Forest-Based Translation Rule Extraction

Haitao Mi
Institute of Computing Technology

Liang Huang
University of Pennsylvania

EMNLP 2008 talk, Honolulu, Hawaii, October 2008
Translation Rule Extraction

- rule extraction is a central problem in Statistical MT
- especially in **linguistically syntax-based** systems
  - use parse trees ("syntax") from either or both sides
  - more informed translation thanks to syn. categories
  - but in practice worse than formal syntax only (Hiero)

<table>
<thead>
<tr>
<th>source</th>
<th>target</th>
<th>examples (partial)</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree-to-tree</td>
<td></td>
<td>Ding and Palmer (2006)</td>
<td>linguistic syntax</td>
</tr>
<tr>
<td>tree-to-string</td>
<td></td>
<td>Liu et al. (2006); Huang et al. (2006)</td>
<td></td>
</tr>
<tr>
<td>string-to-tree</td>
<td></td>
<td>Galley et al. (2006)</td>
<td></td>
</tr>
<tr>
<td>string-to-string</td>
<td></td>
<td>Chiang (2005)</td>
<td>formal</td>
</tr>
</tbody>
</table>
How to learn better rules?

- one major problem: parsing error affects rule set quality
- $k$-best trees? limited scope; too slow cf. (Venugopal et al 2008)
- we use packed forest of exponentially many trees
- result: 2.5 BLEU final improvement; better than Hiero
- experiments focused on tree-to-string systems

<table>
<thead>
<tr>
<th>source</th>
<th>target</th>
<th>examples (partial)</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree-to-tree</td>
<td></td>
<td>Ding and Palmer (2006)</td>
<td>linguistic</td>
</tr>
<tr>
<td>tree-to-string</td>
<td></td>
<td>Liu et al. (2006); Huang et al. (2006)</td>
<td>syntax</td>
</tr>
<tr>
<td>string-to-tree</td>
<td></td>
<td>Galley et al. (2006)</td>
<td>formal</td>
</tr>
<tr>
<td>string-to-string</td>
<td></td>
<td>Chiang (2005)</td>
<td></td>
</tr>
</tbody>
</table>
Outline

• Background: Tree-based Translation
  • Basic Rule Extraction Algorithm (Galley et al., 2004)

• Forest-based Rule Extraction

• Related Work

• Experiments
Forest-based Translation Rule Extraction

- get 1-best parse tree; then convert to English

Tree-based Translation

“Bush held a meeting with Sharon”
Tree-based Translation

- recursive rewrite by pattern-matching

\[
\begin{array}{c}
\text{IP} \\
\text{NP} \quad \text{VPB} \\
\text{NPB} \quad \text{CC} \quad \text{NPB} \quad \text{VV} \quad \text{AS} \quad \text{NPB} \\
Bùshí \quad yǔ \quad Shālóng \quad jǔxíng \quad le \quad huìtán
\end{array}
\]
Tree-based Translation

- recursive rewrite by pattern-matching
Tree-based Translation

- recursive rewrite by pattern-matching

(Huang, Knight, Joshi 2006)
Tree-based Translation

- recursive rewrite by pattern-matching
Tree-based Translation

- recursively solve unfinished subproblems

```
NPB
|    
Bùshí

| VPB |
VV   AS   NPB
|     |
jǔxíng le huìtán

with

NPB
|    
Shālóng
```
Tree-based Translation

- recursively solve unfinished subproblems

NPB  
|  Bùshí  

VPB

VV  AS  NPB

jǔxíng le huìtán

with

NPB  
|  Shālóng

Forest-based Translation Rule Extraction  (Huang, Knight, Joshi 2006)
Tree-based Translation

- recursively solve unfinished subproblems

Bush

```
  VPB
  /   \
 VV    AS    NPB
       /     /     /
  jǔxíng le huìtán
```

with

```
  NPB
  / \
 Shālóng
     /
 VPB
    /
 VV   AS   x₁:NPB
     /     /
  jǔxíng le
  → held x₁
```

Forest-based Translation Rule Extraction (Huang, Knight, Joshi 2006)
Tree-based Translation

