Experiments in English↔Japanese Tree-to-String Machine Translation

Graham Neubig
Nara Institute of Science and Technology
10/20/2012
Introduction/Motivation
Translation Models

**String**

he visited the white house

**Tree (Phrase Structure)**

he visited the white house

**Dependency**

he visited the white house

**String**

彼はホワイトハウスを訪問した

**Tree (Phrase Structure)**

彼はホワイトハウスを訪問した

**Dependency**

彼はホワイトハウスを訪問した
Recent Usage in English ↔ Japanese

- Phrase-based translation [Koehn+ 03] is still popular

English: he visited the white house

Japanese: 彼はホワイトハウスを訪問した

- Moses used in 25 papers at NLP2012
- Also, hierarchical phrase-based translation [Chiang 07] ([Feng+ 11] is one of the few examples)
Recent Usage in English ↔ Japanese

- Pre-ordering [Xia+ 04] is another popular technique

Source Dependencies:

Pre-ordering:

Translation:

• First used for Japanese by [Komachi+ 06]?

• Used by Google [Xu+ 09], NTT [Isozaki+ 11], others [Nguyen+ 08, Neubig+ 12]
Recent Usage in English ↔ Japanese

- Dependency-to-dependency used by Kyoto U [Nakazawa+ 06] and rule based systems

\[
\text{he visited the white house} \quad \rightarrow \quad \text{彼はホワイトハウスを訪問した}
\]
Recent Usage in English ↔ Japanese

- String-to-tree models [Yamada+ 01] used by NTT in NTCIR task [Sudoh+ 11]
Recent Usage in English ↔ Japanese

(string) he visited the white house

(tree (phrase structure))

(dependency) he visited the white house

(H)PBMT

(Pre-ordering) S2T

(D2D) dependency

(string) 彼はホワイトハウスを訪問した

(tree (phrase structure))

(dependency) 彼はホワイトハウスを訪問した
What about Tree-driven Models?!

**string**
he visited the white house

**tree (phrase structure)**
```
S
  VP
    NP
      PRP VBD DT NNP NNP
```
he visited the white house

**dependency**
```
dobj
  nsubj
    det
      n
```
he visited the white house

**string**
彼はホワイトハウスを訪問した

**tree (phrase structure)**
```
S
  VP
    PP
      NP
        N P N N P N V
```
彼はホワイトハウスを訪問した

**dependency**
```
subj
dobj
  n
```
彼はホワイトハウスを訪問した
Tree-to-String Models [Liu+ 06]

Experiments in English-Japanese Tree-to-String Machine Translation
Dependency-to-String Models
[Quirk+ 05]
T2S/D2S vs Phrase Based

- **Better reordering** through use of syntactic structure
- **Very fast!** (especially compared to HPBMT)
- Better lexical choice because long-range context considered (especially D2S)
- Requires a parser
- Sensitive to parse errors
T2S/D2S vs Pre-ordering

• + T2S/D2S **jointly searches** for reordering and translation

• + T2S/D2S can easily handle **lexicalized reordering**

- Pre-ordering can find translation rules that **overlap constituent boundaries**

```
X が 高い
```

- X is high

```
X が 好き
```

- likes X
T2S vs. D2S

- T2S: Can handle de-lexicalized rules = more general?

- D2S: Dependent words are close → good for lexical choice?
Experiments and Summary
Question:

How well do modern statistical tree-to-string methods work for English ↔ Japanese translation?
Previous Research

• Three examples for $\text{En} \rightarrow \text{Ja}$?
  
  • [Quirk+ 06] Uses dependency treelet translation and shows improvement over PBMT
  
  • [Wu+ 10] Uses HPSG input and shows improvement over Joshua (HPBMT)
  
  • [DeNero+ 11] Shows forest-to-string does slightly better than syntactic pre-ordering in terms of BLEU

• One example for $\text{Ja} \rightarrow \text{En}$?
  
