Example-based Decoding for Statistical Machine Translation

Taro Watanabe† and Eiichiro Sumita
taro.watanabe@atr.co.jp

ATR Spoken Language Translation Research Laboratories
Overview

- Statistical Machine Translation (SMT)
- Decoding Problem in SMT
- Example-based Decoder
- Experiments
- Summary
Statistical Machine Translation

- Translation from $J$ into $E$ (Berger et al. 1993)

\[
\hat{E} = \arg\max_E P(E|J) = \arg\max_E P(E)P(J|E)
\]
Statistical Machine Translation

- Translation from $J$ into $E$ (Berger et al. 1993)

\[
\hat{E} = \arg \max_E P(E|J) \\
= \arg \max_E P(E)P(J|E)
\]

Translation Model

$P(J|E)$

Language Model

$P(E)$

Decoder

$\arg \max_E P(J|E)P(E)$

$J$ $\rightarrow$ \hat{E} $\leftarrow$
Word Alignment Based Statistical Translation

\[ P(J|E) = \sum_A P(J, A|E) \]

\( E = \text{NULL}_0 \text{show}_1 \text{me}_2 \text{the}_3 \text{one}_4 \text{in}_5 \text{the}_6 \text{window}_7 \)

\( J = \text{uindo}_1 \text{no}_2 \text{shinamono}_3 \text{mie}_4 \text{mise}_5 \text{tekudasai}_6 \)

\( A = (7 \ 0 \ 4 \ 0 \ 1 \ 1) \)
Word Alignment Based Statistical Translation

\[ P(J|E) = \sum_A P(J, A|E) \]

\( E = \text{NULL}_0 \text{ show}_1 \text{ me}_2 \text{ the}_3 \text{ one}_4 \text{ in}_5 \text{ the}_6 \text{ window}_7 \)

\( J = \text{uindo}_1 \text{ no}_2 \text{ shinamono}_3 \text{ o}_4 \text{ mise}_5 \text{ tekudasai}_6 \)

\( A = (7 \ 0 \ 4 \ 0 \ 1 \ 1) \)

- Generative Process of \( P(J, A|E) \)
Word-by-Word (or Phrase-by-Phrase) Decoding

- Prefix of partial translation with score by TM and LM
- Breadth first search to extend the prefix
Word-by-Word (or Phrase-by-Phrase) Decoding

- Prefix of partial translation with score by TM and LM
- Breadth first search to extend the prefix

input: 精算書 を 確認 させ てください
Word-by-Word (or Phrase-by-Phrase) Decoding

- Prefix of partial translation with score by TM and LM
- Breadth first search to extend the prefix

**input:** 精算書を確認させてください

{ }
Word-by-Word (or Phrase-by-Phrase) Decoding

- Prefix of partial translation with score by TM and LM
- Breadth first search to extend the prefix

**Input:** 精算書を確認させてください

```plaintext
{} let me
  i'd like
```
Word-by-Word (or Phrase-by-Phrase) Decoding

- Prefix of partial translation with score by TM and LM
- Breadth first search to extend the prefix

**input:** 精算書を確認させてください

\[ P( てください | \text{let me}) P(\text{let me}) \]

\[ \{ \} \rightarrow \text{let me} \]

\[ \text{i 'd like} \]

\[ P( てください | \text{i 'd like}) P(\text{i 'd like}) \]
Word-by-Word (or Phrase-by-Phrase) Decoding

- Prefix of partial translation with score by TM and LM
- Breadth first search to extend the prefix

**input**: 精算書を確認させてください

{ } → let me

{ } → i’d like

{ } → to check

{ } → to confirm
Word-by-Word (or Phrase-by-Phrase) Decoding

- Prefix of partial translation with score by TM and LM
- Breadth first search to extend the prefix

**input:** 精算書を確認させてください

```
{ } → let me → let me check
      ↓                         ↓
    i'd like → let me confirm
      ↓                         ↓
    to check → let me make sure
      ↓                         ↓
  i'd like to check → let me reconfirm
      ↓                         ↓
  i'd like to confirm → i'd like to make sure
```

MT Summit IX – p. 5/19
Decoding Problem in SMT

- Application to similar language pairs
  - Similar pairs: French–English, German–English, etc.
  - Distant pairs?: Japanese–English, Chinese-English, etc.
Decoding Problem in SMT

- Application to similar language pairs
  - Similar pairs: French–English, German–English, etc.
  - Distant pairs? : Japanese–English, Chinese-English, etc.

