CS 294-5: Statistical Natural Language Processing

Machine Translation III
Dan Klein
includes slides from Yamada, Knight, Koehn

Why Syntactic Translation?

He adores listening to music.

From Yamada and Knight (2001)

Two Places for Syntax?

• Language Model
  Can use with any translation model
  Syntactic language models seem to be better for MT than ASR (why?)
  Not thoroughly investigated [Charniak et al 03]

• Translation Model
  Can use any language model
  Linear LM can complement a tree-based TM (why?)
  Also not thoroughly explored [Yamada and Knight 01]

Parse Tree (E) → Sentence (J)

1. Reorder

Assignment 2 Honors

V(P(NN NN) → TO NN) = 0.893
V(PP TO → TO VB) = 0.749
V(PP VB1 VBP VB2 VBP) = 0.723
Conditioning Feature = Child label Sequence
Parameter Table: Reorder

| Original Order | Reordering | P(reorder|original) |
|----------------|------------|---------------|
| PRP VB1 VB2    | PRP VB1 VB2| 0.723         |
| PRP VB1 VB2    | PRP VB1    | 0.037         |
| VB1 PRP VB2    | VB2 PRP    | 0.074         |
| VB1 PRP VB2    | VB1 VB2 PRP| 0.083         |
| VB2 PRP VB1    | VB2 PRP    | 0.021         |
| VB1 VB2        | VB2 VB1 PRP| 0.107         |
| VB TO          | VB TO      | 0.093         |
| TO NN          | TO NN      | 0.749         |

2. Insert

P(non(TOP-VB)) = 0.735
P(right)(VB-PRP) * P(ha) = 0.652 * 0.219
P(right)(VB-VB) * P(ga) = 0.252 * 0.062
P(non(TO-TO)) = 0.900

Conditioning Feature = Parent Label & Node Label (position)
none (word selection)

Parameter Table: Insert

| Original Order | Reordering | P(insert|original) |
|----------------|------------|--------------|
| PRP VB1 VB2    | PRP VB1 VB2| 0.004        |
| PRP VB1 VB2    | PRP VB1    | 0.687        |
| VB1 PRP VB2    | VB2 PRP    | 0.260        |
| VB1 PRP VB2    | VB1 VB2 PRP| 0.652        |
| VB2 PRP VB1    | VB2 PRP    | 0.004        |
| VB2 VB1 PRP    | VB2 VB1    | 0.062        |
| VB2 VB1 PRP    | VB2 VB1    | 0.021        |
| VB TO          | VB TO      | 0.219        |
| TO NN          | TO NN      | 0.094        |

3. Translate

P(cos+ kare) = 0.952
P(music+ ongaku) = 0.900
P(to+ wo) = 0.333
P(adore+ daisuki) = 1.000

Conditioning Feature = word (E) identity

Parameter Table: Translate

<table>
<thead>
<tr>
<th>Original Order</th>
<th>Reordering</th>
<th>Reordering (NULL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>1.000</td>
<td>NULL</td>
</tr>
<tr>
<td>NULL</td>
<td>0.133</td>
<td>NULL</td>
</tr>
<tr>
<td>NULL</td>
<td>0.133</td>
<td>NULL</td>
</tr>
<tr>
<td>NULL</td>
<td>0.133</td>
<td>NULL</td>
</tr>
<tr>
<td>NULL</td>
<td>0.100</td>
<td>NULL</td>
</tr>
<tr>
<td>NULL</td>
<td>0.204</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Note: Translation to NULL = deletion

Experiment: Y+K 03

- Training Corpus: J E2K sentence pairs
- J: Tokenized by Chasen [Matsumoto, et al., 1999]
- E: Parsed by Collins Parser [Collins, 1999]
- E: Flattened parse tree
- EM Training: 20 Iterations
- 50 miniter (Sparc 200MHz 1-CPU) or
- 30 seciter (Pentium3 700MHz 30-CPU)
Result: Alignments

<table>
<thead>
<tr>
<th>Model</th>
<th>Ave. Score</th>
<th># perf sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y/K</td>
<td>0.582</td>
<td>10</td>
</tr>
<tr>
<td>IBM 5</td>
<td>0.431</td>
<td>0</td>
</tr>
</tbody>
</table>

- Ave. by 3 humans for 50 sents
- okay(1.0), not sure(0.5), wrong(0.0)
- precision only

Result: Alignment Example

Syntax-based Model:
He adores listening to music

IBM Model 3:
彼は音楽を聞くのが大好きです

Synchronous Grammars

- Multi-dimensional PCFGs (Wu 95, Melamed 04)
- Both texts share the same parse tree:

```
S → NP VP
S → NP VP
VP → V NP
VP → NP V
```

- ... with probabilities, of course!
- Distribution over tree pairs
- Strong assumption: constituents in one language are constituents in the other
- Is this a good assumption? Why?

Details

- Distinctions in lines of work are in the details:
  - What about insertions?
  - What about deletions?
  - How flat can rules be?
  - Multiple transductions of rules?

- Recent work (Eisner 04, Melamed 04) much more flexible than early work
  - ... but still no killer results