  • [Menezes+ 05] Uses dependency treelet translation, no direct comparison to other methods
Experimental Setup

- **System:** In-house forest-to-string decoder “travataar”
  - Forest-to-string translation [Mi+ 08] with tree transducers
  - Alignment GIZA++, extraction GHKM, tuning MERT
- **Data:** Kyoto Free Translation Task (KFTT [Neubig 11]), ~350k sentences of Wikipedia data for training
- **Baseline:** Moses PBMT, PBMT + Preordering [Neubig+ 12]
- **Evaluation:** BLEU, RIBES, Acceptability (0-5)
Tree-to-String Settings (Explained in Detail Later)

- **Language Analysis:**
  - **En Parser:** Stanford, Berkeley, *Egret* (Tree, Forest)
  - **Ja:** Juman+KNP, MeCab+Cabocha, *KyTea*+EDA

- **Composed Rules:** 1, 2, 3, 4

- **Non-terminals:** 1, 2, 3

- **Binarization:** Left, **Right**

- **Null Attachment:** Top, Exhaustive (1, 2)

- **Tuning:** BLEU, RIBES, *(BLEU+RIBES)/2*
Summary (En-Ja)

**BLEU**
- PBMT: 19.5
- PBMT+Pre: 21.0
- F2S: 19.5

**RIBES**
- PBMT: 64.0
- PBMT+Pre: 67.0
- F2S: 68.0

**Acceptability**
- PBMT: 2.6
- PBMT+Pre: 2.8
- F2S: 3.2
Experiments in English-Japanese Tree-to-String Machine Translation

Summary (Ja-En)

- BLEU:
  - PBMT: 16.6
  - PBMT+Pre: 17
  - T2S: 16.6

- RIBES:
  - PBMT: 64.5
  - PBMT+Pre: 65
  - T2S: 65.5

- Acceptability:
  - PBMT: 2.6
  - PBMT+Pre: 3.2
  - T2S: 3

The diagrams illustrate the performance metrics for each method (PBMT, PBMT+Pre, T2S) in the context of BLEU, RIBES, and acceptability.
En-Ja F2S vs. PBMT+Pre

**Input:**
Department of Sociology in Faculty of Letters opened.

**PBMT+Pre:**
開業 年 文学 部 社会 学科。

**F2S:**
文学 部 社会 学科 を 開設。

Properly interprets noun phrase + verb
En-Ja F2S vs. PBMT+Pre

Input:
Afterwards it was reconstructed but its influence declined.

PBMT+Pre:
その後衰退したが、その影響を受けて再建されたものである。

F2S:
その後再建されていてが、影響力は衰えた。

Properly reconstructs relationship between two verb phrases
Experiments in English-Japanese Tree-to-String Machine Translation

En-Ja F2S vs. PBMT+Pre

**Input:**
Introduction of KANSAI THRU PASS Miyako Card

PBMT+Pre:
スルッと kansai 都 カード の 導入

F2S:
伝来 スルッと KANSAI 都 カード

Parsing error:
(NP (NP Introduction) (PP of KANSAI THRU PASS) (NP Miyako) (NP Card))
Ja-En T2S vs. PBMT+Pre

**Input:**
史実 に は 直接 の 関係 は な い。

**PBMT+Pre:**
in the historical fact is not directly related to it.

**T2S:**
is not directly related to the historical facts.

Properly translates “に は … 関係 が” as “related to”
Ja-En T2S vs. PBMT+Pre

Input:
九条 道家 は 嫡男 ・ 九条 教実 に 先立 た れ 、 次男 ・ 二 条 良実 は 事実 上 の 勘当 状態 に あ っ た。

PBMT+Pre:
michiie kujo was his eldest son and heir, norizane kujo, and his second son, yoshizane nijo was disinherited.

T2S:
michiie kujo to his legitimate son kujo norizane died before him, and the second son, nijo yoshizane was virtually disowned.