- Frequent insertion/deletion and intricated word alignments
Decoding Problem in SMT

- Application to similar language pairs
  - Similar pairs: French–English, German–English, etc.
  - Distant pairs?: Japanese–English, Chinese-English, etc.

- Frequent insertion/deletion and intricated word alignments

- Word-by-word or phrase-by-phrase decoding
  - Frequent insertion/deletion — huge search space
  - Pruning is inevitable — search error
Example-based Decoder

Input: $J_0$
Example-based Decoder

Input: $J_0$

Examples

$$(J_1, E_1) : J_1 \approx J_0$$
$$(J_2, E_2) : J_2 \approx J_0$$
...

Bilingual Corpus

$$(J_1, E_1)$$
$$(J_2, E_2)$$
$$(J_3, E_3)$$
...

Example-based Decoder

Input: $J_0$

Examples

$(J_1, E_1) : J_1 \approx J_0$
$(J_2, E_2) : J_2 \approx J_0$
...

Bilingual Corpus

$(J_1, E_1)$
$(J_2, E_2)$
$(J_3, E_3)$
...

Viterbi Alignments

$(J_0, A_1, E_1)$
$(J_0, A_2, E_2)$
...

Translation Model & Language Model

$P(J|E)P(E)$
Example-based Decoder

Input: $J_0$

Examples

$(J_1, E_1) : J_1 \approx J_0$
$(J_2, E_2) : J_2 \approx J_0$
...

Viterbi Alignments

$(J_0, A_1, E_1)$
$(J_0, A_2, E_2)$
...

Bilingual Corpus

$(J_1, E_1)$
$(J_2, E_2)$
$(J_3, E_3)$
...

Translation Model & Language Model

$P(J|E)P(E)$

Greedy Search

$(J_0, A_1, E_1) \rightarrow (J_0, A'_1, E'_1)$
$(J_0, A_2, E_2) \rightarrow (J_0, A'_2, E'_2)$
...

Output: $\hat{E}$
Example-based Decoder

Input: $J_0$

Examples

$\begin{align*}
(J_1, E_1) : J_1 &\approx J_0 \\
(J_2, E_2) &\approx J_0 \\
&\vdots
\end{align*}$

Retrieval

Bilingual Corpus

$\begin{align*}
(J_1, E_1) \\
(J_2, E_2) \\
(J_3, E_3) &\vdots
\end{align*}$

Translation Model & Language Model

$P(J|E)P(E)$

Modification

$\begin{align*}
(J_0, A_1, E_1) &\rightarrow (J_0, A'_1, E'_1) \\
(J_0, A_2, E_2) &\rightarrow (J_0, A'_2, E'_2) \\
&\vdots
\end{align*}$

Greedy Search

Output: $\hat{E}$
Retrieval of Examples

- Similarity measure
  - Edit distance criteria

\[
\text{dis}(J_k, J_0) = I(J_k, J_0) + D(J_k, J_0) + S(J_k, J_0)
\]

\[
I(J_k, J_0), D(J_k, J_0), S(J_k, J_0) = \text{# of ins/del/sub}
\]
Retrieval of Examples

- **Similarity measure**
  - **Edit distance criteria**
    \[
    dis(J_k, J_0) = I(J_k, J_0) + D(J_k, J_0) + S(J_k, J_0)
    \]
    \[
    I(J_k, J_0), D(J_k, J_0), S(J_k, J_0) = \# \text{ of ins/del/sub}
    \]
  - **tf/idf criteria:** one translation pair = one document
    \[
    P_{tf/idf}(J_k, J_0) = \sum_{i: J_0, i \in J_k} \frac{\log(N/df(J_{0,i}))}{\log N} / |J_0|
    \]
Retrieval of Examples

- Similarity measure
  - Edit distance criteria
    \[
    dis(J_k, J_0) = I(J_k, J_0) + D(J_k, J_0) + S(J_k, J_0)
    \]
    \[
    I(J_k, J_0), D(J_k, J_0), S(J_k, J_0) = \# \text{ of ins/del/sub}
    \]
  - \(tf/idf\) criteria: one translation pair = one document
    \[
    P_{tf/idf}(J_k, J_0) = \sum_{i: J_0, i \in J_k} \frac{\log(N/df(J_0,i))}{\log N} / |J_0|
    \]

