Forest-Based Translation

Haitao Mi
Institute of Computing Technology

Liang Huang
University of Pennsylvania

Qun Liu
Institute of Computing Technology

ACL 2008 talk, Columbus, OH, June 2008
prepared and presented by L. H.
Two Approaches in Syntax MT
Two Approaches in Syntax MT

- **string-based** (Wu 97; Chiang 05; Galley et al 06)
  - parse the source-language string
  - with a synchronous grammar
  - generate translations accordingly
Two Approaches in Syntax MT

- **string-based** (Wu 97; Chiang 05; Galley et al 06)
  - parse the source-language **string**
  - with a synchronous grammar
  - generate translations accordingly

*Bushi yu Shalong juxing le huitan*
Two Approaches in Syntax MT

- **string-based** (Wu 97; Chiang 05; Galley et al. 06)
  - parse the source-language **string**
    - with a synchronous grammar
  - generate translations accordingly

Forest-based Translation
Two Approaches in Syntax MT

- **string-based** (Wu 97; Chiang 05; Galley et al 06)
  - parse the source-language **string**
  - with a synchronous grammar
  - generate translations accordingly

```
Bushi yu Shalong juxing le huitan
```

Forest-based Translation
Two Approaches in Syntax MT

- **string-based** ([Wu 97; Chiang 05; Galley et al. 06])
  - parse the source-language *string*
  - with a synchronous grammar
  - generate translations accordingly

```
Bush with Shalong held a talk with Sharon
S0,1 PP1,3 VP3,6
Bushi yu Shalong juxing le huitan
```
Two Approaches in Syntax MT

- **string-based** (Wu 97; Chiang 05; Galley et al 06)
  - parse the source-language **string**
  - with a synchronous grammar
  - generate translations accordingly

Bush held a talk with Sharon

Bush with Shalong held a talk

Bushi yu Shalong juxing le huitan
Two Approaches in Syntax MT

- **string-based** (Wu 97; Chiang 05; Galley et al 06)
  - parse the source-language string
  - with a synchronous grammar
  - generate translations accordingly

- **tree-based** (Quirk et al 05; Liu et al 06; Huang et al 06)
  - start from source-language parse tree
  - recursively convert it to the target-language
  - faster decoding; more expressive translation grammar
  - **Problem**: commits to 1-best parse tree! \( \Rightarrow \) \( k \)-best trees?
Two Approaches in Syntax MT

- **string-based** (Wu 97; Chiang 05; Galley et al 06)
  - parse the source-language string
  - with a synchronous grammar
  - generate translations accordingly

- **tree-based** (Quirk et al 05; Liu et al 06; Huang et al 06)
  - start from source-language parse tree
  - recursively convert it to the target-language
  - faster decoding; more expressive translation grammar
  - **Problem**: commits to 1-best parse tree! => k-best trees?
  - **Idea**: use a parse forest! **Results**: ~2 Bleu points better
Outline

• Tree-based Translation
• Forest-based Translation
  • Parse Forest
  • Translation on Parse Forest
  • Integrating Language Model on Translation Forest
• Experiments
Tree-based Translation

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

Diagram:

```
  IP
   /\  
  NPB VP
   /   /\  
  NR PP VPB
   /       /\  
 Bùshí P NPB VV AS NPB
   / |   |  |   |   |  |
 yǔ NR jǔxíng le NN
   / |   |
 Shālóng huìtán
```
Tree-based Translation

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

IP

NPB    VP

NR    PP    VPB

Bǔshí P NPB VV AS NPB

yǔ NR jǔxíng le NN

Shālóng

huìtán

Galley et al., 2004; Liu et al., 2006; Huang et al., 2006
Tree-based Translation

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

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

Forest-based Translation

(Galley et al., 2004; Liu et al., 2006; Huang et al., 2006)
• get 1-best parse tree; then convert to English

Forest-based Translation

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

- get 1-best parse tree; then convert to English
Tree-based Translation

- recursively solve unfinished subproblems

Forest-based Translation

(Liu et al 06; Huang et al 06)
Tree-based Translation

- recursively solve unfinished subproblems

Forest-based Translation (Liu et al. 06; Huang et al. 06)
Tree-based Translation

- pattern-match tree-to-string translation rules

Bush

Forest-based Translation

(Liu et al 06; Huang et al 06)
Tree-based Translation

- pattern-match tree-to-string translation rules

Bush

Forest-based Translation (Liu et al. 06; Huang et al. 06)
• pattern-match tree-to-string translation rules

**Bush**

Forest-based Translation (Liu et al 06; Huang et al 06)
• continue pattern-matching

Bush held with

\[
\begin{array}{c}
\text{NPB} \\
| \\
\text{NN} \\
| \\
\text{huítán}
\end{array}
\quad
\begin{array}{c}
\text{NPB} \\
| \\
\text{NR} \\
| \\
\text{Shālóng}
\end{array}
\]
Tree-based Translation

- continue pattern-matching

Bush held with

NPB | NN | huìtán

NPB | NR | Shālóng

talk Sharon

(Liu et al. 06; Huang et al. 06)
Tree-based Translation

- continue pattern-matching

Bush held a talk with Sharon

(Galley et al 04; Liu et al 06; Huang et al 06)
Tree-based Translation

- continue pattern-matching

Bush held a talk with Sharon

**pros**: simplicity, faster decoding, expressive grammar, no need for binarization, ...

**cons**: commits to 1-best tree

Forest-based Translation

(Galley et al. 04; Liu et al. 06; Huang et al. 06)
Forest-based Translation

using a packed parse forest to direct the translation
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