Much better division between clauses
Ja-En T2S vs. PBMT+Pre

Input:
日本 語 日本 文学 科
1474 年 ~ 1478 年 - 山名 政 豊

PBMT+Pre:
the department of japanese language and literature
in 1474 to 1478 - masatoyo yamana

T2S:
japanese language and literature
masatoyo yamana 1474 shokoku-ji in -

Errors due to more restrictive rule extraction (first example),
parse errors (second example, “Yamana” is a single noun phrase)
Effect of Language Analysis
Question:

How much do the language analysis tools used effect translation?
Language Analysis (En-Ja):

- Which parser provides better translations?
- **Stanford Parser, Berkeley Parser, Egret** (a clone of the Berkeley parser that can output forests)
Language Analysis (Ja-En):

- 3 morphological/dependency analysis combinations

<table>
<thead>
<tr>
<th></th>
<th>Juman+KNP</th>
<th>MeCab+CaboCha</th>
<th>KyTea+EDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation</td>
<td>Long</td>
<td>Medium</td>
<td>Short</td>
</tr>
<tr>
<td>OOV</td>
<td>Simple</td>
<td>Simple</td>
<td>Model</td>
</tr>
<tr>
<td>Parsing Unit</td>
<td>Bunsetsu</td>
<td>Bunsetsu</td>
<td>Word</td>
</tr>
<tr>
<td>Algorithm</td>
<td>CKY-Style</td>
<td>Cascaded Chunking</td>
<td>MST</td>
</tr>
</tbody>
</table>

- Use head rules to change dependency into CFG
  - For bunsetsu-based, last content word is head
  - Punctuation dependencies reversed
Language Analysis (Ja-En):

- PBMT
- PBMT+Pre
- Juman+KNP
- MeCab+CaboCha
- KyTea+EDA

**BLEU**

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBMT</td>
<td>15</td>
</tr>
<tr>
<td>PBMT+Pre</td>
<td>17</td>
</tr>
<tr>
<td>Juman+KNP</td>
<td>12</td>
</tr>
<tr>
<td>MeCab+CaboCha</td>
<td>14</td>
</tr>
<tr>
<td>KyTea+EDA</td>
<td>16</td>
</tr>
</tbody>
</table>

**RIBES**

<table>
<thead>
<tr>
<th>Method</th>
<th>RIBES Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBMT</td>
<td>66</td>
</tr>
<tr>
<td>PBMT+Pre</td>
<td>63</td>
</tr>
<tr>
<td>Juman+KNP</td>
<td>58</td>
</tr>
<tr>
<td>MeCab+CaboCha</td>
<td>62</td>
</tr>
<tr>
<td>KyTea+EDA</td>
<td>66</td>
</tr>
</tbody>
</table>

**Acceptability**

<table>
<thead>
<tr>
<th>Method</th>
<th>Acceptability</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBMT</td>
<td>2.6</td>
</tr>
<tr>
<td>PBMT+Pre</td>
<td>2.8</td>
</tr>
<tr>
<td>Juman+KNP</td>
<td>2.4</td>
</tr>
<tr>
<td>MeCab+CaboCha</td>
<td>2.8</td>
</tr>
<tr>
<td>KyTea+EDA</td>
<td>2.6</td>
</tr>
</tbody>
</table>
EDA vs. KNP/CaboCha

**Input:**
向嶽寺派  
祇園女御妹-後に平忠盛妻

**MeCab+CaboCha:**
向嶽寺 school  
祇園女御 younger sister : later became the wife of taira no tadamori

**KyTea+EDA:**
kogaku-ji temple school  
gion no nyogo younger sister - , later taira no tadamori 's wife

Smaller, more accurate segmentation provides better translations (EDA)
EDA vs. CaboCha/KNP

Input:
大宮学舎旧守衛所
文学部社会学科を設置

MeCab+CaboCha:  
former omiya campus . office  
department of faculty of letters society was established .

KyTea+EDA:  
omiya campus former guard office  
department of sociology , faculty of letters was established .