- Score function
  \[
  score(J_k, J_0) = \begin{cases} 
  (1.0 - \alpha)(1.0 - \frac{dis(J_k, J_0)}{|J_0|}) & \text{dis}(J_k, J_0) > 0 \\
  +\alpha P_{tf/idf}(J_k, J_0) & \text{otherwise} \\
  1.0 & \text{otherwise}
  \end{cases}
  \]
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\)
  2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\).
  2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

- Hill-climbing operators (Germann et al. 2001)
  - Translate words
  - Translate and align words
  - Translate and insert words
  - Move alignments
  - Swap segments
  - Remove words
  - Join words

NULL please let me make it sure

精算書を確認させてください
Modification of Examples

For each translation example \((J_k, E_k)\),

1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\).
2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

Hill-climbing operators (Germann et al. 2001)

- Translate words
- Translate and align words
- Translate and insert words
- Move alignments
- Swap segments
- Remove words
- Join words

Hill-climbing operators:
- Translate
- Check
- See
- Confirm
- ...
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\).
  2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

- Hill-climbing operators (Germann et al. 2001)
  - Translate words
  - Translate and align words
  - Translate and insert words
  - Move alignments
  - Swap segments
  - Remove words
  - Join words
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\)
  2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

- Hill-climbing operators (Germann et al. 2001)
  ♦ Translate words
  ♦ Translate and align words
  ♦ Translate and insert words
  ♦ Move alignments
  ♦ Swap segments
  ♦ Remove words
  ♦ Join words
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\)
  2. Perform greedy decoding algorithm thorough hill-climbing for 
     \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

- Hill-climbing operators (Germann et al. 2001)
  - Translate words
  - Translate and align words
  - Translate and insert words
  - Move alignments
  - Swap segments
  - Remove words
  - Join words

NULL please let me make it sure
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\)
  2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

- Hill-climbing operators (Germann et al. 2001)
  - Translate words
  - Translate and align words
  - Translate and insert words
  - Move alignments
  - Swap segments
  - Remove words
  - Join words

NULL please let me make it sure
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\).
  2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

- Hill-climbing operators (Germann et al. 2001)
  - Translate words
  - Translate and align words
  - Translate and insert words
  - Move alignments
  - Swap segments
  - Remove words
  - Join words

NULL please let me make it sure
Modification of Examples

- For each translation example \((J_k, E_k)\),
  1. Compute the viterbi alignment \(A_k\) for the pair \((J_0, E_k)\).
  2. Perform greedy decoding algorithm thorough hill-climbing for \((J_0, A_k, E_k)\) to obtain \((J_0, A'_k, E'_k)\) by modifying \(A_k\) and \(E_k\).

- Hill-climbing operators (Germann et al. 2001)
  - Translate words
  - Translate and align words
  - Translate and insert words
  - Move alignments
  - Swap segments
  - Remove words
  - Join words

Hill-climbing operators (Germann et al. 2001)
Example of Greedy Process

NULL what’s the fastest way to get there

この 小包 を 日本 に 送り たい の ですが 一 番 遅い 方法 は何 ですか
Example of Greedy Process

NULL what’s the fastest way to get there

this package to Japan to send it there

Translate and insert words (get → send and insert it)
Example of Greedy Process

```
null what's the fastest way to get there

this small box to Japan does not want a quick way of how can

Translate and insert words (get → send and insert it)

null what's the fastest way to send it there

this small box to Japan does not want a quick way of how can

Translate and insert words (there → Japan and insert to)

null what's the fastest way to send it to Japan

this small box to Japan does not want a quick way of how can
```
Example of Greedy Process

NULL what’s the fastest way to get there

この 小包 を 日本 に 送りたい の です が 一番速い方法は何ですか

Translate and insert words (get → send and insert it)

NULL what’s the fastest way to send it there

この 小包 を 日本 に 送りたい の です が 一番速い方法は何ですか

Translate and insert words (there → japan and insert to)

NULL what’s the fastest way to send it to japan

この 小包 を 日本 に 送りたい の です が 一番速い方法は何ですか

Translate and align words (it → parcel and insert this)

NULL what’s the fastest way to send this parcel to japan

この 小包 を 日本 に 送りたい の です が 一番速い方法は何ですか
Example of Greedy Process

NULL what’s the fastest way to get there

この 小包 を 日本 に 送り たい の で す が 一 番 遽い 方法 は 何 で すか

Translate and insert words (get → send and insert it)

