Deriving Multi-Headed Planar Dependency Parses from Link Grammar Parses

Juneki Hong, Jason Eisner

January 28, 2015
Introduction

- This talk is about automatically converting from one annotation style to another.
- The conversion could be hard, where information is fragmented or missing.
- We use a general technique, Integer Linear Programming to recover this missing information.
In Our Case: What We Started With

- n-u
  v
  e
  e
  v
  v-d
  r
  n-u

the matter may never even be tried in court.

Link Grammar: Parse with undirected edges
What We Wanted:

- n-u  v  e  e  v  v-d  r  n-u  -
the matter may never even be tried in court.

Multiheaded parse with directionalized edges
Why We Wanted That

- We want to develop parsing algorithms for parses that look like this
- We couldn’t figure out where to get the data to test them.
Single-headedness

- Dependency parse treebanks today are either single-headed or not planar.
- Stanford Dependencies are multiheaded but not planar

Some example dependency parse.
Single-headedness

- Dependency parse treebanks today are either single-headed or not planar.
- Stanford Dependencies are multiheaded but not planar.

Some example dependency parse.

Link Grammar is almost a multiheaded planar corpora! We just need to directionalize the links.
Why Multi-headedness?

Multi-headedness Can Capture Additional Linguistic Phenomenon

- Control
- Relativization
- Conjunction
Control

Jill is the subject of two verbs

Jill persuaded Jack to skip

Jack is the object of one verb and the subject of another
The boy is the object of *with* as well as the subject of *fell.*
Conjunction

Jack and Jill serve as the two arguments to *and*, but are also subjects of *went*.
Motivation

- A multiheaded dependency corpus would be useful for testing new parsing algorithms
Motivation

- A multiheaded dependency corpus would be useful for testing new parsing algorithms
- Such a corpus could be automatically annotated using Integer Linear Programming
Motivation

- A multiheaded dependency corpus would be useful for testing new parsing algorithms.
- Such a corpus could be automatically annotated using Integer Linear Programming.
- We explored whether the Link Grammar could be adapted for this purpose.
Motivation

- A multiheaded dependency corpus would be useful for testing new parsing algorithms
- Such a corpus could be automatically annotated using Integer Linear Programming
- We explored whether the Link Grammar could be adapted for this purpose.
- The results of this are mixed, but provides a good case study.
Link Grammars

Grammar-based formalism for projective dependency parsing with undirected links

Original formalism and English Link Grammar created by Davy Temperley, Daniel Sleator, and John Lafferty (1991)
# Link Grammars: How They Work

These figures were clipped from the original Link Grammar paper: “Parsing English with a Link Grammar” by Sleator and Temperley

\(^1\)These figures were clipped from the original Link Grammar paper: “Parsing English with a Link Grammar” by Sleator and Temperley
# Link Grammars: How They Work

![Diagram of Link Grammars](image-url)

- the
- cat
- chased
- a
- snake
Link Grammars: How They Work
Link Grammars: Same Example Parse From Before Again

the matter may never even be tried in court.

Link Parse of a sentence from Penn Tree Bank
Link Grammars: Converting into a Directed Acyclic Graph

Directionalize the edges

-the matter may never even be tried in court.
Link Grammars

Compare resulting dependency parse with CoNLL 2007 shared task.

Bottom half is CoNLL. Top half is the directionalized link parse.
Link Grammars

Compare resulting dependency parse with CoNLL 2007 shared task.

Bottom half is CoNLL. Top half is the directionalized link parse.
What is Integer Linear Programming?