- recursively solve unfinished subproblems

Forest-based Translation Rule Extraction (Huang, Knight, Joshi 2006)
Tree-based Translation

- recursively solve unfinished subproblems

Bush held with

NPB

| Shālóng

held

NPB

| huìtán

From Bush to held

NPB

| huìtán

(Huang, Knight, Joshi 2006)
Tree-based Translation

- continue pattern-matching

Bush held **NPB** with **NPB**

| huìtán | Shālóng |
Tree-based Translation

- continue pattern-matching

Bush held a meeting with Sharon

NPB | huìtán

NPB | Shālóng

(Huang, Knight, Joshi 2006)
Tree-based Translation

- continue pattern-matching

Bush held a meeting with Sharon
Tree-based Translation

- continue pattern-matching

Bush held a meeting with Sharon

pros: simplicity: separate parsing and decoding (fast!)
expressive grammar,
“extended domain of locality”
...

(Galley et al. 2004; Huang, Knight, Joshi 2006)
Tree-based Translation

- continue pattern-matching

Bush held a meeting with Sharon

**pros:** simplicity: separate parsing and decoding (fast!)
expressive grammar,
“extended domain of locality”
...

Q: where are the rules from?
Where are the rules from?

- source parse tree, target sentence, and alignment
- intuition: contiguous span

IP

NP  VPB

NPB  CC  NPB  VV  AS  NPB

Bùshí  yǔ  Shālóng  jǔxíng  le  huìtán

Bush  held  a  meeting  with  Sharon
Where are the rules from?

- source parse tree, target sentence, and alignment
- intuition: contiguous span

GHKM - (Galley et al 2004; 2006)
Where are the rules from?

- source parse tree, target sentence, and alignment
- intuition: contiguous span

GHKM - (Galley et al 2004; 2006)
Where are the rules from?

- source parse tree, target sentence, and alignment
- compute target spans
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span

```
IP
   "Bush .. Sharon"
   /
  "Bush \ with Sharon"
  /
NP
  "Bush"  CC  NPB
NPB  "with"  NPB
   "Sharon"

VPB
   "held .. meeting"
   /
  "held"  AS  NPB
V V  "led"  "led"
  "a meeting"
  /
AS  "le"
  "huítán"

Bush  held  a  meeting  with  Sharon
```
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span

admissible nodes
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span

GHKM - (Galley et al 2004; 2006)
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span

GHKM - (Galley et al 2004; 2006)
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span

 GHKM - (Galley et al 2004; 2006)
Forest-based Translation Rule Extraction

The Baseline Pipeline

training time

source sentence

parser

GIZA

word alignment

1-best parse

target sentence

rule extractor

translation ruleset
The Baseline Pipeline

Forest-based Translation Rule Extraction
Forest-based Translation Rule Extraction

The Baseline Pipeline

- **training time**
  - source sentence → parser → 1-best parse → rule extractor
    - GIZA → word alignment
  - target sentence

- **decoding time**
  - source sentence → parser → 1-best parse → pattern-matcher → translation ruleset
    - target sentence
Outline

- Background: Tree-based Translation and Rule Extraction
- Forest-based Rule Extraction
  - Background: Parse Forest
  - Forest-based Extraction
  - Inside-Outside Forest Pruning
  - Fractional Rule Counts
- Related Work
- Experiments
Packed Forest

- A compact representation of many parses
- By sharing common sub-derivations
- Polynomial-space encoding of exponentially large set

`(Klein and Manning, 2001; Huang and Chiang, 2005)`
Packed Forest

- a compact representation of many parses
- by sharing common sub-derivations
- polynomial-space encoding of exponentially large set

nodes → VP_{1,6} → hyperedges

a hypergraph

0 1 saw 2 him 3 with 4 a 5 mirror 6
parse the input into a forest instead of 1-best tree

Chinese *yu* can be either a CC ("and") or P ("with")
Chinese Forest

- parse the input into a forest instead of 1-best tree
- Chinese *yu* can be either a CC (“and”) or P (“with”)
Chinese Forest

- parse the input into a forest instead of 1-best tree
- Chinese yu can be either a CC ("and") or P ("with")
Forest-based Rule Extraction

