A Syntax-based Statistical Machine Translation Model

Introduction

The model

Experiment

Conclusion
- Statistical Translation Model (STM):
  - mathematical model
  - statistical modelling of human-language translation
  - parameters estimated with training corpus
First steps

- IBM 1998: string-to-string word-based translation

- completely independent process
Question: Why not word-based translation?

- no structural or syntactic aspects

- how to handle different word order?
Solution:

A Syntax-based Statistical Translation Model
- Input: parse tree (syntactic parser)

- Output: string

*kare ha ongaku wo kiku no ga daisuki desu*
Three operations on each node of the tree:

- reorder
- inserting
- translating
- Reorder
  - different word order
  - English vs. Japanese
- Word-insertion
  - e.g. capture linguistic differences in specifying syntactic cases
■ Translation

- Translate leaf words into the destination language
Output

A Syntax-based Statistical Translation Model

Reading off Leaves

```
kare ha ongaku wo kiku no ga daisuki desu
```

```
  VB
    +--- PRP
        +--- kare
        +--- ha
    +--- VB2
        +--- TO
            +--- kiku
            +--- no
    +--- VB1
        +--- ga
            +--- daisuki
            +--- desu
    +--- ONGAKU
        +--- TO
            +--- wo
```
Introduction

The model

Experiment

Conclusion
- English parse tree into a noisy channel
- Output should be a Japanese sentence
- Stochastical operations an each nodes
- **Reorder**
  - N! possible reorderings for N children
  - probability given by the *r-table*

<table>
<thead>
<tr>
<th>original order</th>
<th>reordering</th>
<th>P(reorder)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRP VB1 VB2</td>
<td>PRP VB1 VB2</td>
<td>0.074</td>
</tr>
<tr>
<td></td>
<td>PRP VB2 VB1</td>
<td>0.723</td>
</tr>
<tr>
<td></td>
<td>VB1 PRP VB2</td>
<td>0.061</td>
</tr>
<tr>
<td></td>
<td>VB1 VB2 PRP</td>
<td>0.037</td>
</tr>
<tr>
<td></td>
<td>VB2 PRP VB1</td>
<td>0.083</td>
</tr>
<tr>
<td></td>
<td>VB2 VB1 PRP</td>
<td>0.021</td>
</tr>
<tr>
<td>VB TO</td>
<td>VB TO</td>
<td>0.251</td>
</tr>
<tr>
<td></td>
<td>TO VB</td>
<td>0.749</td>
</tr>
<tr>
<td>TO NN</td>
<td>TO NN</td>
<td>0.107</td>
</tr>
<tr>
<td></td>
<td>NN TO</td>
<td>0.893</td>
</tr>
</tbody>
</table>

*r-table*
- **Word-insertion**
  - left, right oder nowhere
  - probability given by the *n-table*

| parent node | TOP | VB | VB | VB | TO | TO | NN | ...
|-------------|-----|----|----|----|----|----|----|-----|
| VB          | 0.735 | 0.687 | 0.344 | 0.709 | 0.900 | 0.800 | ...
| VB          | 0.004 | 0.061 | 0.004 | 0.030 | 0.003 | 0.096 | ...
| PRP         | 0.260 | 0.252 | 0.652 | 0.261 | 0.007 | 0.104 | ...

<table>
<thead>
<tr>
<th>w</th>
<th>P(ins–w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ha</td>
<td>0.219</td>
</tr>
<tr>
<td>ta</td>
<td>0.131</td>
</tr>
<tr>
<td>wo</td>
<td>0.099</td>
</tr>
<tr>
<td>no</td>
<td>0.094</td>
</tr>
<tr>
<td>ni</td>
<td>0.080</td>
</tr>
<tr>
<td>te</td>
<td>0.078</td>
</tr>
<tr>
<td>ga</td>
<td>0.062</td>
</tr>
<tr>
<td>desu</td>
<td>0.0007</td>
</tr>
</tbody>
</table>
Translation