- An optimization problem where some or all of the variables are integers.
- The objective function and constraints are linear.
- In general, it’s NP-Hard! But good solvers exist that work well most of the time.
- Our ILP is encoded as a ZIMPL program and solved using the SCIP Optimization Suite\(^2\)

\(^2\)http://scip.zib.de/
Corpus Building Strategy

- We start with a corpus of link grammar parses.
- We try to directionalize this corpus, choosing a direction for every link in every parse.
- We use an Integer Linear Program to find consistent directions for every link.
### Integer Linear Programming Model

Encoded Constraints:

- Acyclicity
- Connectedness
- Consistency of Directionalized Links
Integer Linear Programming Model

Encoded Constraints:

- Acyclicity: No cycles!
- Connectedness: Every word is reachable from a root
- Consistency of Directionalized Links: Similar links oriented the same way
**Integer Linear Programming Model**

For each sentence, for each edge $i, j$, where $i < j$

Variables:

$x_{ij}, x_{ji} \in \mathbb{Z} \geq 0$: orientation of each link

$x_{ij} + x_{ji} = 1$
Integer Linear Programming Model

For each sentence, for each edge $i, j$, where $i < j$

\[
\begin{array}{c}
\text{L} \\
\text{i} \\
\text{j}
\end{array}
\]

\[
\ldots i \ldots j \ldots
\]

Variables:

\[
\begin{align*}
x_{ij}, x_{ji} &\in \mathbb{Z} \geq 0: \text{orientation of each link} \\
x_{ij} + x_{ji} &= 1
\end{align*}
\]

A link can either be oriented left or oriented right.
Acyclicity, Connectedness

Acyclicity
Given that node $u$ is the parent of $v$

$n_v$: length of the sentence containing node $v$

$d_v \in [0, n_v]$: depth of the node from the root of the sentence

\[
(\forall u) \ d_v + (1 + n_v) \cdot (1 - x_{uv}) \geq 1 + d_u
\]  

(1)

Connectedness

\[
\sum_u x_{uv} \geq 1
\]  

(2)
Acyclicity, Connectedness

Acyclicity

Given that node $u$ is the parent of $v$

$n_v$: length of the sentence containing node $v$

$d_v \in [0, n_v]$: depth of the node from the root of the sentence

$$(\forall_u) \ d_v + (1 + n_v) \cdot (1 - x_{uv}) \geq 1 + d_u \quad (1)$$

The depth of a child is greater than the depth of the parent

Connectedness

$$\sum_u x_{uv} \geq 1 \quad (2)$$

A word has at least 1 parent
Consistency of Directionalized Links

Consistency of Directionalized Links

\( r_L, \ell_L \in \{0, 1\} \): whether links with label \( L \) allowed left/right

\[
x_{ij} \leq r_L \quad \quad \quad x_{ji} \leq \ell_L
\]  

(3)

\[
\min \left( \sum_L r_L + \ell_L \right)
\]  

(4)
Consistency of Directionalized Links with Slack

Consistency of Directionalized Links

\( r_L, \ell_L \in \{0, 1\} \): whether links with label \( L \) allowed left/right

\[
\begin{align*}
    x_{ij} & \leq r_L + s_{ij} \\
    x_{ji} & \leq \ell_L + s_{ij}
\end{align*}
\]  

(3)

\[
\min \left( \sum_{L} r_L + \ell_L \right) \cdot \frac{N_L}{4} + \sum_{ij} s_{ij}
\]

(4)

\( s_{ij} \in \mathbb{R} \geq 0 \): slack variable

\( N_L \): Number of link tokens with label \( L \)

Slack allows a few links with label \( L \) in disallowed directions
Data Sets

Data Sets taken from:

CoNLL 2007 Shared Task (English)
ACL 2013 Shared Task of Machine Translation (Russian)

<table>
<thead>
<tr>
<th></th>
<th>Input Sentences</th>
<th>Output Connected Parses</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>18,577</td>
<td>10,960</td>
</tr>
<tr>
<td>Russian</td>
<td>18,577</td>
<td>4,913</td>
</tr>
</tbody>
</table>
Stability of Results

- We worried that the recovered direction mapping might be unstable and sensitive to the input corpus.
- We compared the results of increasing runs of sentences.
On the English Data Set:
On the English Data Set:

Multiheadedness

Link Data has 8% additional edges over the CoNLL. (average about 2 multiheaded words per sentence)
On the English Data Set:

**Multiheadedness**

Link Data has 8% additional edges over the CoNLL. (average about 2 multiheaded words per sentence)