- same at “where to cut”; different at “how to cut”

Forest-based Translation Rule Extraction
• same at “where to cut”; different at “**how** to cut”
Forest-based Rule Extraction

- same at “where to cut”; different at “how to cut”
Forest-based Rule Extraction

• same at “where to cut”; different at “how to cut”

$$IP(x_1;NPB \ x_2;VP) \rightarrow x_1 \ x_2$$
Forest-based Rule Extraction

- same at "where to cut"; different at "how to cut"

\[ IP(x_1:NPB \ x_2:VP) \rightarrow x_1 \ x_2 \]

Also in (Wang, Knight, Marcu, 2007); see related work.
Forest-based Rule Extraction

- same at “where to cut”; different at “how to cut”

\[
\text{IP}(x_1:\text{NPB} \ x_2:\text{VP}) \rightarrow x_1 \ x_2
\]
Forest-based Rule Extraction

- same at “where to cut”; different at “how to cut”

\[ IP(x_1:NPB \ x_2:VP) \rightarrow x_1 \ x_2 \]
Forest-based Rule Extraction

- same at “where to cut”; different at “how to cut”

Also in (Wang, Knight, Marcu, 2007); see related work
Forest-based Rule Extraction

- forest can extract smaller chunks of rules

\[ \text{IP}(x_1: \text{NPB} \ x_2: \text{VP}) \rightarrow x_1 \ x_2 \]

Also in (Wang, Knight, Marcu, 2007); see related work.
Forest-based Rule Extraction

- forest can extract smaller chunks of rules

$IP(x_1:NPB\ x_2:VP) \rightarrow x_1\ x_2$

$VP(x_1:PP\ x_2:VPB) \rightarrow x_2\ x_1$

also in (Wang, Knight, Marcu, 2007); see related work
Forest-based Rule Extraction

- forest can extract smaller chunks of rules

![Forest diagram with rules and examples]

\[ IP(x_1:NPB \ x_2:VP) \rightarrow x_1 \ x_2 \]

\[ VP(x_1:PP \ x_2:VPB) \rightarrow x_2 \ x_1 \]

\[ PP(x_1:P \ x_2:NPB) \rightarrow x_1 \ x_2 \]

Also in (Wang, Knight, Marcu, 2007); see related work.
Rule Extraction Pipeline

source sentence → parser → 1-best parse → word alignment → rule extractor → rule set

training time

baseline
Rule Extraction Pipeline

1. GIZA
2. Parse forest
3. Word alignment
4. Rule extractor
5. Rule set

training time

source sentence → parser → parse forest → word alignment → rule extractor → rule set

target sentence
Rule Extraction Pipeline

training time

source sentence \( \xrightarrow{\text{parser}} \) parse forest \( \xrightarrow{\text{word alignment}} \) pruned forest \( \xrightarrow{\text{rule extractor}} \) rule set

GIZA

target sentence

practical
**Inside-Outside Forest Pruning**

- prune *unpromising* hyperedges
- cost of best derivation that traverses $e$
- inside-outside, (max) marginal probs
- first compute Viterbi inside $\beta$, outside $\alpha$
- merit $\alpha \beta(e) = \alpha(v) \cdot p(e) \cdot \beta(u) \beta(w)$
- similar to “expected count” in EM
- prune away a hyperedge $e$ if
  \[ \frac{\alpha \beta(e)}{\beta(\text{TOP})} > p \]
  for some threshold $p$

(amount of deviation from 1-best derivation)

(Huang 2008)
Fractional Rule Counts

- tree-based: every rule extracted gets a unit count
- forest-based: should penalize rules extracted from non 1-best parses
  - each rule gets a fractional count based on parse hyperedges
- same idea as forest pruning: inside-outside merit

\[
\alpha\beta(r) = \alpha\beta(\{e, e'\}) = \alpha(v) \\
\cdot p(e) p(e') \\
\cdot \beta(u)\beta(x)\beta(y)
\]

\[
\text{count}(r) = \frac{\alpha\beta(r)}{\beta(\text{TOP})}
\]
The Whole Forest Pipeline