NULL what’s the fastest way to send it there

この 小包 を 日本 に 送り たい の で す が 一 番 遽い 方法 は 何 で すか

Translate and insert words (there → japan and insert to)

NULL what’s the fastest way to send it to japan

この 小包 を 日本 に 送り たい の で す が 一 番 遽い 方法 は 何 で すか

Translate and align words (it → parcel and insert this)

NULL what’s the fastest way to send this parcel to japan

この 小包 を 日本 に 送り たい の で す が 一 番 遽い 方法 は 何 で すか

Move alignments (the → this)

NULL what’s the fastest way to send this parcel to japan

この 小包 を 日本 に 送り たい の で す が 一 番 遽い 方法 は 何 で すか
## Experiments

### Basic Travel Expression Corpus

<table>
<thead>
<tr>
<th></th>
<th>Chinese</th>
<th>English</th>
<th>Japanese</th>
<th>Korean</th>
</tr>
</thead>
<tbody>
<tr>
<td># of sentences</td>
<td>167,163</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of words</td>
<td>956,732</td>
<td>980,790</td>
<td>1,148,428</td>
<td>1,269,888</td>
</tr>
<tr>
<td>vocabulary size</td>
<td>16,411</td>
<td>15,641</td>
<td>21,896</td>
<td>13,395</td>
</tr>
<tr>
<td># of singletons</td>
<td>5,207</td>
<td>5,547</td>
<td>9,220</td>
<td>4,191</td>
</tr>
<tr>
<td>3-gram perplexity</td>
<td>45.53</td>
<td>35.35</td>
<td>24.06</td>
<td>20.34</td>
</tr>
</tbody>
</table>
Two decoders on IBM Model 4 + 3-gram:

- Left-to-right generation word-by-word beam search decoder
- Example-based decoder

Evaluation Metrics

WER: Word-error-rate — edit distance penalty
PER: Position independent WER — WER but ignore positional disfluencies
BLEU: Geometric mean of n-gram precision
SE: Subjective evaluation — ranks ranging from
  A: perfect, B: fair, C: acceptable and D: non-sense judged by a native speaker
SER: Search error rate — judged by TM+LM scores of outputs from two systems

