Inspecting the Structural Biases of Dependency Parsing Algorithms

Yoav Goldberg and Michael Elhadad

Ben Gurion University

CoNLL 2010, Sweden
There are many ways to parse a sentence
There are many ways to parse a sentence

**Transition Based Parsers**
- Covington
- Multiple Passes
- Arc-Eager
- Arc-Standard
- With Swap Operator
- First-Best Parser
- DAG Parsing
- With a Beam
- With Dynamic Programming
- With Tree Revision
- Left-to-right
- Right-to-left

**Easy-First Parsing** (check out our naacl 2010 paper)

**Graph Based Parsers**
- First Order
- Second Order (two children / with grandparent)
- Third Order
- MST Algorithm / Matrix Tree Theorem
- Eisner Algorithm
- Belief Propagation
- With global constraints (ILP / gibbs sampling)

**Combinations**
- Voted Ensembles (Sagae’s way, Attardi’s way)
- Stacked Learning
We can build many reasonably accurate parsers
We can build many reasonably accurate parsers

Parser combinations work
We can build many reasonably accurate parsers

Parser combinations work
  ⇒ every parser has its strong points
We can build many reasonably accurate parsers

Parser combinations work
  ⇒ every parser has its strong points

Different parsers behave differently
Open questions
Open questions

**WHY** do they behave as they do?
Open questions

**WHY** do they behave as they do?

**WHAT** are the differences between them?
More open questions

Which linguistic phenomena are hard for parser X?
More open questions

Which linguistic phenomena are hard for parser X?

What kinds of errors are common for parser Y?
More open questions

Which linguistic phenomena are hard for parser X?

What kinds of errors are common for parser Y?

Which parsing approach is most suitable for language Z?
Performance

We are around here

Understanding

I want to be there
Some people would settle for here

We are around here

I want to be there
Some people would settle for here

But they probably would benefit from bypassing this wall.

We are around here

I want to be there
Some people would settle for here, but they probably would benefit from bypassing this wall. We are around here. This work brings us here. I want to be there.
Previously

McDonald and Nivre 2007:

“Characterize the Errors of Data-Driven Dependency Parsing Models”
Previously

McDonald and Nivre 2007:

“Characterize the Errors of Data-Driven Dependency Parsing Models”

- Focus on **single-edge** errors
Previously

McDonald and Nivre 2007:

“Characterize the Errors of Data-Driven Dependency Parsing Models”

- Focus on single-edge errors
  - MST better for long edges, MALT better for short
  - MST better near root, MALT better away from root
  - MALT better at nouns and pronouns, MST better at others

... but all these differences are very small
Previously

McDonald and Nivre 2007:

“Characterize the Errors of Data-Driven Dependency Parsing Models”

- Focus on **single-edge** errors
  - MST better for long edges, MALT better for short
  - MST better near root, MALT better away from root
  - MALT better at nouns and pronouns, MST better at others

- …but all these differences are very small
we do something a bit different
Assumptions

- Parsers fail in predictable ways
- those can be analyzed
- analysis should be done by inspecting trends rather than individual decisions
Note: We do not do error analysis
Note: We do not do error analysis

- Error analysis is *complicated*
  - one error can yield another / hide another
Note: We do not do error analysis

- Error analysis is **complicated**
  - one error can yield another / hide another

- Error analysis is **local** to one tree
  - many factors may be involved in that single error
Note: We do not do error analysis

- Error analysis is **complicated**
  - one error can yield another / hide another

- Error analysis is **local** to one tree
  - many factors may be involved in that single error

we are aiming at more global trends
Structural Preferences
Structural preferences

for a given language+syntactic theory

- Some structures are more common than others
  - (think Right Branching for English)
Structural preferences

for a given language+syntactic theory

- Some structures are more common than others
  - (think Right Branching for English)

- Some structures are very rare
  - (think non-projectivity, OSV constituent order)
Structural preferences

parsers also exhibit structural preferences
Structural preferences

parsers also exhibit structural preferences

▶ Some are explicit / by design
  ▶ e.g. projectivity
Structural preferences

parsers also exhibit structural preferences

- Some are explicit / by design
  - e.g. projectivity

- Some are implicit, stem from
  - features
  - modeling
  - data
  - interactions
  - and other stuff

These trends are interesting!
Structural preferences

parsers also exhibit structural preferences

- Some are explicit / by design
  - e.g. projectivity

- Some are implicit, stem from
  - features
  - modeling
  - data
  - interactions
  - and other stuff

These trends are interesting!
Structural Bias
Structural bias

“The difference between the structural preferences of two languages”
Structural bias

“The difference between the structural preferences of two languages”

For us:

*Which structures tend to occur more in language than in parser?*
Bias vs. Error

related, but not the same

*Parser X makes many PP attachment errors*

▶ claim about error pattern
Bias vs. Error

related, but not the same

*Parser X makes many PP attachment errors*
  ▶ claim about error pattern

*Parser X tends to attach PPs low, while language Y tends to attach them high*
  ▶ claim about structural bias (and also about errors)
Bias vs. Error

related, but not the same

*Parser X makes many PP attachment errors*
  - claim about error pattern

*Parser X tends to attach PPs low, while language Y tends to attach them high*
  - claim about structural bias (and also about errors)

*Parser X can never produce structure Y*
  - claim about structural bias
Formulating Structural Bias

“given a tree, can we say where it came from?”
Formulating Structural Bias

“given two trees of the same sentence, can we tell which parser produced each parse?”

