Efficient Incremental Decoding for Tree-to-String Translation

Liang Huang
Information Sciences Institute
University of Southern California

Haitao Mi
Institute of Computing Technology
Chinese Academy of Sciences

EMNLP 2010 talk, Cambridge, MA, Oct 2010
MT: Phrase-based vs. Syntax-based

- Most of statistical machine translation falls into phrase-based models
  - Allow arbitrary reorderings: exponential-time decoding
  - In practice: quadratic-time beam search
  - Linear-time with constant distortion limit; pretty fast

- Syntax-based models
  - Grammar-based reorderings: polynomial-time decoding
  - In practice: slower than phrase-based when with LM

Q: Borrow phrase-based decoding for syntax-based?
Preview of Results

• a phrase-based-style, incremental decoding algorithm for tree-to-string translation
• polynomial-time in theory, linear-time in practice
• 30 times faster than phrase-based Moses

<table>
<thead>
<tr>
<th></th>
<th>in theory</th>
<th>in practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>phrase-based</td>
<td>exponential</td>
<td>quadratic</td>
</tr>
<tr>
<td>tree-to-string</td>
<td>polynomial</td>
<td>linear</td>
</tr>
</tbody>
</table>
Preview of Results

- A phrase-based-style, incremental decoding algorithm for tree-to-string translation
- Polynomial-time in theory, linear-time in practice
- 30 times faster than phrase-based Moses

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses (in C++)</td>
<td>29.4</td>
<td>10.8s</td>
</tr>
<tr>
<td>Tree-to-string (in Python)</td>
<td>29.5</td>
<td>0.3s</td>
</tr>
</tbody>
</table>
Outline

- Background: Tree-to-String Translation
- Background: Phrase-based Decoding
- Incremental Decoding for Tree-to-String Translation
- Complexity Analysis
- Experiments
Tree-to-String Translation

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

![Tree-to-String Diagram]

