Outline

1. Model of Translation
2. Decoding Strategy
3. Data Processing
4. Evaluation
Issues Addressed

- Extended Translation models for SMT
- Data Processing (Case + Punctuation)
- Syntax Augmented Machine Translation via Chart Parsing
Generalized Rules

- Extending the model of translational equivalence
  - he does not go, *il ne va pas*
  - does not go, *ne va pas*
  - does not X, *ne X pas*
    - Chiang, 05, Watanabe et al. 06
  - S → does not VB0, *ne X0 pas*
    - Galley, 04, Zollmann, Venugopal, 06, Galley et al. 06
Start with word alignments (a) on $f, e +$ phrase based model
We want to model target language syntactic structure
Take advantage of target side parse tree $\pi$
Data Oriented Approach

- Consider $\pi, f, e, phrases(a)$ as given
- Extract rules consistent with $phrases(a)$
- Begin with contiguous src, tgt phrases
- Annotate, Generalize, Re-order
- Producing “flattened” xRS style rules
- CCG style operations-add/subtract
Alignment Graph

```
S
  NP    VP
    PRP  AUX  RB  VB
    he   does not  go
  il  ne  va  pas
```

He does not go
Il ne va pas
Extraction Approach

il ne va pas

he does not go

goes

does he

il ne va pas

S

RB+VB

VB

VP-AUX

NP+AUX

NP

il ne va pas
Alignment Graph

- **INITIAL+ANNOTATED**
  - PRP → he, il
  - VB → go, va
  - VP → does not go, ne va pas
  - S → he does not go, il ne va pas
  - PRP+AUX → il, he does

- **GENERALIZED**
  - S → he VP₀, il x₀
  - VP → does not VP₀, ne x₀ pas

Ashish Venugopal  
IWSLT2006, 11/28/06
All rules extracted

- \( S \rightarrow PP \text{ ne } VB \text{ pas}, 1 \text{ do not } 2 \)
- \( PP+AUX \rightarrow PP, 1 \text{ do} \)
- \( S \rightarrow PP+AUX \text{ RB+VB}, 1 \ 2 \)
- \( RB+VB \rightarrow \text{ ne vais pas}, \text{ not go} \)
- \( S \rightarrow PP+AUX \text{ ne vais pas}, 1 \text{ not go} \)
- \( S \rightarrow \text{ je RB+VB}, i \text{ do } 1 \)
- \( S \rightarrow \text{ PP RB+VB}, 1 \text{ do } 2 \)
- \( VP \rightarrow \text{ ne VB pas}, \text{ do not } 1 \)
- \( S \rightarrow \text{ je ne VB pas}, i \text{ do not } 1 \)
- \( S \rightarrow \text{ je VP}, i \ 1 \)
- \( PP \rightarrow \text{ je }, i \)
- \( RB+VB \rightarrow \text{ ne VB pas}, \text{ not } 1 \)
- \( VB \rightarrow \text{ vais}, \text{ go} \)
- \( S \rightarrow \text{ je ne vais pas}, i \text{ do not go} \)
Chart Parser Based Decoding

- Earley style bottom-up parsing
- We do not require rule binarization as in (Huang, 06)
- Integrated N-gram Language Model (Wu, 98)
  - Single-pass Cube Pruning (Chiang, 05)
  - Multi-pass heuristic search (Zollmann, Venugopal, 06)
- All rules stored in Berkeley DB
Log-Linear translation model features

- Source, target conditioned lexical weights as in Koehn, 2003
- Relative frequencies conditioned on ⋅ ⋅ ⋅
  - Left-hand side category
  - Source phrase
  - Target phrase
- Counters: Rule applications, Target words generated
- Bias terms
  - IsPurelyLexical
  - IsPurelyAbstract
  - IsXRule (non-syntactic span)
  - IsGlueRule
- Penalty features
  - Rareness: $e^{1 - \text{RuleFrequency}}$
  - Lexical Length Balance: $|\text{MeanTargetSourceRatio} \times |src| \times tgt|$
Case Handling

- Training vs Translation time options
  - TrainLowerCase: training lower case + true-case output
  - TrainTrueCase: training true case
  - SmartCase: the ambiguity is in the first word!
    - Upper-case words used within the sentence tend to be consistent
    - For each first-word - estimate non-first word case frequency
    - Use most common case consistently
    - Upper-case first word of output

