Forest Rescoring

Faster Decoding with Integrated Language Models

Liang Huang  David Chiang

ACL 2007, Praha, Česká republika
Statistical Machine Translation

Huang and Chiang

(Knight and Koehn, 2003)
Statistical Machine Translation

- Spanish/English Bilingual Text
- Statistical Analysis
- translation model (TM)
- competency
- Broken English
- English
- language model (LM)
- fluency
- n-best rescoring
- Que hambre tengo yo
- What hunger have I
- Hungry I am so
- Have I that hunger
- I am so hungry
- How hunger have I
- ...
Statistical Machine Translation

- Spanish/English Bilingual Text
  - Statistical Analysis
  - Translation model (TM) competency
  - Broken English
  - Language model (LM) fluency
  - English

Huang and Chiang
Statistical Machine Translation

Huang and Chiang

Que hambre tengo yo → integrated decoder → I am so hungry

computationally challenging! 😞
Statistical Machine Translation

- Spanish/English Bilingual Text
- English Text
- Integrated decoder

- Phrase-based Translation Model (TM)
- Syntax-based Translation Model
- n-gram Language Model (LM)

Que hambre tengo yo → I am so hungry

Computationally challenging! 😞
Forest Rescoring

Spanish/English Bilingual Text

Statistical Analysis

English Text

Statistical Analysis

Spanish

Bilingual Text

English

Sentence: Que hambre tengo yo

I am so hungry

Integrated decoder

n-gram LM

Broken English

Packed forest

computationally challenging! 😞
Forest Rescoring

- **Spanish/English Bilingual Text**
  - Statistical Analysis
  - Spanish
  - Que hambre tengo yo

- **English Text**
  - Statistical Analysis
  - English
  - I am so hungry

- **n-gram LM**
  - Broken English

- **integrated decoder**

**on-the-fly rescoring**

Computationally challenging! 😞

Huang and Chiang
**Forest Rescoring**

- **Spanish/English Bilingual Text**
- **English Text**
- **Statistical Analysis**
- **n-gram LM**
- **Integrated decoder**
  - on-the-fly rescoring
  - Que hambre tengo yo → I am so hungry

**Significant speed-up:** 10~30 times faster! 😊

- **Forest rescoring**

-Huang and Chiang
The Forest Framework

unifying phrase- and syntax-based decoding
Phrase-based Decoding

与 沙龙 举行 了 会谈

yu Shalong juxing le huitan

held a talk with Sharon

source-side: coverage vector

held a talk

target-side: grow hypotheses strictly left-to-right

held a talk

held a talk with Sharon

source-side: coverage vector

...
Syntax-based Translation

• synchronous context-free grammars (SCFGs)
• context-free grammar in two dimensions
• generating pairs of strings/trees simultaneously
• co-indexed nonterminal further rewritten as a unit

\[
\begin{align*}
VP & \rightarrow PP^{(1)} VP^{(2)}, \\
VP & \rightarrow juxing le huitan, \\
PP & \rightarrow yu Shalong,
\end{align*}
\]

held a meeting

with Sharon
Translation as Parsing

- translation with SCFGs $\Rightarrow$ monolingual parsing
- parse the source input with the source projection
- build the corresponding target sub-strings in parallel

$VP \rightarrow PP^{(1)} VP^{(2)}$
$VP \rightarrow juxing le huitan$
$PP \rightarrow yu Shalong$

$VP_{1, 6}$
$PP_{1, 3}$
$VP_{3, 6}$
$yu Shalong$
$juxing le huitan$
Translation as Parsing

- translation with SCFGs => monolingual parsing
- parse the source input with the source projection
- build the corresponding target sub-strings in parallel

\[
\begin{align*}
\text{VP} &\rightarrow \text{PP}^{(1)} \text{ VP}^{(2)}, & \text{VP}^{(2)} \text{ PP}^{(1)} \\
\text{VP} &\rightarrow juxing \ le \ huitan, & \text{held a meeting} \\
\text{PP} &\rightarrow yu \ Shalong, & \text{with Sharon}
\end{align*}
\]
Translation as Parsing