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

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

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

Diagram:

```
       IP
      /   \
 NPB   VP
   / |   |
 NR PP VPB
 / |   |   |
 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-to-String Translation

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

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

---

Incremental Decoding

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

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

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

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

\[
\text{IP}(x_1:NPB \ x_2:VP) \rightarrow x_1 \ x_2
\]
Tree-to-String Translation

- recursively solve unfinished subproblems

(Liu et al 06; Huang et al 06)
Tree-to-String Translation

- recursively solve unfinished subproblems

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

- pattern-match tree-to-string translation rules

Bush

Incremental Decoding (Liu et al 06; Huang et al 06)
Tree-to-String Translation

- pattern-match tree-to-string translation rules

Bush

\[ \text{Shālóng} \quad \text{huìtán} \]

\[ \text{held } x_2 \text{ with } x_1 \]

(Liu et al 06; Huang et al 06)
Tree-to-String Translation

- pattern-match tree-to-string translation rules

Bush

Incremental Decoding
(Liu et al 06; Huang et al 06)
Tree-to-String Translation

- continue pattern-matching

Bush held with (Liu et al 06; Huang et al 06)
Tree-to-String Translation

- continue pattern-matching

Bush held with

NPB | NN | huìtán

talk

NPB | NR | Shālóng

Sharon

(Liu et al 06; Huang et al 06)
Tree-to-String Translation

- continue pattern-matching

Bush held a talk with Sharon
Tree-to-String Translation

- continue pattern-matching

Bush held a talk with Sharon

really simple! and fast: $O(n)$-time decoding!
Tree-to-String Translation

• continue pattern-matching

Bush held a talk with Sharon

really simple! and fast: $O(n)$-time decoding!

but with language model, it becomes slower...
Decoding w/ Language Model

- bottom-up (equivalent to top-down)
- each node is now split into several +LM nodes
- maintain LM signatures at both ends; and cross-product
Decoding w/ Language Model

- bottom-up (equivalent to top-down)
- each node is now split into several +LM nodes
- maintain LM signatures at both ends; and cross-product

Incremental Decoding
Decoding w/ Language Model

- bottom-up (equivalent to top-down)
- each node is now split into several +LM nodes
- maintain LM signatures at both ends; and cross-product

\[ \text{time: } O(n V^4(m-1)) \text{ for } m\text{-gram model} \]
Phrase-based Decoding (-LM)

held a talk with Sharon

source-side: coverage vector

held a talk

target-side: grow hypotheses

strictly left-to-right

时间复杂度：$O(2^n n^2)$ -- cf. 旅行销售员问题 (TSP)
Phrase-based Decoding (+LM)

- “refined” graph: annotated with language model words
- still dynamic programming, just larger search space

Incremental Decoding
Phrase-based Decoding (+LM)

- “refined” graph: annotated with language model words
- still dynamic programming, just larger search space

Incremental Decoding
Phrase-based Decoding (+LM)

- “refined” graph: annotated with language model words
- still dynamic programming, just larger search space
Phrase-based Decoding (+LM)

- “refined” graph: annotated with language model words
- still dynamic programming, just larger search space

\[ \text{time: } O(2^n n^2) \Rightarrow O(2^n n^2 V^m) \]

for \( m \)-gram language models
Why Phrase-based is Fast?

- phrase-based is exponential-time in theory
- in practice, linear-time w/ beam search + distortion limit
- key difference due to incremental expansion:
  - only need to keep rightmost LM words
Why Phrase-based is Fast?

- phrase-based is exponential-time in theory
- in practice, linear-time w/ beam search + distortion limit
- key difference due to incremental expansion:
  - only need to keep rightmost LM words

Q: can tree-to-string also become incremental?
Outline

- Background: Tree-to-String Translation
- Background: Phrase-based Decoding
- Incremental Decoding for Tree-to-String Translation
- Complexity Analysis
- Experiments
Incremental for Tree-to-String

- key intuition: tree coverage-vector?
Tree Coverage Vector as Stack

- stack (active derivation history): \[\varepsilon \rightarrow \text{IP} \] \[\text{IP} \rightarrow \text{NPB} \rightarrow \text{VP} \]
- three colors for nodes: white (uncovered), grey (partially covered), and black (covered)
Tree Coverage Vector as Stack

- stack (**active** derivation history):  \([\varepsilon \rightarrow \cdot \text{IP}] \ [\text{IP} \rightarrow \text{NPB} \cdot \text{VP}]\)

- three colors for nodes: white (uncovered), grey (partially covered), and black (covered)

“I have finished NPB subtree but not started with VP subtree”
Example Incremental Decoding

\[
[\varepsilon \rightarrow \langle s \rangle \cdot \text{IP} \langle /s \rangle]
\]

\[