Remarks: 16 references for WER, PER and BLEU
## Results

<table>
<thead>
<tr>
<th></th>
<th>Exact [%]</th>
<th>WER [%]</th>
<th>PER [%]</th>
<th>BLEU [%]</th>
<th>SE [%]</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A+B</td>
<td>A+B+C</td>
<td></td>
</tr>
<tr>
<td>C-E</td>
<td></td>
<td>45.0/34.3</td>
<td>39.8/30.3</td>
<td>43.6/56.7</td>
<td>48.4/65.3</td>
<td>65.9/76.9</td>
<td>71.4/81.0</td>
<td></td>
</tr>
<tr>
<td>C-J</td>
<td>52.7</td>
<td>35.7/25.5</td>
<td>31.3/22.6</td>
<td>56.9/67.8</td>
<td>50.8/69.0</td>
<td>59.4/74.3</td>
<td>66.9/80.2</td>
<td></td>
</tr>
<tr>
<td>C-K</td>
<td></td>
<td>38.4/29.1</td>
<td>34.2/26.2</td>
<td>56.1/65.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>E-C</td>
<td></td>
<td>45.0/38.0</td>
<td>39.7/33.4</td>
<td>42.1/51.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>E-J</td>
<td>40.8</td>
<td>34.2/29.0</td>
<td>30.5/26.1</td>
<td>59.2/65.7</td>
<td>55.8/65.1</td>
<td>62.4/71.6</td>
<td>70.2/77.8</td>
<td></td>
</tr>
<tr>
<td>E-K</td>
<td></td>
<td>38.7/35.6</td>
<td>34.3/31.6</td>
<td>57.3/61.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>J-C</td>
<td></td>
<td>46.8/33.0</td>
<td>38.9/27.8</td>
<td>39.7/57.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>J-E</td>
<td>33.7</td>
<td>42.9/35.0</td>
<td>37.4/30.3</td>
<td>47.6/57.4</td>
<td>50.8/63.7</td>
<td>65.7/74.5</td>
<td>70.2/77.6</td>
<td></td>
</tr>
<tr>
<td>J-K</td>
<td></td>
<td>27.7/20.8</td>
<td>25.4/19.2</td>
<td>67.2/73.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>K-C</td>
<td></td>
<td>41.9/32.9</td>
<td>34.4/27.6</td>
<td>45.1/55.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>K-E</td>
<td>39.2</td>
<td>45.1/36.4</td>
<td>38.5/32.1</td>
<td>44.3/56.8</td>
<td>49.2/61.6</td>
<td>65.7/72.9</td>
<td>72.2/78.4</td>
<td></td>
</tr>
<tr>
<td>K-J</td>
<td></td>
<td>26.8/20.8</td>
<td>24.6/19.3</td>
<td>64.3/70.8</td>
<td>56.5/69.2</td>
<td>66.5/77.5</td>
<td>78.4/84.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exact [%]</td>
<td>WER [%]</td>
<td>PER [%]</td>
<td>BLEU [%]</td>
<td>A</td>
<td>A+B</td>
<td>A+B+C</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>----------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>C-E</td>
<td></td>
<td>45.0/34.3</td>
<td>39.8/30.3</td>
<td>43.6/56.7</td>
<td>48.4/65.3</td>
<td>65.9/76.9</td>
<td>71.4/81.0</td>
<td></td>
</tr>
<tr>
<td>C-J</td>
<td>52.7</td>
<td>35.7/25.5</td>
<td>31.3/22.6</td>
<td>56.9/67.8</td>
<td>50.8/69.0</td>
<td>59.4/74.3</td>
<td>66.9/80.2</td>
<td></td>
</tr>
<tr>
<td>C-K</td>
<td></td>
<td>38.4/29.1</td>
<td>34.2/26.2</td>
<td>56.1/65.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-C</td>
<td></td>
<td>45.0/38.0</td>
<td>39.7/33.4</td>
<td>42.1/51.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-J</td>
<td>40.8</td>
<td>34.2/29.0</td>
<td>30.5/26.1</td>
<td>59.2/65.7</td>
<td>55.8/65.1</td>
<td>62.4/71.6</td>
<td>70.2/77.8</td>
<td></td>
</tr>
<tr>
<td>E-K</td>
<td></td>
<td>38.7/35.6</td>
<td>34.3/31.6</td>
<td>57.3/61.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J-C</td>
<td></td>
<td>46.8/33.0</td>
<td>38.9/27.8</td>
<td>39.7/57.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J-E</td>
<td>33.7</td>
<td>42.9/35.0</td>
<td>37.4/30.3</td>
<td>47.6/57.4</td>
<td>50.8/63.7</td>
<td>65.7/74.5</td>
<td>70.2/77.6</td>
<td></td>
</tr>
<tr>
<td>J-K</td>
<td></td>
<td>27.7/20.8</td>
<td>25.4/19.2</td>
<td>67.2/73.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-C</td>
<td></td>
<td>41.9/32.9</td>
<td>34.4/27.6</td>
<td>45.1/55.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-E</td>
<td>39.2</td>
<td>45.1/36.4</td>
<td>38.5/32.1</td>
<td>44.3/56.8</td>
<td>49.2/61.6</td>
<td>65.7/72.9</td>
<td>72.2/78.4</td>
<td></td>
</tr>
<tr>
<td>K-J</td>
<td></td>
<td>26.8/20.8</td>
<td>24.6/19.3</td>
<td>64.3/70.8</td>
<td>56.5/69.2</td>
<td>66.5/77.5</td>
<td>78.4/84.7</td>
<td></td>
</tr>
</tbody>
</table>