**CoNLL Matches**

52% of links match CoNLL arcs

57% of CoNLL arcs match links
On the English Data Set:

Multiheadedness

Link Data has 8% additional edges over the CoNLL. (average about 2 multiheaded words per sentence)

CoNLL Matches

52% of links match CoNLL arcs
57% of CoNLL arcs match links

Directionality

6.19% of link types allowed both directions
2.07% of link tokens required disallowed direction via slack
## ILP Results: Top 25 Most Occurring Labels

<table>
<thead>
<tr>
<th>Label</th>
<th>Rightward</th>
<th>Multiheaded</th>
<th>CoNLL Match</th>
<th>CoNLL Dir Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0% (0/8501)</td>
<td>0% (0/8501)</td>
<td>84% (7148/8501)</td>
<td>98% (7002/7148)</td>
</tr>
<tr>
<td>AN</td>
<td>0% (0/9401)</td>
<td>0% (0/9401)</td>
<td>83% (7825/9401)</td>
<td>98% (7639/7825)</td>
</tr>
<tr>
<td>B</td>
<td>100% (1514/1515)</td>
<td>61% (919/1515)</td>
<td>53% (806/1515)</td>
<td>84% (678/806)</td>
</tr>
<tr>
<td>C</td>
<td>100% (3272/3272)</td>
<td>0% (0/3272)</td>
<td>3% (85/3272)</td>
<td>53% (45/85)</td>
</tr>
<tr>
<td>CO</td>
<td>0% (0/2478)</td>
<td>1% (32/2478)</td>
<td>5% (114/2478)</td>
<td>68% (78/114)</td>
</tr>
<tr>
<td>CV</td>
<td>100% (3237/3237)</td>
<td>100% (3237/3237)</td>
<td>56% (1827/3237)</td>
<td>28% (512/1827)</td>
</tr>
<tr>
<td>D</td>
<td>0% (56/19535)</td>
<td>0% (71/19535)</td>
<td>85% (16656/19535)</td>
<td>100% (16608/16656)</td>
</tr>
<tr>
<td>E</td>
<td>0% (0/1897)</td>
<td>0% (2/1897)</td>
<td>67% (1279/1897)</td>
<td>99% (1263/1279)</td>
</tr>
<tr>
<td>G</td>
<td>0% (0/6061)</td>
<td>0% (0/6061)</td>
<td>70% (4258/6061)</td>
<td>96% (4070/4258)</td>
</tr>
<tr>
<td>I</td>
<td>100% (5405/5424)</td>
<td>60% (3247/5424)</td>
<td>95% (5168/5424)</td>
<td>47% (2408/5168)</td>
</tr>
<tr>
<td>IV</td>
<td>100% (1626/1627)</td>
<td>100% (1626/1627)</td>
<td>85% (1389/1627)</td>
<td>97% (1353/1389)</td>
</tr>
<tr>
<td>J</td>
<td>98% (16400/16673)</td>
<td>2% (280/16673)</td>
<td>87% (14522/16673)</td>
<td>97% (14069/14522)</td>
</tr>
<tr>
<td>M</td>
<td>100% (9594/9596)</td>
<td>0% (16/9596)</td>
<td>74% (7124/9596)</td>
<td>92% (6583/7124)</td>
</tr>
<tr>
<td>MV</td>
<td>100% (13375/13376)</td>
<td>0% (61/13376)</td>
<td>51% (6797/13376)</td>
<td>98% (6681/6797)</td>
</tr>
<tr>
<td>MX</td>
<td>100% (1999/1999)</td>
<td>4% (83/1999)</td>
<td>42% (836/1999)</td>
<td>91% (763/836)</td>
</tr>
<tr>
<td>O</td>
<td>100% (11027/11028)</td>
<td>0% (0/11028)</td>
<td>81% (8932/11028)</td>
<td>96% (8535/8932)</td>
</tr>
<tr>
<td>P</td>
<td>100% (3755/3756)</td>
<td>31% (1167/3756)</td>
<td>94% (3528/3756)</td>
<td>100% (3523/3528)</td>
</tr>
<tr>
<td>S</td>
<td>97% (13138/13520)</td>
<td>57% (7662/13520)</td>
<td>92% (12476/13520)</td>
<td>5% (586/12476)</td>
</tr>
<tr>
<td>SJ</td>
<td>50% (2736/5468)</td>
<td>0% (0/5468)</td>
<td>69% (3778/5468)</td>
<td>93% (3502/3778)</td>
</tr>
<tr>
<td>TO</td>
<td>100% (1733/1734)</td>
<td>0% (1/1734)</td>
<td>0% (5/1734)</td>
<td>100% (5/5)</td>
</tr>
<tr>
<td>VJ</td>
<td>51% (765/1500)</td>
<td>1% (8/1500)</td>
<td>71% (1059/1500)</td>
<td>89% (939/1059)</td>
</tr>
<tr>
<td>W</td>
<td>100% (10528/10528)</td>
<td>0% (5/10528)</td>
<td>5% (504/10528)</td>
<td>46% (232/504)</td>
</tr>
<tr>
<td>WV</td>
<td>100% (7563/7563)</td>
<td>100% (7557/7563)</td>
<td>57% (4345/7563)</td>
<td>97% (4214/4345)</td>
</tr>
<tr>
<td>X</td>
<td>80% (13132/16406)</td>
<td>5% (806/16406)</td>
<td>8% (1364/16406)</td>
<td>95% (1300/1364)</td>
</tr>
<tr>
<td>YS</td>
<td>0% (0/1645)</td>
<td>0% (0/1645)</td>
<td>98% (1619/1645)</td>
<td>0% (0/1619)</td>
</tr>
</tbody>
</table>
Link Results: Subject-Verb Links are Backwards