- dependent only on the word itself

- Probability given by the *t*-table

<table>
<thead>
<tr>
<th>E</th>
<th>adores</th>
<th>he</th>
<th>i</th>
<th>listening</th>
<th>music</th>
<th>to</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>daisuki</td>
<td>1.000</td>
<td>NULL</td>
<td>kiku</td>
<td>NULL</td>
<td>NULL</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>NULL</td>
<td>0.016</td>
<td>watasi</td>
<td>kii</td>
<td>NULL</td>
<td>NULL</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>nani</td>
<td>0.005</td>
<td>kare</td>
<td>mi</td>
<td>NULL</td>
<td>NULL</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>da</td>
<td>0.003</td>
<td>shi</td>
<td></td>
<td>NULL</td>
<td>NULL</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>shi</td>
<td>0.003</td>
<td>nani</td>
<td></td>
<td>NULL</td>
<td>NULL</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>i</td>
<td></td>
<td></td>
<td></td>
<td>NULL</td>
<td>NULL</td>
<td>...</td>
</tr>
</tbody>
</table>

*t*-table
Total probability

- product of the single operation probabilities

Tables

- English-Japanese training corpus
Formal Transcription

Input:
English parse tree $\mathcal{E}$ (in nodes $\varepsilon_1$, $\varepsilon_2$, $\ldots$ $\varepsilon_n$)

Output:
French sentence $\mathbf{f}$ (in words $f_1$, $f_2$, $\ldots$ $f_m$)
Probability getting $f$ for $\mathcal{E}$

$$P(f|\mathcal{E}) = \sum_{\theta: \text{Str}(\Theta(\mathcal{E}))=f} P(\theta|\mathcal{E})$$

Str($\Theta(\mathcal{E})$) is a sequence of leaf words
- \( P(\Theta|\mathcal{E}) \): Probability of a particular set of RVs in a parse tree

\[
P(\theta|\mathcal{E}) = P(\theta_1, \theta_2, \ldots, \theta_n|\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)
= \prod_{i=1}^{n} P(\theta_i|\varepsilon_i).
\]
RVs $\theta_i = \langle \nu_i, \rho_i, \tau_i \rangle$ independent on each other, but dependent on features of $\epsilon_i$.

$$
P(\theta_i | \epsilon_i) = P(\nu_i, \rho_i, \tau_i | \epsilon_i) \\
= P(\nu_i | \epsilon_i) P(\rho_i | \epsilon_i) P(\tau_i | \epsilon_i) \\
= P(\nu_i | N(\epsilon_i)) P(\rho_i | R(\epsilon_i)) P(\tau_i | T(\epsilon_i)) \\
= n(\nu_i | N(\epsilon_i)) r(\rho_i | R(\epsilon_i)) t(\tau_i | T(\epsilon_i))$$
Probability of getting a French sentence $f$
given an English parse tree $\mathcal{E}$

\[
P(f|\mathcal{E}) = \sum_{\theta : \text{Str}(\theta(\mathcal{E})) = f} P(\theta|\mathcal{E})
\]

\[
= \sum_{\theta : \text{Str}(\theta(\mathcal{E})) = f} \prod_{i=1}^{n} n(\nu_i|N(\varepsilon_i)) r(\rho_i|R(\varepsilon_i)) t(\tau_i|T(\varepsilon_i))
\]
- Automatic Estimation of model parameters

- update parameter to maximize the likelihood of the training corpus
Algorithm

1. Initialize probability tables
2. Reset counters
3. For each iteration the number of events are counted and weighted by the probability of events
4. Parameter re-estimated by the counts
Introduction

The model

Experiment

Conclusion
Experiment with small English-Japanese corpus

- 2121 translation sentence pairs

- taggers build the English parse trees

Comparison with IBM 5
Evaluation

- generate the most probable alignment of the training corpus (Viterbi)

