Conversions between dependencies & constituencies

L715: Seminar on: Data manipulation for parser improvement
Dept. of Linguistics, Indiana University
Fall 2011
Introduction

We are going to focus on dependency & constituency conversions

▶ Mostly in the direction of: constituency $\mapsto$ dependency
  ▶ Parser evaluation: dependencies are argued to better capture important aspects of evaluation (e.g., Lin 1995)
  ▶ Dependency parsing has become increasingly popular (linear time algorithms, closeness to semantics, multi-linguality, ...)
    ▶ But constituency treebanks were developed first (at least for English)
  ▶ Even for PCFGs, dependency information can help the parsing model (e.g., Collins 1999)

We’ll focus on English ...
Earlier work on conversions

- Magerman (1994); Collins (1999) developed rules to identify the head of a constituent in the PTB
  
  (1) $\text{NP} \rightarrow \text{DT} \text{ NN}^*$
  
  (2) $\text{VP} \rightarrow \text{VBD}^* \text{ NP}$
  
  (3) $\text{S} \rightarrow \text{NP} \text{ VP}^*$

  ▶ *head percolation table*: priority lists to identify the head in each type of constituent

- Yamada and Matsumoto (2003): modified the table

  
  ▶ plus: defined arc labeling rules
  
  ▶ See figures 1 & 2 in Johansson and Nugues (2007)
Extending conversions to be more semantic
Johansson and Nugues (2007)

**Goal:** improve upon previous methods by making dependencies which interlace better with semantics

- richer set of labels
- better treatment of long-distance phenomena
  - PTB-II contains information on *wh*-movement, topicalization, *it*-clefts, expletives, & gapping
  - older conversions do not use such information

Make use of extended structure in PTB-II

- The conversion procedure will illustrate a number of issues facing any parser or any annotation scheme
New procedure

**Penn2Malt:**

Why, they wonder, should it belong to the EC?

New conversion (LTH):

Why, they wonder, should it belong to the EC?
New procedure (2)

1. Modify dependency links
   - e.g., \textsc{Penn2Malt} misses relation between \textit{belong} and \textit{Why}

2. Richer set of dependency labels
   - e.g., \textsc{Penn2Malt} only used SBJ and PRD from PTB-II grammatical function labels
Heuristically deepening NPs

Need to add internal structure to NPs: PTB has flat structure

- e.g., flat NP of *other small apparel makers*
- ... but not every word is truly dependent on the head noun (*makers*)

Heuristics:

- certain adverbs (e.g., *quite*) are joined with consecutive adjective to form ADJP
- certain words in coordinated NPs (e.g., *and Sons*) provide clues as to bracketing
- words with identical POS around a conjunction assumed to be coordinated (e.g., *small and venomous snake*)

nb: see also Vadas and Curran (2007) for NP deepening
Head rules from before are adapted

- make use of the context of phrases
- make use of grammatical functions
  - e.g., SQ ← VBZ VBD VBP VB MD VP *-PRD VP SQ

See table 1 in the paper
Head rule modifications (2)

- Coordinated phrases
  - Leftmost conjunct is consistently the head & all other conjuncts/conjunctions are children of the first conjunct
  - Conjunction-as-head would better capture premodifier ambiguities, but “this is usually not preferred since it makes parsing more difficult”

- PPs, subordinate & relative clauses
  - Preposition (subordinating conjunction, relativizer) is treated as a case marker
  - i.e., they are dependents, and the main contentful word is the head
Modification of arc labeling rules
Grammatical functions from Penn

Used 17 of 21 grammatical function labels to label dependency relations

- properties may be combined (e.g., LOC-PRD-TPC)
- excluded ones reflecting structural properties & not grammatical functions (e.g., HLN (headline))
Modification of arc labeling rules

Inferred labels

Most edges in PTB have no label, so they must be inferred

- Objects includes clause complements (S, SBAR): different from previous approaches
- Distinguish OBJ from IOBJ
- Distinguish types of ROOTs: ROOT-S, ROOT-SBARQ, ROOT-SQ, ROOT-FRAG

If the inference fails, DEP is returned

(See Algorithm 1 for more details)
Structural labels
Expeltives & clefts

EXP (expletive) and CLF (cleft) are structural labels, but represent complex constructions

- result in a fronted *it*
- handled different in PTB, but similarly after conversion
Gapping involves a verb elided in a coordinate structure

- Secondary edges are used in the PTB to account for this

Prices were mixed in Zurich and lower in Stockholm
Relinking of secondary edges

Secondary edges used for a variety of purposes in PTB

- When they represent a “deep governor”, they are useful as dependency arcs (close to semantics)
  - e.g., *T* (trace of wh & topicalization), *ICH* (discontinuous constituent)
  - Such cases are relinked (unless cyclicity is introduced)
- *RNR* (right node raising), e.g., *a U.S. and a Soviet naval vessel*
  - Of the two secondary edges, only the first one is used for conversion

Conversion introduces nonprojectivity: 6.17% of the sentences
Experiments

Impact on parsing performance

Parsing is more difficult with the new conversion:

- links can now be nonprojective
- more function tags (e.g., PENN2MALT does not distinguish temporal & locative adjuncts)
- “linking words” (e.g., prepositions) do not attach to the verb
  - MSTParser cannot use grandchildren features & thus loses lexical information