\langle s \rangle
\]

Incremental Decoding
Incremental Decoding

Example Incremental Decoding

\[
[\epsilon \rightarrow <s> \cdot IP \ </s>] [\cdot IP \rightarrow \cdot NPB \ VP]
\]

<s>

\[\text{IP} \]

\[\text{NPB} \quad \text{VP}\]

\[\text{NR} \quad \text{PP} \quad \text{VPB}\]

\[\text{Bush}\]

\[\text{P} \quad \text{NPB} \quad \text{VV} \quad \text{AS} \quad \text{NPB}\]

\[\text{yǔ} \quad \text{NR} \quad \text{jǔxíng} \quad \text{le} \quad \text{NN}\]

\[\text{Shālóng} \quad \text{huìtán}\]

stack hypothesis

action: predict (push)
Example Incremental Decoding

\[
[\varepsilon \rightarrow <s> \text{ IP } <s>] [\text{IP} \rightarrow \text{NPB} \text{ VP}] [\text{NPB} \rightarrow \text{Bush}]
\]

<s>

\[
\text{stack} \quad \text{hypothesis}
\]

Incremental Decoding
Example Incremental Decoding

\[
\begin{align*}
[\varepsilon \rightarrow <s>, \text{IP} \langle /s \rangle ] & \ [\text{IP} \rightarrow \text{NPB}, \text{VP}] & [\text{NPB} \rightarrow \text{Bush}.]
\end{align*}
\]

<s> Bush

\[
\begin{align*}
& <s> \text{ IP } </s> \\
& \text{NPB} \\
& \text{VP} \\
& \text{PP} \\
& \text{VPB} \\
& \text{P} \\
& \text{NPB} \\
& \text{VV} \\
& \text{AS} \\
& \text{NPB} \\
& \text{yǔ} \\
& \text{NR} \\
& \text{jǔxíng} \\
& \text{le} \\
& \text{NN} \\
& \text{Shālóng} \\
& \text{huìtán}
\end{align*}
\]

action: scan

stack

hypothesis
Example Incremental Decoding

\[ \varepsilon \rightarrow <s> \cdot \text{IP} <\!/s> \] \[ \text{IP} \rightarrow \text{NPB} \cdot \text{VP} \]

<s> Bush

stack

decision

<s> IP <\!/s>

NPB

VP

action: complete (pop)
Example Incremental Decoding

\[
[\varepsilon \rightarrow <s> \cdot \text{IP} \langle/s\rangle] \ [\text{IP} \rightarrow \text{NPB} \cdot \text{VP}] \ [\text{VP} \rightarrow \cdot \text{held} \ \text{NPB} \ \text{with} \ \text{NPB}]
\]

\(<s>\) Bush

\(\text{stack}\)

\(\text{hypothesis}\)

\(\text{action: predict (push)}\)
Example Incremental Decoding

\[
[\varepsilon \rightarrow <s> \bullet \text{IP} \langle/s\rangle] \quad [\text{IP} \rightarrow \text{NPB} \bullet \text{VP}] \quad [\text{VP} \rightarrow \text{held} \bullet \text{NPB with NPB}]
\]

\(<s>\) Bush held

(action: scan)
Example Incremental Decoding

\[
[\epsilon \rightarrow <s> \text{ IP } <s>] \quad [\text{ IP } \rightarrow \text{ NPB } \cdot \text{ VP}] \quad [\text{ VP } \rightarrow \text{ held } \cdot \text{ NPB with NPB}] \quad [\text{ NPB } \rightarrow . \text{ talks }]
\]

<s> Bush held

\[
<\text{s}> \quad \text{IP} \quad <\text{s}>
\]

\[
\text{NPB} \quad \text{VP}
\]

\[
\text{PP} \quad \text{VPB}
\]

\[
\text{P} \quad \text{NPB} \quad \text{VV} \quad \text{AS} \quad \text{NPB}
\]

\[
\text{yǔ} \quad \text{NR} \quad \text{jiǔxíng} \quad \text{le} \quad \text{NN} \quad \text{huìtán}
\]

\[
\text{Bush} \quad \text{held} \quad \text{NPB} \quad \text{with} \quad \text{NPB}
\]

\[
\text{Shālóng} \quad \text{talks}
\]

\[
\text{action: predict (push)}
\]
Example Incremental Decoding

\[
\begin{align*}
&[\varepsilon \rightarrow <s> \quad \text{IP} \quad <s>] \\
&[\text{IP} \rightarrow \text{NPB} \quad \text{VP}] \\
&[\text{VP} \rightarrow \text{held} \quad \text{NPB} \quad \text{with} \quad \text{NPB}] \\
&[\text{NPB} \rightarrow \text{talks}] \\
\end{align*}
\]

<s> Bush held talks

\text{action: scan}
Example Incremental Decoding

\[(\varepsilon \rightarrow <s> \cdot \text{IP} \langle /s \rangle) \ [\text{IP} \rightarrow \text{NPB} \cdot \text{VP}] \ [\text{VP} \rightarrow \text{held} \ \text{NPB} \cdot \text{with} \ \text{NPB}]\]

<s> Bush held talks

\[\text{action: complete (pop)}\]
Example Incremental Decoding

\[
[\varepsilon \rightarrow <s> \bullet IP \langle/s\rangle] \ [IP \rightarrow NPB\bullet VP] \ [VP \rightarrow \text{held} \ NPB \ \text{with} \bullet NPB]
\]

\(<s>\) Bush held talks with

\(\text{action: scan}\)
Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP → held NPB with•NPB][NPB→•Sharon]

<s> Bush held talks with

action: predict (push)
Example Incremental Decoding

\[
[\varepsilon \rightarrow <s> \cdot \text{IP} <\!/s>] \ [\text{IP} \rightarrow \text{NPB} \cdot \text{VP}] \ [\text{VP} \rightarrow \text{held} \ \text{NPB} \ \text{with} \cdot \text{NPB}] [\text{NPB} \rightarrow \text{Sharon} \cdot]
\]

\(<s>\) Bush held talks with Sharon

Action: scan
Example Incremental Decoding

$$\left[ \epsilon \rightarrow \langle s \rangle \ \cdot \text{IP} \ \rangle \langle s \rangle \right] \ [\text{IP} \rightarrow \text{NPB} \cdot \text{VP}] \ [\text{VP} \rightarrow \text{held} \ \text{NPB} \ \text{with} \ \text{NPB} \cdot]$$

$$\langle s \rangle \ \text{Bush held talks with Sharon}$$

**action:** complete (pop)
Example Incremental Decoding

\[ [\varepsilon \rightarrow <s> \cdot \text{IP} \cdot <s>] [\text{IP} \rightarrow \text{NPB} \cdot \text{VP} \cdot ] \]

<s> Bush held talks with Sharon

**action**: complete (pop)
Example Incremental Decoding

\[ \varepsilon \rightarrow <s> \text{ IP. } </s> \]

<s> Bush held talks with Sharon

action: complete (pop)
Example Incremental Decoding