- average score of the first 50 sentences
<table>
<thead>
<tr>
<th>Alignment Status</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment okay</td>
<td>1 point</td>
</tr>
<tr>
<td>Not sure</td>
<td>0.5 point</td>
</tr>
<tr>
<td>Alignment wrong</td>
<td>0 points</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Average score</th>
<th>Perfect sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Model</td>
<td>0.582</td>
<td>10</td>
</tr>
<tr>
<td>IBM Model 5</td>
<td>0.431</td>
<td>0</td>
</tr>
</tbody>
</table>
he adores listening to music

hypocrisy is abhorrent to them

he has unusual ability in English

he was ablaze with anger
- Perplexity

  - Our Model: 15.79
  - IBM Model 5: 9.84
- Introduction
- The model
- Experiment
- Conclusion
- Syntax-based translation model

- Statistical modelling the translation process

- Syntactic information for languages with different word order

- Better alignment results in an experiment
Outline – 2nd Part

- Problems of Tree to String
- Clone Operation
- Tree to Tree

- Phrasal Translation
- Translation System
Problem of Tree to String Model

- Not all re-orderings of terminal nodes are possible

- Constrains syntactic correspondence between native and foreign language
The Clone operation

- Insert a copy of a subtree at any point in the tree
- Delete original node
When to clone?

- Should a clone be inserted as child of $\mathcal{E}_j$

  $$P_{ins}(\text{clone} \mid \mathcal{E}_j)$$

- Decide which node should be cloned

  $$P_{clone}(\mathcal{E}_i \mid \text{clone} = 1) = \frac{P_{makeclone}(\mathcal{E}_i)}{\sum_k P_{makeclone}(\mathcal{E}_k)}$$

- Probability of cloning is independent of previous cloning operations
Example
Tree – to - Tree

- Syntactic trees for foreign and native language
- Output tree instead of output string
- Additional tree transformations
  - Single source node → Two target nodes
  - Two source nodes → Single target node
- New model: $P(T_b \mid T_a)$
Building the output tree

- At each level at the output tree:
  - Choose an elementary tree
  - Align the children of the elementary tree
- Translate the leaves
Elementary tree

- \( P_{\text{elem}}(t_a \mid \varepsilon_a \Rightarrow \text{children } (\varepsilon_a)) \)
- Means that two nodes can grouped together
- **Example:**
  - Nodes A and B are considered a elementary tree
Alignment of children

- All children of an elementary tree are aligned at once according to:

\[ P_{align} (\alpha \mid \epsilon_a \Rightarrow \text{children} (t_a)) \]

- Insertions and Deletions are also done in this step
Tree – to – Tree Clone

- Same reordering problems as Tree – to – String
- Cloning will now added to alignment step

\[ P_{ins}(\text{clone} \in \alpha \mid \varepsilon_a \Rightarrow \text{children}(t_a)) \]

\[ P_{\text{clone}}(\varepsilon_i \mid \text{clone} \in \alpha) = \frac{P_{\text{makeclone}}(\varepsilon_i)}{\sum_k P_{\text{makeclone}}(\varepsilon_k)} \]
## Parameter comparison

<table>
<thead>
<tr>
<th></th>
<th>Tree-to-String</th>
<th>Tree-to-Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>elementary tree grouping</strong></td>
<td>$P_{order}(ρ</td>
<td>ε \Rightarrow \text{children}(ε))$</td>
</tr>
<tr>
<td><strong>re-order</strong></td>
<td>$P_{ins}(\text{left}, \text{right}, \text{none}</td>
<td>ε)$</td>
</tr>
<tr>
<td><strong>insertion</strong></td>
<td>$P_t(f</td>
<td>ε)$</td>
</tr>
<tr>
<td><strong>lexical translation</strong></td>
<td></td>
<td>$P_t(f</td>
</tr>
<tr>
<td><strong>with cloning</strong></td>
<td>$P_{ins}(\text{clone}</td>
<td>ε)$</td>
</tr>
<tr>
<td></td>
<td>$P_{makeclone}(ε)$</td>
<td>$P_{makeclone}(ε)$</td>
</tr>
</tbody>
</table>
Experiments