- translation with SCFGs => monolingual parsing
- parse the source input with the source projection
- build the corresponding target sub-strings in parallel

\[
\begin{align*}
\text{VP} & \rightarrow \text{PP}^{(1)} \text{VP}^{(2)}, \\
\text{VP} & \rightarrow \text{juxing le huitan}, \\
\text{PP} & \rightarrow \text{yu Shalong}, \\
\text{VP}^{(2)} & \rightarrow \text{PP}^{(1)} \\
\text{VP}^{(2)} & \rightarrow \text{held a meeting} \\
\text{PP}^{(1)} & \rightarrow \text{held a talk with Sharon} \\
\text{PP}^{(1)} & \rightarrow \text{yu Shalong} \\
\text{VP}^{(2)} & \rightarrow \text{juxing le huitan}
\end{align*}
\]
Packed Forest

- A compact representation of all translations
- Has a structure of hypergraph (graph is a special case)

<table>
<thead>
<tr>
<th>Phrase-based: graph</th>
<th>Syntax-based: hypergraph</th>
</tr>
</thead>
</table>

\[\text{PP}_{1,3} \rightarrow \text{VP}_{1,6} \rightarrow \text{VP}_{3,6}\]
Packed Forest

- a compact representation of all translations
- has a structure of hypergraph (graph is a special case)

phrase-based: graph  syntax-based: hypergraph
Adding a Bigram Model

- PP1,3: with ... Sharon, along ... Sharon, with ... Shalong
- VP1,6: held ... talk, held ... meeting, hold ... talks
- VP3,6: held ... talk

LM items: +LM items
Adding a Bigram Model

[Diagram showing the structure of adding a bigram model, with nodes representing phrases like "... meeting", "... talk", "... talks", and edges indicating transitions and additions.]

+LM items

- PP\(_{1,3}\)
  - with ... Sharon
  - along ... Sharon
  - with ... Shalong

- VP\(_{1,6}\)
  - held ... talk
  - held ... meeting
  - hold ... talks

Huang and Chiang

Forest Rescoring
Adding a Bigram Model

PP₁, 3 → with ... Sharon

VP₁, 6 → held ... talk
along ... Sharon
with ... Shalong

VP₃, 6 → held ... meeting
hold ... talks
Adding a Bigram Model

Huang and Chiang

Forest Rescoring 10
Adding a Bigram Model

Huang and Chiang

Forest Rescoring
Adding a Bigram Model

PP₁, 3

VP₁, 6

VP₃, 6

+LM items

with ... Sharon
along ... Sharon
with ... Shalong

held ... talk
held ... meeting
hold ... talks

held ... Sharon

bigram

... talk

bigram

... meeting

... Shalong

... Sharon

Sharon

... talks
Adding a Bigram Model

+LM items

PP₁, 3

VP₁, 6

VP₃, 6

PP₁, 3

VP₁, 6

VP₃, 6

Huang and Chiang
Conventional Beam Search

- beam search: only keep top-$k$ +LM items at each node
- but there are many ways to derive each node
- can we avoid enumerating all combinations?
  - best-first enumeration?

Huang and Chiang

Forest Rescoring
Cube Pruning

monotonic grid?

(VP_{3,6} \text{ held} \star \text{meeting})

(VP_{3,6} \text{ held} \star \text{talk})

(VP_{3,6} \text{ hold} \star \text{conference})

<table>
<thead>
<tr>
<th></th>
<th>1.0</th>
<th>3.0</th>
<th>8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2.0</td>
<td>4.0</td>
<td>9.0</td>
</tr>
<tr>
<td>1.1</td>
<td>2.1</td>
<td>4.1</td>
<td>9.1</td>
</tr>
<tr>
<td>3.5</td>
<td>4.5</td>
<td>6.5</td>
<td>11.5</td>
</tr>
</tbody>
</table>
## Cube Pruning

### Non-monotonic Grid Due to LM Combo Costs

<table>
<thead>
<tr>
<th></th>
<th>PP with * Sharon</th>
<th>PP along * Sharon</th>
<th>PP with * Shalong</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP held * meeting)</td>
<td>1.0 2.0 + 0.5</td>
<td>4.0 + 5.0</td>
<td>9.0 + 0.5</td>
</tr>
<tr>
<td>(VP held * talk)</td>
<td>1.1 2.1 + 0.3</td>
<td>4.1 + 5.4</td>
<td>9.1 + 0.3</td>
</tr>
<tr>
<td>(VP hold * conference)</td>
<td>3.5 4.5 + 0.6</td>
<td>6.5 + 10.5</td>
<td>11.5 + 0.6</td>
</tr>
</tbody>
</table>
Cube Pruning

non-monotonic grid due to LM combo costs

bigram (meeting, with)