Training time

Source sentence → Parser → Parse forest → Pruned forest → Rule extractor → Ruleset

GIZA → Word alignment

Target sentence
The Whole Forest Pipeline

**Training Time**
- Source sentence
- **Parser**
- Parse forest
- **GIZA**
- Word alignment
- **Rule Extractor**
- Pruned forest
- **Rule Set**

**Decoding Time**
- Source sentence
- **Parser**
- 1-best parse
- **Tree-based Decoder**
  - (Huang, Knight, Joshi, 2006)
- Target sentence

Forest-based Translation Rule Extraction
The Whole Forest\textsuperscript{2} Pipeline

**Training Time**

- Source sentence
  - Parser → Parse forest
  - GIZA → Word alignment
  - Target sentence

**Decoding Time**

- Source sentence
  - Parser → Parse forest
  - Pruned forest
  - Rule extractor → Ruleset
  - Target sentence

Forest-based Translation Rule Extraction

(Mi, Huang, Liu, ACL 2008)
Related Work

• forest in rule extraction
  • Wang, Knight, Marcu (2007) pack different binarizations of a single parse tree into a binarization forest
  • we use a real parse forest of many different parses
  • then use EM to guess best binarization for each parse; real rules only extracted from one single binarized tree

• multiple parses for rule extraction
  • Venugopal et al (2008) use $k$-best trees; negative results
  • we will show small improvement from $k$-best trees, but big improvement from forests
Experiments

both small-scale and large scale
Small-Scale Experiments

- Chinese-to-English translation
  - on a tree-to-string system similar to (Liu et al, 2006)
- 31k sentences pairs (0.8M Chinese & 0.9M English words)
- GIZA++ aligned
- trigram language model trained on the English side
- dev: NIST 2002 (878 sent.); test: NIST 2005 (1082 sent.)
- Chinese-side parsed by the parser of Xiong et al. (2005)
  - modified to output a forest for each sentence (Huang 2008)
- $1$-best$^2$ baseline: 0.2430; Pharaoh: 0.2297
Forest vs. $k$-best Extraction

1.0 Bleu improvement over 1-best, twice as fast as 30-best extraction
Forest-based Translation Rule Extraction

Large-Scale Experiments

- FBIS: 239k sentence pairs (7M/9M Chinese/English words)
- forest in both extraction and decoding
- forest\(^2\) results is 2.5 points better than 1-best\(^2\)
- and outperforms Hiero (by quite a bit)

<table>
<thead>
<tr>
<th></th>
<th>1-best tree</th>
<th>forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-best tree</td>
<td>0.2560</td>
<td>0.2674</td>
</tr>
<tr>
<td>30-best trees</td>
<td>0.2634</td>
<td>0.2767</td>
</tr>
<tr>
<td>forest</td>
<td>0.2679</td>
<td>0.2816</td>
</tr>
<tr>
<td>Hiero</td>
<td></td>
<td>0.2738</td>
</tr>
</tbody>
</table>
Translation Examples

- **src**
  - 鲍威尔 说 与 阿拉法特 会谈 很 重要
  - Powell say with Arafat talk very important

- **ref**
  - Powell Said Talks with Arafat Very Important (headline)

- **1-best**
  - Powell said the very important talks with Arafat

- **forest**
  - Powell said his meeting with Arafat is very important

- **hiero**
  - Powell said very important talks with Arafat
Conclusion

• forest provides flexibility to extract better rules
  • contains exponentially more trees than k-best parses
  • efficient extraction thanks to structure sharing
• applicable to all linguistically syntax-based systems
  • tree-to-string, string-to-tree, tree-to-tree, tree-seq-to-str, ...
• very simple idea, but works very well in practice
  • \(~1\) Bleu points better than 1-best extraction
  • \(~2.5\) Bleu better when combined with forest decoding
    • outperforms the state-of-the art Hiero (Chiang, 2005)
Forest is your friend in machine translation.

Thank you!

you may need to prune, but please save the forest.

Thanks to Qun Liu, Kevin Knight, Aravind Joshi, Wei Wang, and the anonymous reviewers.