- Data from Korean – English corpus (Military domain)
- Korean suffixes often carry meaning
  - This suffixes became leaves in the syntax tree
  - Vocabulary was reduced from 10059 to 3279
- Average Korean: 13 words and 21 tokens
- Average English: 16 words
## Results

<table>
<thead>
<tr>
<th></th>
<th>Alignment Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Model 1</td>
<td>.37</td>
</tr>
<tr>
<td>IBM Model 2</td>
<td>.35</td>
</tr>
<tr>
<td>IBM Model 3</td>
<td>.43</td>
</tr>
<tr>
<td>Tree-to-String</td>
<td>.42</td>
</tr>
<tr>
<td>Tree-to-String, Clone</td>
<td>.36</td>
</tr>
<tr>
<td>Tree-to-String, Clone $P_{ins} = .5$</td>
<td>.32</td>
</tr>
<tr>
<td>Tree-to-Tree</td>
<td>.49</td>
</tr>
<tr>
<td>Tree-to-Tree, Clone</td>
<td>.36</td>
</tr>
</tbody>
</table>

\[
AER = 1 - \frac{2 | A \cap G |}{| A | + | G |}
\]

Word pairings by System (A) and Gold Standard(G)
Phrasal Translation

- 1 to 1 word translation not perfect
- Compound nouns
  - German vs English
- Idiomatic phrases
  - “To kick the bucket” vs “Den Löffel abgeben”
Model

- **1 to 1 Model:** \( t(\tau \mid T) = t(f \mid e) \)
- **1 to N with fertility \( \mu \):**
  \[
  t(\tau \mid T) = t(f_1 f_2 \ldots f_l \mid e) = \mu(l \mid e) \prod_{i=1}^{l} t(f_i \mid e)
  \]
- **N to N:**
  \[
  ph(\phi \mid \Phi) = t(f_1 f_2 \ldots f_l \mid e_1 e_2 \ldots e_m) = \\
  \mu(l \mid e_1 e_2 \ldots e_m) \prod_{i=1}^{l} t(f_i \mid e_1 e_2 \ldots e_m)
  \]
Incorporation into TM

\[ P(\theta_i \mid \varepsilon_i) = \lambda_{\Phi_i} p_h(\phi_i \mid \Phi_i) + (1 - \lambda_{\Phi_i}) r(\rho_i \mid R_i) n(\nu_i \mid N_i) \]

- R: feature for reordering
- N: feature for insertion
- \( \rho \): Reorder operation
- \( \nu \): Insert operation
From Models to the Translation

- Task: Translate from Foreign to English

\[ P(E \mid F) \]

- Reformulation

\[ P(E \mid F) = P(F \mid E)P(E) \]
Language Model

- Probability of an English Sentence $P(E)$
- N-gram LM in original Implementation
- No syntactic Information used
- Improvement: Immediate-head parsing for language models
Immediate-head parsing

- English Sentence
- Non-lexical PCFG $\rightarrow$ Large Parse Forest
- Pruning of the Large Parse Forest
  - Which edges have high probability of being correct?
- Evaluation of pruned Parse with a lexical PCFG
Decoder

- Decoder works in reverse direction to TM
- Find most probable syntactic tree E from a Sentence F
- Basic idea: “Translate“ Parse tree using TM to the foreign language
- Parse the foreign sentence
- “Translate” back to English
- And check LM
Conclusion

- Improvements to Syntactic Translation Model
  - Tree to String Clone
  - Tree to Tree
  - Tree to Tree Clone
  - Phrasal Translation

- Brief overview over Translation Process
References

- A Syntax-based Statistical Translation Model
  Kenji Yamada and Kevin Knight, Proceedings of the Conference of the Association for Computational Linguistics 2001

- Loosely Tree-based Alignment for Machine Translation

- Syntax-based Language Models for Statistical Machine Translation

- A Decoder for Syntax-Based Statistical MT
  Kenji Ymada and Kevin Knight, Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02), 2002
Thank you for your attention!