<table>
<thead>
<tr>
<th></th>
<th>1.0</th>
<th>3.0</th>
<th>8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP with 1,3</td>
<td>2.0 + 0.5</td>
<td>4.0 + 5.0</td>
<td>9.0 + 0.5</td>
</tr>
<tr>
<td>PP along 1,3</td>
<td>2.1 + 0.3</td>
<td>4.1 + 5.4</td>
<td>9.1 + 0.3</td>
</tr>
<tr>
<td>PP with 1,3</td>
<td>4.5 + 0.6</td>
<td>6.5 + 10.5</td>
<td>11.5 + 0.6</td>
</tr>
</tbody>
</table>

Huang and Chiang
Cube Pruning

- **VP\(_{1,6}\)**
- **PP\(_{1,3}\)**
- **VP\(_{3,6}\)**

### Non-monotonic Grid due to LM Combo Costs

<table>
<thead>
<tr>
<th></th>
<th>(PP 1,3 with * Sharon)</th>
<th>(PP 1,3 along * Sharon)</th>
<th>(PP 1,3 with * Shalong)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP(_{1,6}) held * meeting)</td>
<td>1.0</td>
<td>3.0</td>
<td>8.0</td>
</tr>
<tr>
<td>(VP(_{1,6}) held * talk)</td>
<td>1.0</td>
<td>2.5</td>
<td>9.0</td>
</tr>
<tr>
<td>(VP(_{1,6}) hold * conference)</td>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
</tr>
<tr>
<td>(VP(_{1,6}) hold * conference)</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>
**Cube Pruning**

**k-best parsing**  
(Huang and Chiang, 2005)

- a priority queue of candidates  
- extract the best candidate

<table>
<thead>
<tr>
<th></th>
<th>PP with * Sharon</th>
<th>PP along * Sharon</th>
<th>PP with * Shalong</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP&lt;sup&gt;held&lt;/sup&gt; * meeting)&lt;sub&gt;3,6&lt;/sub&gt;</td>
<td>1.0</td>
<td>2.5</td>
<td>9.0</td>
</tr>
<tr>
<td>(VP&lt;sup&gt;held&lt;/sup&gt; * talk)&lt;sub&gt;3,6&lt;/sub&gt;</td>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
</tr>
<tr>
<td>(VP&lt;sup&gt;hold&lt;/sup&gt; * conference)&lt;sub&gt;3,6&lt;/sub&gt;</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>

K-best parsing (Huang and Chiang, 2005)
**Cube Pruning**

*k*-best parsing  
(Huang and Chiang, 2005)

- a priority queue of candidates
- extract the best candidate
- push the two successors

<table>
<thead>
<tr>
<th></th>
<th>(PP with * Sharon)</th>
<th>(PP along * Sharon)</th>
<th>(PP with * Shalong)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP held * meeting)</td>
<td>1.0</td>
<td>3.0</td>
<td>8.0</td>
</tr>
<tr>
<td>(VP held * talk)</td>
<td>1.0</td>
<td>2.5</td>
<td>9.0</td>
</tr>
<tr>
<td>(VP hold * conference)</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>
**Cube Pruning**

**k-best parsing**  
(Huang and Chiang, 2005)

- a priority queue of candidates  
- extract the best candidate  
- push the two successors

<table>
<thead>
<tr>
<th></th>
<th>(PP with * Sharon)</th>
<th>(PP along * Sharon)</th>
<th>(PP with * Shalong)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(VP (_{3,6}^{\text{held}}) * meeting)</strong>**</td>
<td>1.0</td>
<td>3.0</td>
<td>8.0</td>
</tr>
<tr>
<td><strong>(VP (_{3,6}^{\text{held}}) * talk)</strong> *</td>
<td>1.0</td>
<td>2.5</td>
<td>9.0</td>
</tr>
<tr>
<td><strong>(VP (_{3,6}^{\text{hold}}) * conference)</strong></td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>