Ashish Venugopal
IWSLT2006, 11/28/06
### SmartCasing on Training Data

<table>
<thead>
<tr>
<th>English word</th>
<th>Frequency</th>
<th>Fr. of lower-case variant</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>21611</td>
<td>10</td>
</tr>
<tr>
<td>Japan</td>
<td>1939</td>
<td>0</td>
</tr>
<tr>
<td>Japanese</td>
<td>1816</td>
<td>0</td>
</tr>
<tr>
<td>Tokyo</td>
<td>813</td>
<td>0</td>
</tr>
<tr>
<td><strong>Hotel</strong></td>
<td>703</td>
<td>1166</td>
</tr>
<tr>
<td>Mr.</td>
<td>691</td>
<td>0</td>
</tr>
<tr>
<td>New</td>
<td>487</td>
<td>315</td>
</tr>
<tr>
<td>York</td>
<td>438</td>
<td>0</td>
</tr>
<tr>
<td>English</td>
<td>372</td>
<td>0</td>
</tr>
<tr>
<td>Boston</td>
<td>318</td>
<td>0</td>
</tr>
</tbody>
</table>
Punctuation

- Remove punctuation in source training to match test
- Move sentence-end marks “.?!” to the beginning
- Train model that generates punctuation
- Move leading punctuation to end in output
Times and Numbers

- Number handling based on quantifier markers
- Specific handling for Chinese “point” symbol
Data Conditions

- C-Star Data Track participants
- Focussing on correct transcription evaluation
- Note: Error in data alignment, only Supplied Data was parallel
- 2006 Development development data only
## Comparison of Casing - PostEval

<table>
<thead>
<tr>
<th>Processing</th>
<th>Dev IBM-BLEU</th>
<th>Test IBM-BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrainLowerCase</td>
<td>22.04 (23.91)</td>
<td>19.16 (21.55)</td>
</tr>
<tr>
<td>TrainTrueCase</td>
<td>22.47 (24.05)</td>
<td>19.23 (20.48)</td>
</tr>
<tr>
<td>SmartCase</td>
<td>23.50 (25.17)</td>
<td>20.04 (21.76)</td>
</tr>
</tbody>
</table>

**Table:** Comparison of different case-handling methods using the syntax-augmented translation system evaluated on the official case- and punctuation-sensitive IBM-BLEU metric. The numbers in parentheses indicate the IBM-BLEU score when case (but not punctuation) is ignored.
### Comparison of Models - PostEval

<table>
<thead>
<tr>
<th>Rules</th>
<th>Dev IBM-BLEU</th>
<th>Test IBM-BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chiang-sim</td>
<td>21.25</td>
<td>18.08</td>
</tr>
<tr>
<td>Pharaoh</td>
<td>23.2 → 22.0</td>
<td>19.3</td>
</tr>
<tr>
<td>SAMT</td>
<td>23.50</td>
<td>20.04</td>
</tr>
</tbody>
</table>

**Table:** Comparison of translation-models system using “SmartCase”, evaluated on the official case and punctuation sensitive IBM-BLEU metric. Note the change for the Pharaoh result. The initial result was run with the incorrect version of BLEU.
Example

@S_1-11: 从 这 可以 坐 机场 大巴 直接 去 机场

@S_1-10: 从 这 可以 坐 机场 大巴 直接 去 机场

@S_1-7: 从 这 可以 坐 机场 大巴

@S_1-1: 从 这 可以 坐 机场 大巴

@S/NP_4-5: 可以 坐

@NP_6-7: 机场 大巴

@PP_2-3: 从 这

to the airport
directly

@PP_9-10: 去 机场

@RB_8-8: 直接

you can take
the shuttle bus
from here
Decoding Time

- Decoding Time depends on the kind of rules extracted
- Two particularly problematic rules
  - Purely abstract - flattened full sentence structure
  - Target only - insertion of target words with no src
- Runtime in evaluation: 50 minutes for Dev 06
- Runtime w/o abstract/tgt only: 5 minutes for Dev 06
- End-to-end system available at www.cs.cmu.edu/zollmann/samt
Future Work

- Shift emphasis to speech-translation task
- Evaluate parsing for ASR output
- Directly decode ASR lattices