beam search/example-based decoder
## Detailed Results

<table>
<thead>
<tr>
<th></th>
<th>SE [%]</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exactly matched</td>
<td></td>
<td></td>
<td></td>
<td>otherwise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>A+B</td>
<td>A+B+C</td>
<td></td>
<td>A</td>
<td>A+B</td>
<td>A+B+C</td>
<td></td>
<td>A</td>
<td>A+B</td>
<td>A+B+C</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>C-E</td>
<td>65.4/92.6</td>
<td>78.4/97.0</td>
<td>82.5/97.8</td>
<td></td>
<td>29.5/34.9</td>
<td>51.9/54.4</td>
<td>58.9/62.2</td>
<td></td>
<td>J-E</td>
<td>72.1/97.1</td>
<td>80.8/99.4</td>
<td>83.7/99.4</td>
<td></td>
</tr>
<tr>
<td>J-E</td>
<td>69.0/92.0</td>
<td>81.0/97.5</td>
<td>85.0/98.0</td>
<td></td>
<td>36.5/41.9</td>
<td>55.8/57.1</td>
<td>63.9/65.8</td>
<td></td>
<td>K-E</td>
<td>65.4/97.0</td>
<td>73.6/98.1</td>
<td>78.4/98.9</td>
<td></td>
</tr>
<tr>
<td>K-E</td>
<td>79.3/95.2</td>
<td>81.7/98.1</td>
<td>84.6/98.6</td>
<td></td>
<td>39.4/44.4</td>
<td>49.0/53.3</td>
<td>60.3/63.6</td>
<td></td>
<td>K-J</td>
<td>74.5/98.5</td>
<td>80.0/99.5</td>
<td>87.5/99.5</td>
<td></td>
</tr>
</tbody>
</table>

beam search/example-based decoder
<table>
<thead>
<tr>
<th></th>
<th>matched</th>
<th>non-matched</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-E</td>
<td>15.2/58.0</td>
<td>28.2/38.6</td>
<td>21.4/48.8</td>
</tr>
<tr>
<td>C-J</td>
<td>16.7/46.8</td>
<td>30.3/24.9</td>
<td>23.1/24.9</td>
</tr>
<tr>
<td>C-K</td>
<td>14.9/45.0</td>
<td>25.7/29.5</td>
<td>20.0/37.6</td>
</tr>
<tr>
<td>E-C</td>
<td>19.2/46.2</td>
<td>39.1/26.8</td>
<td>31.0/34.1</td>
</tr>
<tr>
<td>E-J</td>
<td>19.7/34.1</td>
<td>33.4/26.2</td>
<td>27.8/29.4</td>
</tr>
<tr>
<td>E-K</td>
<td>14.4/42.8</td>
<td>31.8/32.8</td>
<td>24.7/36.9</td>
</tr>
<tr>
<td>J-C</td>
<td>10.5/63.4</td>
<td>37.6/31.4</td>
<td>28.4/42.2</td>
</tr>
<tr>
<td>J-E</td>
<td>16.3/51.7</td>
<td>35.8/28.1</td>
<td>29.2/36.1</td>
</tr>
<tr>
<td>J-K</td>
<td>14.5/39.5</td>
<td>50.3/10.9</td>
<td>38.2/20.6</td>
</tr>
<tr>
<td>K-C</td>
<td>14.0/51.5</td>
<td>31.9/28.7</td>
<td>24.9/37.6</td>
</tr>
<tr>
<td>K-E</td>
<td>19.5/42.5</td>
<td>31.3/31.9</td>
<td>26.7/36.1</td>
</tr>
<tr>
<td>K-J</td>
<td>15.0/32.5</td>
<td>27.7/12.3</td>
<td>22.7/20.2</td>
</tr>
</tbody>
</table>

beam search/example-based decoder
## Search Error

<table>
<thead>
<tr>
<th></th>
<th>matched</th>
<th>non-matched</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-E</td>
<td>15.2/58.0</td>
<td>28.2/38.6</td>
<td>21.4/48.8</td>
</tr>
<tr>
<td>C-J</td>
<td>16.7/46.8</td>
<td>30.3/24.9</td>
<td>23.1/24.9</td>
</tr>
<tr>
<td>C-K</td>
<td>14.9/45.0</td>
<td>25.7/29.5</td>
<td>20.0/37.6</td>
</tr>
<tr>
<td>E-C</td>
<td>19.2/46.2</td>
<td>39.1/26.8</td>
<td>31.0/34.1</td>
</tr>
<tr>
<td>E-J</td>
<td>19.7/34.1</td>
<td>33.4/26.2</td>
<td>27.8/29.4</td>
</tr>
<tr>
<td>E-K</td>
<td>14.4/42.8</td>
<td>31.8/32.8</td>
<td>24.7/36.9</td>
</tr>
<tr>
<td>J-C</td>
<td>10.5/63.4</td>
<td>37.6/31.4</td>
<td>28.4/42.2</td>
</tr>
<tr>
<td>J-E</td>
<td>16.3/51.7</td>
<td>35.8/28.1</td>
<td>29.2/36.1</td>
</tr>
<tr>
<td>J-K</td>
<td>14.5/39.5</td>
<td>50.3/10.9</td>
<td>38.2/20.6</td>
</tr>
<tr>
<td>K-C</td>
<td>14.0/51.5</td>
<td>31.9/28.7</td>
<td>24.9/37.6</td>
</tr>
<tr>
<td>K-E</td>
<td>19.5/42.5</td>
<td>31.3/31.9</td>
<td>26.7/36.1</td>
</tr>
<tr>
<td>K-J</td>
<td>15.0/32.5</td>
<td>27.7/12.3</td>
<td>22.7/20.2</td>
</tr>
</tbody>
</table>

