Treebank Analysis Using Derivation Trees

Seth Kulick
skulick@seas.upenn.edu
Overview

◆ Background
  ● Treebank – a collection of sentences annotated (by people).
  ● The role of treebanks in Natural Language Processing (NLP).

◆ The Problem
  ● Automatically discovering internal inconsistency in treebanks.

◆ Our approach
  ● System output

◆ Adaptation for parser evaluation/inter-annotator agreement

◆ Future work
The wholesale price index stood at 90.1

- Each sentence has a tree with syntactic information
- The annotation is done (ideally) in conformity with a set of annotation guidelines
  - Treebank annotation is hard and error-prone
The role of treebanks in NLP

- Penn Treebank (1994) – 1M words from the Wall Street Journal – “statistical revolution in NLP”
- Training and evaluation material for parsers
- Parsers – machine learning problem
  - input: new sentence, output: tree for that sentence
  - used for downstream processing
- Many approaches to parsing
  - Break down full trees into smaller parts that can be used to assign structure to a new sentence
  - Rely on consistency of annotation.
- What does it mean for a treebank to be inconsistent?
Annotation inconsistency: Example 1

- Two instances of “The wholesale price index” in a treebank.

- Suppose a parser encountered “The wholesale price index” in new data. How would it know what to do?
Annotation inconsistency: Example 2

Two instances of “foreign trade exports” in a treebank

```
NP
  NML
    foreign
    trade
  exports
```

```
NP
  foreign
  trade
  exports
```
Annotation inconsistency: Example 3

- Two instances of “than average” in a treebank

These examples are not “noise” in naturally occurring data. They are inconsistencies in the annotation that should not exist.
The Problem – How to detect inconsistencies

- A million words of such trees – what to compare, how to find inconsistencies?
The Problem – How to detect inconsistencies

- After approx. 20 years of making treebanks, this is still an open question.

- Usual approach -
  - Hand-crafted searches for structural problems (e.g. two SBJs for a verb, unexpected unary branches, etc.)

- But we want a way to automatically discover inconsistencies.
  - Dickinson & Meurers (TLT 2003…)
  - Kulick et al. (ACL 2011, LREC 2012, NAACL 2013)

- Also for treebanks meant for linguistic research, not as training material for parsers.
The Importance of the Problem

◆ Lots of treebanks
  ● Penn Treebank 1M words (DARPA)
  ● Ontonotes 1.3M words - (DARPA)
  ● Arabic Treebank 600M – (DARPA)
  ● PennParsed Corpora of Historical English 1.8M (NSF)

◆ Treebank construction is expensive for annotation
  ● Need faster and better treebank construction
  ● NSF Linguistics/Robust Intelligence (Kroch & Kulick)

◆ Treebank analysis overlaps with key concerns
  ● How to increase annotation speed
  ● How to determine features for parsers.
Our approach

- Similarities to parsing work – break down full trees to compare parts of them
  - Adapt some ideas from parsing research
  - Tree Adjoining Grammar (Joshi et al...)

- Basic idea: Decompose each tree into smaller chunks of structure
  - “Derivation tree” relates these chunks together
  - Compare strings based on their derivation tree fragments

- Two advantages
  - Group inconsistencies by structural properties
  - Abstract away from interfering surface properties
The wholesale price index stood at 90.1.
Treebank Decomposition

Decomposition done by heuristics controlling the descent through the tree.

The separate chunks (elementary trees) become the nodes of the “derivation tree”.

The wholesale price index stood at 90.1.
Each node is an elementary tree.
Derivation Tree

-derivation tree fragment for “the wholesale price index”
The wholesale price index was 180.9
The wholesale price index was 180.9
another derivation tree fragment for “the wholesale price index”
The Basic Idea

- “nucleus” – a sequence of words being examined for consistency of annotation
- Collect all the instances of that nucleus
  - Each instance has an associated derivation tree fragment
  - Have a set of all derivation tree fragments for the nucleus
- If more than one derivation tree fragment, flag the nucleus as inconsistent
- Simplifying the problem
Derivation Tree Fragments

Two different derivation tree fragments for “the wholespace price index” – flag as inconsistent
Organization by Annotation Structures

◆ Great advantage of this approach
  ● For any set of instances for a nucleus, we have the derivation tree fragments for each instance
  ● Group nuclei by the set of derivation tree fragments used for all the instances that the nucleus has.
Another Annotation Inconsistency

- Two instances of “the economic growth rate”