-the matter may never even be tried in court.
Link Results: Subject-Verb Links are Backwards

-the matter may never even be tried in court.

DT NN MD RB RB VB VB IN NN.
Link Results: Subject-Verb Links are Backwards

- This is due to a possible inconsistency of the Link Grammar, discovered by our method.

\[
\text{Jill thinks he will skip}
\]

\[
\text{Jill hopes to skip}
\]
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Motivation</th>
<th>Link Grammars</th>
<th>ILP Model</th>
<th>Experiments and Results</th>
<th>Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subject-Verb Links

**Link Results: Subject-Verb Links are Backwards**

- The Link Grammar seems to be inconsistent about whether the auxiliary verb or the main verb is the head of a clause.
Link Results: Subject-Verb Links are Backwards

- The Link Grammar seems to be inconsistent about whether the auxiliary verb or the main verb is the head of a clause.
- Sometimes the governing verb links to the auxiliary, and sometimes to the main, depending on the type of clause.
Link Results: Subject-Verb Links are Backwards

- The Link Grammar seems to be inconsistent about whether the auxiliary verb or the main verb is the head of a clause.
- Sometimes the governing verb links to the auxilliary, and sometimes to the main, depending on the type of clause.
- But the governing verb usually links to the subject when there is one.
Link Results: Subject-Verb Links are Backwards

- The Link Grammar seems to be inconsistent about whether the auxiliary verb or the main verb is the head of a clause.
- Sometimes the governing verb links to the auxiliary, and sometimes to the main, depending on the type of clause.
- But the governing verb usually links to the subject when there is one.
- So this makes the subject a consistent choice to make the head of a clause.
Link Results: Subject-Verb Links are Backwards

- The Link Grammar seems to be inconsistent about whether the auxiliary verb or the main verb is the head of a clause.
- Sometimes the governing verb links to the auxiliary, and sometimes to the main, depending on the type of clause.
- But the governing verb usually links to the subject when there is one.
- So this makes the subject a consistent choice to make the head of a clause.

To fix this, we could edit the link grammar, link parses, or the ILP.
Conclusions

- Link Grammar parses can be oriented into connected DAGs
- New corpora for building multi-headed dependency parsers
- ILP can be used to help annotate missing data in corpora.
Questions?