beam search/example-based decoder
### Some Examples of Translations

| input: |
|---|---|
| 銀行の前でバッグをひっかかれました |

<table>
<thead>
<tr>
<th>reference:</th>
</tr>
</thead>
<tbody>
<tr>
<td>I was robbed of my bag in front of the bank</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>beam:</th>
</tr>
</thead>
<tbody>
<tr>
<td>My bag was stolen in the front of the bank</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>I was robbed of my bag in front of the bank</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>retrieved:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I was robbed of my wallet bank on the subway)</td>
</tr>
</tbody>
</table>

| input: |
|---|---|
| 今調べておりますのでしばらくお待ちください |

<table>
<thead>
<tr>
<th>reference:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you wait for a moment while I check</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>beam:</th>
</tr>
</thead>
<tbody>
<tr>
<td>I’m out so please wait a moment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you wait a moment while we check</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>retrieved:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Would you wait for a moment while we check)</td>
</tr>
</tbody>
</table>

| input: |
|---|---|
| お金を入れたのに機械が動きませんお金を返してもらいたいのですが |

<table>
<thead>
<tr>
<th>reference:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Even though I put in some money the machine doesn’t work I’d like a refund please</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>beam:</th>
</tr>
</thead>
<tbody>
<tr>
<td>I’d like a refund for this machine doesn’t money got heading</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Although I put in some coins the machine didn’t work I’d like a refund</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>retrieved:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Although I put in some coins the machine didn’t work I’d like a refund)</td>
</tr>
</tbody>
</table>
### Some Examples of Translations 2

| input:  | 十時 に ここ で 待って いてくれる ん です ね  |
| reference: | you will wait for me here at ten right |
| beam: | here is your pay wait ten o’clock |
| example: | are you waiting here at ten o’clock |
| retrieved: | (why are you waiting here) |

| input:  | いいえ 赤い バック の 隣 に ある もの です  |
| reference: | no the one next to the red bag |
| beam: | no red bag in the room next door is something |
| example: | no it ’s next to the red one |
| retrieved: | (it ’s next to that building) |

| input:  | ええ それ と 似 た 手帳 が 届いて います  |
| reference: | yes someone did turn in a notebook like that |
| beam: | yes it is similar there any messages for me |
| example: | i have a notebook come yes it is similar |
| retrieved: | (yes we have a japanese speaking guide) |
Discussion

- **Example-based decoder**
  - A method to merge example-based framework and statistical machine translation
    - Retrieve-and-tweak strategy
    - Retrieval of examples — edit distance + $tf/idf$
    - Modification of examples — greedy method, but uses retrieved examples as the initial condition
  - Very strong bias to guide the search, especially suitable for long distance language pairs
Related Work: Memory-based SMT (Marcu, 2001)

- Extract phrase translation pattern
- The greedy decoding process is initiated from the concatenation of phrases found in the translation memory
- Difference in unit: Phrase vs. Sentence
Related Work: Memory-based SMT (Marcu, 2001)

- Extract phrase translation pattern
- The greedy decoding process is initiated from the concatenation of phrases found in the translation memory
- Difference in unit: Phrase vs. Sentence

Better model, better translation?

- Some correlation between the probability assigned by the models and the translation quality (Akiba et al. 2002)
Discussion, Contd.

- Related Work: Memory-based SMT (Marcu, 2001)
  - Extract phrase translation pattern
  - The greedy decoding process is initiated from the concatenation of phrases found in the translation memory
  - Difference in unit: Phrase vs. Sentence

- Better model, better translation?
  - Some correlation between the probability assigned by the models and the translation quality (Akiba et al. 2002)

- Future Works
  - Post-translation
  - Chunk-based (or Syntax-based) statistical translation model