- Inconsistent, just like “the wholesale price index”
Derivation Tree Fragments

Aside from the words, exactly the same as the derivation tree fragments for “the wholesale price index”
Organization by Annotation Structures

Why this is good:

- e.g. “a short interest position”, “the economic growth rate”, “the real estate industry” all annotated inconsistently in the same way
- We care more about different types of consistencies/inconsistencies than individual cases of words.
- Very helpful for identifying the sorts of errors annotators might be making, or problematic areas for a parser.
System Overview

◆ Compute derivation tree for each sentence
◆ Identify strings to compare - the “nuclei”
  ● For now, use all strings that are constituents somewhere in the corpus. (this can’t be right in the long run)
◆ Get the derivation tree fragments for each nucleus
  ● Include nuclei with more than one derivation tree fragment.
  ● Sort by set of derivation tree fragments used for a nucleus.
◆ Software – getting in shape for release
  ● Derivation tree and elementary tree extraction – Java
  ● Everything else – MySQL and Python. (trees in MySQL)
  ● Output - static HTML for now.
◆ Used for current treebanking
Why not just compare subtrees?

- Two instances of “The wholesale price index” in a treebank.

  ![Tree Structure]

- Why bother decomposing it?
  - Adjunction/coordination
  - Partial constituents/Treebank simplifications
"of the class" modifying "the teacher"

Suppose we want to look at all instances of "the teacher of the class"
“that I took” modifying “the class”

No subtree with just “the teacher of the class”
Adjunction in Derivation Tree Fragments

- nucleus “the teacher of the class”
- a1-a5 and b1-b5 are compared
- No interference from b6 (and below).
Arabic Treebank example
nucleus is “summit Sharm el-Sheikh”

Annotation is different

Annotation is same
Arabic Treebank example
nucleus is “summit Sharm el-Sheikh”
Further Abstraction from Treebank

Two instances of nucleus “one place”

- Not a constituent in one of them.
Further Abstraction from Treebank

- Change #b2 to not have the QP
- More abstract representation also used for parsing
(Partial) Evaluation

◆ Arabic Treebank 598K words (Maamouri et al, 2008-9)
  ● First 10 annotation types include 266 nuclei, all correctly identified
  ● recall of inconsistencies – hard to measure

<table>
<thead>
<tr>
<th># Nuclei</th>
<th># Reported</th>
<th># non-duplicate</th>
<th>#Annotation Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>54,496</td>
<td>9,984</td>
<td>4,272</td>
<td>1,911</td>
</tr>
</tbody>
</table>

◆ Ontonotes English newswire 525K words
  ● evaluation so far – successful, not perfect

<table>
<thead>
<tr>
<th># Nuclei</th>
<th># Reported</th>
<th># non-duplicate</th>
<th>#Annotation Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>30,497</td>
<td>3,609</td>
<td>3,012</td>
<td>1,186</td>
</tr>
</tbody>
</table>
System Output

- Similar words grouped together
Parser Analysis/Inter-Annotation Agreement

- Method of finding nuclei is in general a problem
  - Reliance on identical strings of words – “later”, “l8r”
  - Must occur at least once as a constituent

- But in one situation it is completely natural
  - Comparing two sets of annotations over the same sentences
    Kulick et al, NAACL 2013

- Parser evaluation –
  - Output of parser compared to treebank gold

- Inter-annotator Agreement Evaluation
  - Two annotators working on the same sentences
Example of Inter-annotator (dis)agreement

- Two annotators on same sentence

- What to compare? Original starting point of work
IAA evaluation

- Evaluation on 4,270 word pre-release subset of Google English Web Treebank (Bies et al, 2012)

<table>
<thead>
<tr>
<th>Inconsistency Type</th>
<th># Found</th>
<th># Accurate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Tags only</td>
<td>53</td>
<td>53</td>
</tr>
<tr>
<td>POS Tags Only</td>
<td>18</td>
<td>13</td>
</tr>
<tr>
<td>Structural</td>
<td>129</td>
<td>122</td>
</tr>
</tbody>
</table>

- Evaluation on 82,701 word pre-release supplement Modern British English (Kroch & Santorini, in prep)
  - 1,532 inconsistency types with 2,194 nuclei
  - First has 88 nuclei, second has 37
  - First 20 (375 nuclei) all true instances of inconsistent annotation
Future work

- Clustering based on words and derivation tree fragments.
  - Problems of multiple spellings - webtext, historical corpora
- Order output based on some notion of what’s more likely to be an error.
- Dependency work.
- Parsing work based on derivation trees.