Features

- “Implicit” Syntax
- Shallow Syntax (POS, chunks)
- Deep Syntax (trees)
- Tricky Syntax (tree fragments)
Deep Syntax

• What is deep? — use of parser output

• Why parser? — grammaticality can be measured by parse trees

• How to use parser output?
  – simple features
  – model-based features
  – dependency-based features
  – tree fragments
Simple features: Parser score

Motivation: grammatical sentences should have higher parse prob.

Feature Functions:

- $\log(\text{parseProb})$  
  (Alex)
- $\log(\frac{\text{parseProb}}{\text{trigramProb}})$  
  (Anoop)

Result: worse than baseline
Does Parser give high probability for grammatical sentence?

Parser LogProb for produced/oracle/reference sentences (Shankar)

<table>
<thead>
<tr>
<th></th>
<th>log(parseProb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>produced</td>
<td>-147.2</td>
</tr>
<tr>
<td>oracle</td>
<td>-148.5</td>
</tr>
<tr>
<td>ref 1</td>
<td>-148.0</td>
</tr>
<tr>
<td>ref 2</td>
<td>-157.5</td>
</tr>
<tr>
<td>ref 3</td>
<td>-155.6</td>
</tr>
<tr>
<td>ref 4</td>
<td>-158.6</td>
</tr>
</tbody>
</table>

Syntax for Statistical MT

JHU 2003 WS
Other simple parse-tree features

Motivation: grammatical sentences should have specific tree shape.

Feature Functions: (Anoop)

- right branching factor
- tree depth
- num. of PPs
- VP probs
- ...

Syntax for Statistical MT  JHU 2003 WS
Model-based features

Translation Model as Feature Function

- Originally developed as a standalone model $P(f|e)$
  - Syntax-based model for parse trees

- $P(f|e)$ can be used as a feature value
  - Tree-based models represent systematic difference between two languages’ grammar
    - e.g. SVO vs. verb-final word order
    - constituents (e.g. NP) tend to move as a unit

- Better translation should yield higher probs

- featureVal = $\log[P(f|e)]$
Syntax-based Translation Model

Tree-based probability model for translation

- Early work:
  - Inversion Transduction Grammar [Wu 1997]
  - Bilingual Head Automata [Alshawi, et. al 2000]

- Tree-to-String [Yamada & Knight 2001]

- Tree-to-Tree [Gildea 2003]
Syntax-based Translation Model (cont)

Probabilistic operation on parse tree:

- Reorder
- Insert
- Translate
- Merge
- Clone

Parameters are estimated from training pairs (Tree/Tree, Tree/String) using EM algorithm.
Tree-to-String Alignment

Yamada & Knight 2001

re-order step: \( P_r(3, 4, 2, 1 \mid S \Rightarrow NP \; NP \; NP \; VB) \)
Tree-to-String Alignment 2

**Insertion Step:** $P_{\text{ins}}(\text{the}) P(\text{ins} | \text{NP})$

**Translation Step:** $P_t(\text{give} | \text{Cu})$

**Syntax for Statistical MT JHU 2003 WS**
Tree-to-Tree Alignment

Chinese tree:

```
   xianzhu
    /   \
  chengshi   chengjiu
   /         /   \
Zhongguo  shisi  bianjing  kaifang  jingji  jianshe
```

Merge/Split nodes:

```
   xianzhu
    /     \
  chengshi  chengjiu  jianshe
   /         /             \
Zhongguo  shisi   ge    bianjing  kaifang  jingji
```

Reorder:

```
   xianzhu
    /   \
  chengshi   chengjiu  jianshe
    /     /       \
Zhongguo  shisi  ge  kaifang  bianjing  jingji
```

Lexical Translation:

```
   marked
     /   \
  cities  achievements
    /         \
's 14 open border economic
   China
```

Syntax for Statistical MT  JHU 2003 WS
Cloning example

Syntax for Statistical MT

JHU 2003 WS
Problems

- $n$-best list doesn’t contain big word jump
  - reordering at upper node is useless

- English/Chinese word-order is almost the same
  - both SVO in general
  - but relative clause comes before noun

- Computationally expensive
  - use word-level alignment from MT output
  - limit by sentence length and fanout
  - break up long sentences into small fragments (machete)
Experiments

Tree-to-String (Kenji, Anoop)

- Trained on 3M words of parallel text
  - English side parsed by Collins
- Max sentence length 20 Chinese characters
  - 273/993 sentences covered

Tree-to-Tree (Dan, Katherine)

- Trained on 40,000 biparsed FBIS sentences
- Max fan-out 6, max sentence length 60
  - 525/993 sentences covered
## Results

<table>
<thead>
<tr>
<th></th>
<th>BLEU%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>31.6</td>
</tr>
<tr>
<td>ParseProb</td>
<td>31.6</td>
</tr>
<tr>
<td>ParseProbDivLM</td>
<td>31.0</td>
</tr>
<tr>
<td>RightBranching</td>
<td>31.6</td>
</tr>
<tr>
<td>TreeDepth</td>
<td>31.5</td>
</tr>
<tr>
<td>numPPs</td>
<td>31.3</td>
</tr>
<tr>
<td>VPPProb</td>
<td>31.3</td>
</tr>
<tr>
<td>Tree-to-String</td>
<td>31.7</td>
</tr>
<tr>
<td>Tree-to-Tree</td>
<td>31.6</td>
</tr>
</tbody>
</table>
Lessons / Directions

- Feature combination: BLEU 31.6 → 33.2
- But two thirds of improvement from lexical probs (IBM model 1)
- Hard to use off-the-shelf taggers, parsers, etc
- Limitations of rescoring n-best lists: syntax-based decoders
- Problems with evaluation metric:
  - human evaluation
  - syntax-based measures