Huang and Chiang
Cube Pruning

items are popped out-of-order

solution: keep a buffer of pop-ups

2.5 2.4 5.1

<table>
<thead>
<tr>
<th></th>
<th>PP with * Sharon</th>
<th>PP along * Sharon</th>
<th>PP with * Shalong</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP held * meeting)</td>
<td>1.0</td>
<td>3.0</td>
<td>8.0</td>
</tr>
<tr>
<td>(VP held * talk)</td>
<td>1.0</td>
<td>2.5</td>
<td>9.0</td>
</tr>
<tr>
<td>(VP hold * conference)</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>
Cube Pruning

items are popped out-of-order

solution: keep a buffer of pop-ups

2.5 2.4 5.1

finally re-sort the buffer
and return inorder:

2.4 2.5 5.1

<table>
<thead>
<tr>
<th>(VP held * meeting)</th>
<th>1.0</th>
<th>3.0</th>
<th>8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP held * talk)</td>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
</tr>
<tr>
<td>(VP hold * conference)</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>

solution: keep a buffer of pop-ups

2.5 2.4 5.1

finally re-sort the buffer
and return inorder:

2.4 2.5 5.1

<table>
<thead>
<tr>
<th>(VP held * meeting)</th>
<th>1.0</th>
<th>3.0</th>
<th>8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP held * talk)</td>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
</tr>
<tr>
<td>(VP hold * conference)</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>

items are popped out-of-order

solution: keep a buffer of pop-ups

2.5 2.4 5.1

finally re-sort the buffer
and return inorder:

2.4 2.5 5.1

<table>
<thead>
<tr>
<th>(VP held * meeting)</th>
<th>1.0</th>
<th>3.0</th>
<th>8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP held * talk)</td>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
</tr>
<tr>
<td>(VP hold * conference)</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>

items are popped out-of-order

solution: keep a buffer of pop-ups

2.5 2.4 5.1

finally re-sort the buffer
and return inorder:

2.4 2.5 5.1

<table>
<thead>
<tr>
<th>(VP held * meeting)</th>
<th>1.0</th>
<th>3.0</th>
<th>8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VP held * talk)</td>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
</tr>
<tr>
<td>(VP hold * conference)</td>
<td>3.5</td>
<td>5.1</td>
<td>17.0</td>
</tr>
</tbody>
</table>
Across Hyperedges

$k$-best parsing
(Huang and Chiang, 2005)

process all hyperedges simultaneously!
significant savings of computation
Across Hyperedges

$k$-best parsing  
(Huang and Chiang, 2005)

on-the-fly rescoring at each node,  
instead of only at the root node

process all hyperedges simultaneously!  
significant savings of computation
Cube Growing

- an even faster variant of cube pruning
- motivation
  - why do we have a fixed beam of size $k$ at each node?
    - why don’t we on-the-fly figure out the minimum $k$?
- cube growing uses
  - lazy $k$-best parsing \((\text{Huang and Chiang, 2005, Algorithm 3})\)
  - on-demand computation
- but harder to implement
Syntax-based Experiments
Tree-to-String System

- syntax-directed, English to Chinese (Huang, Knight, Joshi, 2006)
- first parse input, and then recursively transfer

synchronous tree-substitution grammars (STSG)
(Galley et al., 2004; Eisner, 2003)

search space still a hypergraph

tested on 140 sentences slightly better BLEU scores than Pharaoh
Tree-to-String System

- syntax-directed, English to Chinese \(\text{(Huang, Knight, Joshi, 2006)}\)
- first parse input, and then recursively transfer

Synchronous tree-substitution grammars (STSG)
\(\text{(Galley et al., 2004; Eisner, 2003)}\)

Search space still a hypergraph

Tested on 140 sentences slightly better BLEU scores than Pharaoh
Speed vs. Search Quality

