3, 4), \text{ clockwise from top left} \]
Model: Plate Outline

- \( s \) - coarse symbol (observed)
- \( z \) - refined subsymbol
- \( x \) - word (observed)
- \( \theta_{szc} \) - distr over child coarse symbols for each parent \( s \) and \( z \) and context \( c \)
- \( \beta_s \) - top-level distr over subsymbols for \( s \)
- \( \pi_{ss'z'c} \) - distr over subsymbols for each \( s \), parent \( s' \) and \( z' \), and context \( c \)
- \( \phi_{sz} \) - distr over words for \( s \) and \( z \)
Inference with Constraints

Idea: constrain the posterior to satisfy the rules in expectation during inference.

▶ What? we require that a certain percentage of linguistic universals must occur in the model expectations.

▶ Why? Biases the model-inference towards linguistically more plausible setting.

▶ Advantage: we require only a certain percentage of linguistic universals to hold $\rightarrow$ percentage can be tuned for every language.
Inference with Constraints

Method outline:

▶ **Maximize lower bound on likelihood of observations**
  (equivalent to minimizing Divergence between the true posterior distribution of model parameters and other distributions of model parameters!)

▶ **Implement constraints in constrained optimization problem:**
  ▶ a certain % of universals must hold!
Experiments
Experimental Setup

Languages: English, Danish, Portuguese, Slovene, Spanish, and Swedish
Experiments: Setup

Languages: English, Danish, Portuguese, Slovene, Spanish, and Swedish.

- English data: dependency modification of Penn Treebank [Taylor et al., 2003], sentence-length < 20.
- Other data: 2006 CoNLL-X Shared task [Buchholz and Marsi, 2006], sentence-length < 10.
- each data set provides manually annotated PoS-tags.
Experiments: Setup

Metric: Dependency Accuracy.

- Percentage of words having the correct head.
## Experiments: Results

<table>
<thead>
<tr>
<th>Language</th>
<th>DMV</th>
<th>PGI</th>
<th>No-Split</th>
<th>HDP-DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>47.1</td>
<td>62.3</td>
<td>71.5</td>
<td>71.9 (0.3)</td>
</tr>
<tr>
<td>Danish</td>
<td>33.5</td>
<td>41.6</td>
<td>48.8</td>
<td>51.9 (1.6)</td>
</tr>
<tr>
<td>Portuguese</td>
<td>38.5</td>
<td>63.0</td>
<td>54.0</td>
<td>71.5 (0.5)</td>
</tr>
<tr>
<td>Slovene</td>
<td>38.5</td>
<td>48.4</td>
<td>50.6</td>
<td>50.9 (5.5)</td>
</tr>
<tr>
<td>Spanish</td>
<td>28.0</td>
<td>58.4</td>
<td>64.8</td>
<td>67.2 (0.4)</td>
</tr>
<tr>
<td>Swedish</td>
<td>45.3</td>
<td>58.3</td>
<td>63.3</td>
<td>62.1 (0.5)</td>
</tr>
</tbody>
</table>

DMV, PGI: Baselines.
No-split: This model without refined subsymbols.
HDP_DEP: This model.
Experiments: Ablations

What happens when we exclude certain universal rules?
### Experiments: Ablations

#### English

<table>
<thead>
<tr>
<th>Rule Excluded</th>
<th>Acc</th>
<th>Loss</th>
<th>Gold Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preposition → Noun</td>
<td>61.0</td>
<td>10.9</td>
<td>5.1</td>
</tr>
<tr>
<td>Verb → Noun</td>
<td>61.4</td>
<td>10.5</td>
<td>14.8</td>
</tr>
<tr>
<td>Noun → Noun</td>
<td>64.4</td>
<td>7.5</td>
<td>10.7</td>
</tr>
<tr>
<td>Noun → Article</td>
<td>64.7</td>
<td>7.2</td>
<td>8.5</td>
</tr>
</tbody>
</table>

#### Spanish

<table>
<thead>
<tr>
<th>Rule Excluded</th>
<th>Acc</th>
<th>Loss</th>
<th>Gold Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preposition → Noun</td>
<td>53.4</td>
<td>13.8</td>
<td>8.2</td>
</tr>
<tr>
<td>Verb → Noun</td>
<td>61.9</td>
<td>5.4</td>
<td>12.9</td>
</tr>
<tr>
<td>Noun → Noun</td>
<td>62.6</td>
<td>4.7</td>
<td>2.0</td>
</tr>
<tr>
<td>Root → Verb</td>
<td>65.4</td>
<td>1.8</td>
<td>12.3</td>
</tr>
</tbody>
</table>
Experiments: Constraints Thresholds