<table>
<thead>
<tr>
<th></th>
<th>MaltParser</th>
<th></th>
<th>MSTParser</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LAS</td>
<td>UAS</td>
<td>LAS</td>
<td>UAS</td>
</tr>
<tr>
<td>PENN2MALT</td>
<td>90.30%</td>
<td>91.36%</td>
<td>92.04%</td>
<td>93.06%</td>
</tr>
<tr>
<td>LTH</td>
<td>87.63%</td>
<td>90.54%</td>
<td>86.92%</td>
<td>91.64%</td>
</tr>
</tbody>
</table>
Experiments
Impact on semantic role classification

Semantic role classification displays a different trend, illustrating the usefulness of this conversion:

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PENN2MALT</td>
<td>64.3%</td>
</tr>
<tr>
<td>LTH</td>
<td>72.5%</td>
</tr>
</tbody>
</table>

Granularity of edge labels makes a difference

▶ e.g., in the RECEIVING frame, grammatical functions express twice as many semantic roles ... see table 6
Robust conversion
Choi and Palmer (2010)

**Problem:** conversion tools are highly tailored to a specific annotation scheme

- do not work well on other corpora
- **Goal:** improve conversion portability

Specifically, they work with the OntoNotes data

- POS tags not in the original PTB (e.g., EDITED, META)
- Sometimes depart from PTB conventions (e.g., inserting NML phrases, separating hyphenated words)

Practical effect:

- Fewer unnecessary non-projective dependencies
- Fewer unclassified dependencies
- Improved parser accuracy (across corpora)
Head-percolation rules

They designed a new set of head-percolation rules

▶ incorporates new tags, e.g.,

\[
\text{META} \quad \text{r} \quad \text{VP};\text{NP};^* \\
\]

▶ Makes greater use of function tags: in addition to PRD, use SBJ (subject) and TPC (topic)

\[
\text{VP} \quad \text{l} \quad \text{TO};\text{MD};\text{VB}^*;\text{VP};^*-\text{SBJ};^*-\text{TPC};^*-\text{PRD}; \\
\text{NN};\text{NNS};\text{NP};\text{QP};\text{JJ}^*;\text{ADJP};^* \\
\]

▶ Minor modifications to some rules: e.g., rule for ADJP now gives higher priority to adjectives than nouns
Small clauses

Small clause treatment in the PTB:

```
S
  | NP
  |   VP
  |    | S-1
  |    | NP-SBJ
  |    | ADJP-PRD
  | VBP
  |   happy
  |   us
  | made
  | He
```
LTH:

```
root He made us happy
```

Choi & Palmer:

```
root He made us happy
```

- Latter approach meshes better with PropBank, where “us happy” is annotated as a single argument of *made*.
  - harder to derive a constituent from LTH approach
- nb: both approaches are linguistically valid (cf. coordination treatments)
Function tags

14 function tags are used to create dependency labels

- LTH converts joined tags (e.g., LOC-TMP) into unique tags
- Choi & Palmer select one tag from a joined pair
  - e.g., LOC-TMP $\mapsto$ LOC
  - based on the notion that parsers do not often get joined tags correct (cf. *external criteria*)

Precendence table:

```
DTV|EXT|LGS|SBJ > LOC > BNF|DIR|MNR|PRP|TMP >
  SEZ|VOC > PRD > ADV
IGNORE ::= CLF|CLR|ETC|HLN|IMP|NOM|PUT|TPC|TTL|UNF
```
Coordination

Take a right-branching approach for coordination

- Difficulty: does a phrase contain coordination?
  - contains UCP, a child annotated with a function tag (ETC), or at least one conjunction (CC) or CONJP
  - “Even if there is a conjunction, if either the left or the right conjunct does not appear within the same phrase, we do not consider there to be a coordination”

```
root We sold old books and then bought new books
```

Note in the algorithm (p. 59) that SKIP defines POS tags which are skipped to find the correct conjuncts
Gapping relations

Parsers perform poorly on gapping constructions

- LTH tends to give flat structures with long-distance dependencies
- ... which parsers generally get wrong

LTH:

```plaintext
root Some said₁ Putin visited in April, some said₂ May
```

References

- Gapping relations
- Parsers perform poorly on gapping constructions
- LTH tends to give flat structures with long-distance dependencies
- ... which parsers generally get wrong

**References**

- Johansson and Nugues (2007)
- Choi and Palmer (2010)
Gapping relations (2)

Choi & Palmer:

- Parsers can now learn more local relations
- The GAP relation allows one to recover the original representation
Empty category mappings

Right node raising is treated slightly differently by Choi & Palmer

- remove link between first conjunct and object
- eliminates non-projective dependencies, but keeps semantic interpretation recoverable

root I know his admiration for and trust in you
Evaluation

- Non-projective dependencies go from 0.82% (LTH) to 0.73% (table 4)
  - largely due to *RNR* treatment
- Unclassified dependencies go from 2.20% (LTH) to 0.60% (table 5)

Parsing accuracy also increases (tables 6 & 7), as does accuracy on semantic dependencies (tables 8 & 9)
References