\[ \varepsilon \rightarrow <s> \text{ IP } </s> \]

<s> Bush held talks with Sharon </s>

**action:** scan
Example Incremental Decoding

\[ \varepsilon \rightarrow \langle s \rangle \text{ IP } \langle s \rangle . \]

\(<s>\) Bush held talks with Sharon \(<s>\)

Incremental Decoding
Complexity Analysis

- how many possible derivation stacks?
- exponential in root-to-leaf path length (tree depth)
- tree depth $O(\log n)$; const # rules => $O(c^{\log n})=O(n^{\log c})$
- so avg-case complexity is polynomial (see proof in paper)
Complexity Analysis

- how many possible derivation stacks?
- exponential in root-to-leaf path length (tree depth)
- tree depth $O(\log n)$; const # rules $\Rightarrow O(c^{\log n}) = O(n^{\log c})$
- so avg-case complexity is polynomial (see proof in paper)
Complexity Analysis

- how many possible derivation stacks?
- exponential in root-to-leaf path length (tree depth)
- tree depth $O(\log n)$; const # rules $\Rightarrow O(c^{\log n}) = O(n^{\log c})$
- so avg-case complexity is polynomial (see proof in paper)

constant # of rules per node
Beam Search in Practice

- very similar to phrase-based beam search
- coverage-vectors => derivation stacks
- beaming: # of Chinese tree nodes in black or grey
- assume constant # of rules per tree node: linear-time
Beam Search in Practice

- very similar to phrase-based beam search
- coverage-vectors => derivation stacks
- beaming: # of Chinese tree nodes in black or grey
- assume constant # of rules per tree node: linear-time
Watanabe et al (2006) presents similar incremental decoding algorithm for Hiero-style systems

- but complexity is super-polynomial in theory
- and quadratic in practice (just like phrase-based)
- requires Greibach Normal Form grammar $A \rightarrow a B C D$

Dyer and Resnik (2010) use two-pass decoding

- first-pass: no LM. incremental Earley-style
- second-pass: +LM. bottom-up CKY w/ cube pruning

ours work: one-pass, incremental, +LM, all grammars
Experiments
Experimental Setup

- Chinese-to-English translation
  - on a Python implementation of tree-to-string system
- 1.5M sentence pairs (38M/32M words in Chn/Eng)
- dev: NIST 2006 (616 sent); test: NIST 2008 (691 sent.)
- Chinese-side parsed by Berkeley parser (Petrov & Klein, 07)
- rules extracted using GHKM algorithm (Galley et al, 04; 06)
- trigram language model trained on the English side
- feature weights tuned using MERT (Och, 03)
Comparison with Moses

- we train/tune Moses with various distortion limits
- our incremental tree-to-string is ~30 times faster
  - this includes parsing time (0.2s per sentence)

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses (optimal distortion limit=10)</td>
<td>29.4</td>
<td>10.8s</td>
</tr>
<tr>
<td>tree-to-string: incremental (b=10)</td>
<td>29.5</td>
<td>0.3s</td>
</tr>
<tr>
<td>tree-to-string: incremental (b=50)</td>
<td>30.0</td>
<td>0.8s</td>
</tr>
</tbody>
</table>
Comparison with Moses

- Incremental tree-to-string is linear-time in practice
- and 30 times faster than Moses (distortion limit=10)
### Comparison with Cube Pruning

- Incremental is slightly faster than cube pruning

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses (optimal $d_{\text{max}}=10$)</td>
<td>29.4</td>
<td>10.8s</td>
</tr>
<tr>
<td>tree-to-string: incremental ($b=10$)</td>
<td>29.5</td>
<td>0.3s</td>
</tr>
<tr>
<td>tree-to-string: incremental ($b=50$)</td>
<td>30.0</td>
<td>0.8s</td>
</tr>
<tr>
<td>tree-to-string: cube pruning ($b=10$)</td>
<td>29.5</td>
<td>0.6s</td>
</tr>
<tr>
<td>tree-to-string: cube pruning ($b=50$)</td>
<td>30.0</td>
<td>1.0s</td>
</tr>
</tbody>
</table>
Incremental Decoding

Comparison with Cube Pruning

- incremental is slightly faster than cube pruning
- note they are very different (orthogonal) techniques
  - we envision their combination will be even faster
Comparison with Cube Pruning

- incremental is slightly faster than cube pruning
- note they are very different (orthogonal) techniques
  - we envision their combination will be even faster
Comparison with Cube Pruning

- incremental is slightly faster than cube pruning
- note they are very different (orthogonal) techniques
- we envision their combination will be even faster
Conclusion and Future Work

<table>
<thead>
<tr>
<th></th>
<th>in theory</th>
<th>in practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>phrase-based</td>
<td>exponential</td>
<td>quadratic</td>
</tr>
<tr>
<td>tree-to-string</td>
<td>polynomial</td>
<td>linear</td>
</tr>
</tbody>
</table>

- an incremental algorithm for tree-to-string translation
- linear-time in practice, and 30 times faster than Moses
- very different from cube pruning
  - cube pruning applies to phrase-based also (Huang/Chiang, 07)
- future work 1: combine cube pruning w/ incremental
- future work 2: extend to other syntax-based models
非常 感谢！
Thank you very much!
Tree Depth: Mean and Variance

- logarithmic mean and variance of tree-depth
- (needed for avg.-case polynomial-time complexity)