?
Formulating Structural Bias

“which parser produced which tree?”

any predictor that can help us answer this question is an indicator of structural bias
Formulating Structural Bias

“which parser produced which tree?”

any predictor that can help us answer this question is an indicator of structural bias
Formulating Structural Bias

“which parser produced which tree?”

any predictor that can help us answer this question is an indicator of structural bias

uncovering structural bias = searching for good predictors
Method

- start with two sets of parses for same set of sentences
- look for predictors that allow to distinguish between trees in each group
Our Predictors

- all possible subtrees
Our Predictors

- all possible subtrees
- always encode:
  - parts of speech
  - relations
  - direction

```
JJ
NN  VB  IN
```
Our Predictors

- all possible subtrees
- always encode:
  - parts of speech
  - relations
  - direction
- can encode also:
  - lexical items
Our Predictors

- all possible subtrees
- always encode:
  - parts of speech
  - relations
  - direction
- can encode also:
  - lexical items
  - distance to parent
Search Procedure

boosting with subtree features

very briefly:
Search Procedure

boosting with subtree features

very briefly:

- **input: two sets of constituency trees**
- **while not done:**
  - choose a subtree that classifies most trees correctly
  - re-weight trees based on errors
boosting with subtree features

very briefly:

- **input:** two sets of constituency trees
- **while not done:**
  - choose a subtree that classifies most trees correctly
  - re-weight trees based on errors
- **output:** weighted subtrees (= linear classifier)
Setup

Gold trees
Parsed trees
train
Validation
Setup

Gold Trees

Parsed Trees

Train

Validation

k&m 2004

Weighted Subtrees

= Classifier
Setup

Gold trees  
Parsed trees

train  
validation

KDM 2004

Weighted Subtrees = Classifier
Setup

Gold trees
Parsed trees

train

validation

KSM
2004

Weighted
Subtrees =
Classifier

ignore
Weights

Subtrees

Rescore
(count based)

Bias Predictors
Setup

- Represent as constituents

KAM 200h

→ Weighted Subtrees = Classifier

→ Gold Trees
→ Parsed Trees

Train

→ Validation

→ Ignore Weights

→ Subtrees

→ Restore (Count based)

- Bias Predictors
conversion to constituency

mandatory information at node label
optional information as leaves
conversion to constituency

mandatory information at node label
optional information as leaves
conversion to constituency

mandatory information at node label
optional information as leaves
Experiments

Analyzed Parsers

- Malt Eager
- Malt Standard
- Mst 1
- Mst 2
Experiments

Analyzed Parsers

- Malt Eager
- Malt Standard
- Mst 1
- Mst 2

Data

- WSJ (converted using Johansson and Nugues)
- splits: parse-train (15-18), boost-train (10-11), boost-val (4-7)
- gold pos-tags
Q: Are the parsers biased with respect to English?
Q: Are the parsers biased with respect to English?
A: Yes
Quantitative Results

Q: Are the parsers biased with respect to English?  
A: Yes

<table>
<thead>
<tr>
<th>Parser</th>
<th>Train Accuracy</th>
<th>Val Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MST1</td>
<td>65.4</td>
<td>57.8</td>
</tr>
<tr>
<td>MST2</td>
<td>62.8</td>
<td>56.6</td>
</tr>
<tr>
<td>MALTE</td>
<td>69.2</td>
<td>65.3</td>
</tr>
<tr>
<td>MALTS</td>
<td>65.1</td>
<td>60.1</td>
</tr>
</tbody>
</table>

Table: Distinguishing parser output from gold-trees based on structural information
Qualitative Results (teasers)

Over-produced by ArcEager:
Qualitative Results (teasers)

Over-produced by ArcEager:

\[ \text{ROOT} \rightarrow \text{“} \quad \text{ROOT} \rightarrow \text{DT} \quad \text{ROOT} \rightarrow \text{WP} \]

(we knew it’s bad at root, now we know how!)
Qualitative Results (teasers)

Over-produced by ArcEager:

\[
\text{ROOT} \rightarrow \text{“} \quad \text{ROOT} \rightarrow \text{DT} \quad \text{ROOT} \rightarrow \text{WP}
\]

\[
\text{ROOT VBD} \quad \text{VBD}
\]

(we knew it’s bad at root, now we know how!)
Qualitative Results (teasers)

Over-produced by ArcEager and ArcStandard
Qualitative Results (teasers)

Over-produced by ArcEager and ArcStandard

\[ \rightarrow VBD \overset{9}{\rightarrow} VBD \]

\[ \rightarrow VBD \overset{5}{\rightarrow} VBD \]

ROOT \[ \rightarrow VBZ \rightarrow VBZ \]

(prefer first verb above second one: because of left-to-right processing?)
Qualitative Results (teasers)

Over-produced by MST1
Qualitative Results (teasers)

Over-produced by MST1

(Independence assumption failing)
Qualitative Results (teasers)

Under-produced by MST1 and MST2
Qualitative Results (teasers)

Under-produced by MST1 and MST2

→ NN IN CC NN

(hard time in coordinating “heavy” NPs: due to pos-in-between feature?)
Qualitative Results (teasers)

More in paper

You should read it
Qualitative Results (teasers)

Software available

Try with your language / parser
To Conclude

- understanding HOW parsers behave and WHY is important
  - we should do more of that

- we defined structural bias as way of characterizing behaviour

- we presented an algorithm for uncovering structural bias

- applied to English with interesting results