(Klein and Manning, 2001; Huang and Chiang, 2005)
Pattern-Matching on Forest

“and” / “with”
Pattern-Matching on Forest

Forest-based Translation

“and” / “with”  (Chris Quirk, p.c.)
Pattern-Matching on Forest

“and” / “with” (Chris Quirk, p.c.)
Pattern-Matching on Forest

“and” / “with”  

(IP  
  NPB  
  x₁:NPB  
  CC  
  x₂:NPB  
  yǔ  
  “and”  
  x₁ x₃ with x₂)

NPBₐ₁  
NPB₂ₐ₃  
CC₁₂  
P₁₂  
NPB₀₃  
IP₀₆  
VP₁₆  
PP₁₃  
VPB₃₆  

Būshì  
yǔ  
Shālóng  
jǔxǐng  
le  
huǐtán  

NPBₐ₁  
NPB₀₁  
NR₀₁  

NPB₂ₐ₃  
NR₂₃  

NPBₐ₅₆  
NN₅₆  

(in Chris Quirk, p.c.)
Pattern-Matching on Forest

non-deterministic pattern-matching

and / with

(IP
  NPB
    x1:NPB
    CC
    x2:NPB
    "and"
  yǔ
  "and"
)

→ x1 x3 with x2

(x1:NPB
  yǔ
  "and"
  x3:VPB
)

NPB

NPB0,1
  NR0,1
  "Bushi"
  "and" / "with"

NPB2,3
  NR2,3
  "Shalong"

VPB3,6

VPB1,6

PP1,3

IP1,6

IP0,6

NP0,3

NP1,2

CC1,2
Translation Forest

Forest-based Translation
Translation Forest
"Sharon" held a talk with "Bush" and "Sharon".
“Bush held a talk with Sharon”
Decoding with Language Model

- decoding with \( n \)-gram language model
- is just intersecting a finite-state machine with the translation forest
- result in the finer-grained “translation+LM forest”
- we use *cube pruning* (Chiang 07; Huang and Chiang 07) to speed up the intersection
- for \( k \)-best translations (e.g., in MERT)
  - just run \( k \)-best Algorithms 3 (Huang and Chiang 05) on the translation+LM forest
The Whole Pipeline

- input sentence
- parse forest
  - pattern-matching w/ translation rules
- translation forest
  - cube pruning
- translation+LM forest
  - best derivation
  - $k$-best Algorithm 3
  - $1$-best output
  - $k$-best output
The Whole Pipeline

- Input sentence
  - Parser
    - Parse forest
      - Pattern-matching w/ translation rules
        - Translation forest
          - Cube pruning
            - Translation+LM forest
              - Best derivation
              - k-best Algorithm 3
                - 1-best output
                - k-best output
Experiments

both small-scale and large-scale experiments on Chinese-to-English translation
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
- Chinese-side parsed by the parser of Xiong et al. (2005)
  - 346k tree-to-string translation rules
- trigram language model trained on the English side
- dev: NIST 2002 (878 sent.); test: NIST 2005 (1082 sent.)
Results (BLEU)

- Pharaoh (Koehn, 2004) -- 0.2182
- 1-best tree decoding -- 0.2302
- 30-best trees decoding -- 0.2410
- forest-based decoding -- 0.2485
  - 1.8 Bleu over than 1-best, significant (p < 0.01)
  - forests from a modified version of the Chinese parser, similar to Huang (2008)
  - forests pruned by an Inside-Outside-style algorithm
  - even faster than 30-best trees!
$k$-best trees vs. forest-based
forest as virtual $\infty$-best list

- how often is the $i$th-best tree picked by the decoder?

```
30-best trees

forest decoding
```

Percentage of sentences (%) vs. Rank of the tree picked in $n$-best list

- 32% beyond 100-best
- 20% beyond 1000-best

suggested by Mark Johnson
Large-Scale Experiments

- 2.2M sentence pairs (57M Chinese and 62M English words)
- larger trigram models (1/3 of Xinhua Gigaword)
- also use bilingual phrases (BP) as flat translation rules
  - phrases that are consistent with syntactic constituents
- forest enables larger improvement with BP

<table>
<thead>
<tr>
<th></th>
<th>T2S</th>
<th>T2S+BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-best tree</td>
<td>0.2666</td>
<td>0.2939</td>
</tr>
<tr>
<td>30-best trees</td>
<td>0.2755</td>
<td>0.3084</td>
</tr>
<tr>
<td>forest</td>
<td>0.2839</td>
<td>0.3149</td>
</tr>
<tr>
<td>improvement</td>
<td>1.7</td>
<td>2.1</td>
</tr>
</tbody>
</table>
Conclusion and Future Work

- forest: a compact representation of ambiguities
- compromise between tree-based and string-based
  - combining the advantages of both
    - fast decoding, but does not commit to 1-best trees
    - separate translation grammar (STSG) from parsing (CFG)
- very simple idea, but works well in practice
  - ~2 Bleu points better than 1-best tree decoding
  - ~1 Bleu points better than 30-best trees, and faster!
- future work: use forest in rule-extraction also
Forest is your friend in machine translation.

stay tuned for another “forest-based” talk on parsing tomorrow morning

Thank you!

Acknowledgments: Chris Quirk, Kevin Knight, Mark Johnson, Yang Liu, ...