What happens when we increase/decrease the percentage of dependencies which must be in accordance with the universals?
Experiments: Constraints Thresholds

- **Average**
- **English**

![Bar Chart](chart.png)

- **Accuracy**
- **Constraints Threshold**

- **Gold**
- **70**
- **75**
- **80**
- **85**
- **90**
Experiments: Constraints Thresholds

Accuracy

Gold  70  75  80  85  90

Constraints Threshold

Average  English
Conclusions
Conclusions

- it is good to have only a percentage, accuracy is stable between 75% and 90%.
- a value of 80% seems to perform well across languages.
- Setting the value to the true proportion (for all languages $\leq 70\%$) in the gold labellings does not increase performance.
- English performs best.
Experiments: Sentence Lengths, Universal Rules

<table>
<thead>
<tr>
<th></th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>≤ 10</td>
</tr>
<tr>
<td></td>
<td>≤ 20</td>
</tr>
</tbody>
</table>

**Universal Dependency Rules**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HDP-DEP</td>
<td>71.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50.4</td>
</tr>
</tbody>
</table>

**No Rules (Random Init)**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>HDP-DEP</td>
<td>24.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24.4</td>
</tr>
<tr>
<td>3</td>
<td>Headden III et al. (2009)</td>
<td>68.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>
Experiments: Sentence Lengths, English Specific Rules

<table>
<thead>
<tr>
<th>English-Specific Parsing Rules</th>
<th>Deterministic (rules only)</th>
<th>70.0</th>
<th>62.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>HDP-DEP</td>
<td>73.8</td>
<td>66.1</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Druck et al. (2009) Rules</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Druck et al. (2009)</td>
<td>61.3</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>HDP-DEP</td>
<td>64.9</td>
<td>42.2</td>
</tr>
</tbody>
</table>
Conclusions

- Longer sentences are more difficult to parse.
- Using no universal rules at all results in “desastrous” performance.
- With additional language-specific rules, performance increases by almost 2%.
Outlook
Another Approach to LR Dependency Parsing

Grammar Induction from Text Using Small Syntactic Prototypes.
[Boonkwan and Steedman, 2011]
Another Approach to LR Dependency Parsing

[Boonkwan and Steedman, 2011] about [Naseem et al., 2010]:

▶ “method still needs language specific rules to boost accuracy”
Another Approach to LR Dependency Parsing

Idea: Use Categorial Grammar rules as prototypes.
Example

Words are from atomic categories or they are functors from categories to categories.
Example

John, sandwiches \(\vdash np\)
delicious \(\vdash np/\rightarrow np\)
eats \(\vdash s\rightarrow np/<np\)

\(<, >: as “head right - left child, head left-right child”

/: application from right
\/: application from left
Example: Derivation Rules

John, sandwiches ⊢ np

delicious ⊢ np |> np

eats ⊢ s |> np <| np

\[ \begin{align*}
X/Y : d_1 & \quad Y : d_2 & \Rightarrow & \quad X : h(d_1) \rightarrow h(d_2) \\
X/Y : d_1 & \quad Y : d_2 & \Rightarrow & \quad X : h(d_1) \leftarrow h(d_2) \\
Y : d_1 & \quad X/Y : d_2 & \Rightarrow & \quad X : h(d_1) \rightarrow h(d_2) \\
Y : d_1 & \quad X/Y : d_2 & \Rightarrow & \quad X : h(d_1) \leftarrow h(d_2)
\end{align*} \]
Anyone wants to derive “John eats a delicious sandwich”?
Using Universal Linguistic Knowledge to Guide Grammar Induction

John eats delicious sandwiches

$np \xrightarrow{s
op/np} \frac{np}{np} \xrightarrow{np/np} np \xrightarrow{np} np \xrightarrow{s
op/np} s$
Language Parametrization

Ask non-linguist native-speaker about word orders (e.g. subj-verb-obj), derive rules from that.
They manage to improve over Naseem et al. 1. without language specific rules and (+ 3% F1) 2. with language specific rules (+ 1% F1).
Comparison of Grammar Induction Approaches

Performance:
- [Boonkwan and Steedman, 2011] approach wins.

Abstraction, Universality:
- Naseem et al. rely on only a small set of universal rules
- Approach from [Boonkwan and Steedman, 2011] needs work of a native speaker for each language to be parsed.
- [Naseem et al., 2010] approach seems more universal (to me).
Thank you for listening.

Literatur II