Word-based noun-phrase parsing helps translation (EDA)
EDA vs. CaboCha/KNP

Input:
芳崖と雅邦はともに地方の狩野派系絵師の家の出身であった。

MeCab+CaboCha:
hogai and gaho both was from a family of local painters of the kano school.

KyTea+EDA:
hogai and gaho from the family of the region of the kano together school series painter.

CaboCha/KNP wins followed no clear pattern. This case: CaboCha: “とみに→出身” EDA: “ともに→地方”
CaboCha vs. KNP

Input:
谷万太郎
1391年-山名氏清
1392年～1394年-畠山基国

JUMAN/KNP:
taro million tani
in 1391 , - the yamana clan
- in 1392 - 1394 hakekeyama ) province

MeCab+CaboCha:
mantaro tani
1391 , : ujikiyo yamana
1392 1394 : motokuni hakekeyama

Most prominent wins for CaboCha were segmentation
Conclusion

- **Egret** is best for English, and forests are important.
- **KyTea+EDA** is best for Japanese
  - At the moment, *morphological analysis is more important than parsing*?
- **Future directions**:
  - Forest-based parser!
  - Better bunsetsu → word dependency conversion rules
Other Settings
Question:

What other settings have a significant effect on translation results?
Composed Rules

• Combine two minimal rules into larger rules:

- Combine two minimal rules into larger rules:

  - English: "ate x1 x0"
  - Japanese: "ご飯を食べた"

- Composed rule:

  - English: "ate x0"
  - Japanese: "ご飯を食べた"
Composed Rules (En-Ja)

- Composed rules are very important
Number of Non-Terminals

<table>
<thead>
<tr>
<th>0 NT</th>
<th>1 NT</th>
<th>2 NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP</td>
<td>PP</td>
<td>VP</td>
</tr>
<tr>
<td>V</td>
<td>N</td>
<td>V</td>
</tr>
<tr>
<td>4</td>
<td>2-3</td>
<td>4-5</td>
</tr>
<tr>
<td>5</td>
<td>P</td>
<td>SUF</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>eat</td>
<td>た</td>
<td>eat</td>
</tr>
<tr>
<td>た</td>
<td>た</td>
<td>た</td>
</tr>
</tbody>
</table>

ate
ate x0
x1 x0
Number of Non-Terminals (En-Ja)

- 2 Non-terminals are necessary, but more are harmful
- Why? Larger are more noisy?
Binarization (En-Ja)

- Right or left much better than none
- In general right > left for En-Ja, left > right for Ja-En
Tuning

• Two evaluation measures:
  • BLEU correlated with fluency
  • RIBES correlated with adequacy
• Tune both of these measures with MERT
• Also, might be worth considering both [Duh+ 12], so we use linear combination BLEU+RIBES also
Experiments in English-Japanese Tree-to-String Machine Translation

**Tuning**

**En-Ja**

<table>
<thead>
<tr>
<th>BLEU</th>
<th>RIBES</th>
<th>BLEU+RIBES</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>17</td>
<td>18</td>
</tr>
</tbody>
</table>

**Ja-En**

<table>
<thead>
<tr>
<th>BLEU</th>
<th>RIBES</th>
<th>BLEU+RIBES</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>15.8</td>
<td>16</td>
</tr>
</tbody>
</table>

**Scores**

- BLEU: 66.5, 63.5, 65.5
- RIBES: 68, 64.5, 63.5
- BLEU+RIBES: 67, 64, 63
Conclusion
Insights

- How well does tree-to-string work for En-Ja, Ja-En?
  - As well as phrase-based with pre-ordering [Neubig+ 12]
  - Forest-to-string translation works better for En-Ja
- Egret worked best for English-Japanese KyTea+EDA worked the best for Japanese-English
- For Ja-En we need:
  - Better morphological analysis!
  - Pass multiple morphological analysis results to parsing!
  - n-best or forest based parser!
Thank You!