- log Prob

average model cost

average number of +LM items explored per sentence

speed ++

quality ++

full-integration

cube pruning

cube growing
Speed vs. Search Quality

- speed ++
- quality ++

10 times faster
Huang and Chiang

Speed vs. Search Quality

- speed ++
- quality ++

same parameters

10 times faster
Speed vs. Search Quality

- speed ++
- quality ++
- 10 times faster

Huang and Chiang

Forest Rescoring 23
Speed vs. Search Quality

- speed ++
- quality ++

10 times faster

average number of +LM items explored per sentence

full-integration
cube pruning
cube growing

same parameters

Huang and Chiang
Speed vs. Translation Accuracy

- speed ++
- quality ++

The diagram shows the relationship between BLEU score and the average number of +LM items explored per sentence for different methods:
- Full-integration
- Cube pruning
- Cube growing

The x-axis represents the average number of +LM items explored per sentence, while the y-axis represents the BLEU score.
Cube-Pruning for Phrase-based Decoding
Syntax vs. Phrase-based

VP

PP₁, 3

PP₁, 4

NP₁, 4

VP₃, 6

VP₄, 6

VP₄, 6

... talk

... meeting

... talks

... Sharon

... Shalong

... minister

... held

... hold

... did

with Sharon

held a talk

a talk
Syntax vs. Phrase-based

Huang and Chiang

Forest Rescoring 26
Alternative Phrase-Pairs

grouping into hyperedge bundles

... talk
... meeting
... talks

... held
... hold
... did

... Sharon
... Shalong
... minister
Alternative Phrase-Pairs

grouping into hyperedge bundles

Pharaoh would explore all cells

... talk
... meeting
... talks

... Sharon
... Shalong
... minister

... held
... hold
... did

held a meeting
held a talk
conference

with Ariel Sharon
and Sharon
with Sharon
Cube Pruning

but we explore the grids in a best-first fashion

... talk
... meeting
... talks

...held a meeting
...hold a talk

...Sharon
...Shalong
...minister

... held
... hold
... did

... held a reunion

with Ariel Sharon
and Sharon
with Sharon

conference
a meeting
a talk

in a best-first fashion
but we explore the grids in a best-first fashion

in practice we use per-bin pruning as in Pharaoh

... talk
... meeting
... talks

... held
... hold
... did

... Sharon
... Shalong
... minister

in practice we use per-bin pruning as in Pharaoh

... held
... hold
... did
In Practice: per-bin Pruning

Pharaoh

1
2
3
4
5
In Practice: per-bin Pruning

Pharaoh

Cube Pruning
In Practice: per-bin Pruning

Cube Pruning

( ___●●● meeting )
( ___●●● talk )
( ___●●● conference )

<table>
<thead>
<tr>
<th>1.0</th>
<th>2.5</th>
<th>8.3</th>
<th>8.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
<td>8.4</td>
</tr>
<tr>
<td>3.5</td>
<td>9.2</td>
<td>17.0</td>
<td>15.2</td>
</tr>
</tbody>
</table>

Huang and Chiang
Forest Rescoring 29
In Practice: per-bin Pruning

Cube Pruning

(___ meeting) 1.0  2.5  8.3  8.5
(____ talk)    1.1  2.4  9.5  8.4
(____ conference) 3.5  9.2 17.0 15.2
In Practice: per-bin Pruning

Cube Pruning

hyperedge bundles

(meeting) 1.0 2.5 8.3 8.5
(talk) 1.1 2.4 9.5 8.4
(conference) 3.5 9.2 17.0 15.2

with Sharon and Sharon
with Ariel Sharon

1.0 4.0 7.0
In Practice: per-bin Pruning

hyperedge bundles

Cube Pruning

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2.5</td>
<td>8.3</td>
<td>8.5</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>2.4</td>
<td>9.5</td>
<td>8.4</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>9.2</td>
<td>17.0</td>
<td>15.2</td>
<td></td>
</tr>
</tbody>
</table>

with Sharon and Sharon
with Ariel Sharon

1.0 4.0 7.0

meeting

talk

conference

close up
In Practice: per-bin Pruning

hyperedge bundles

Cube Pruning

(close up)
In Practice: per-bin Pruning

Cube Pruning

hyperedge bundles

(____●● meeting) 1.0 2.5 8.3
(____●● talk) 1.1 2.4
(____●● conference) 3.5

with Sharon and Sharon
with Ariel Sharon

1.0 4.0 7.0
Speed vs. Search Quality

tested on our faithful clone of Pharaoh

( - log Prob )

average model cost

- log Prob

average number of hypotheses per sentence

10^2 10^3 10^4 10^5 10^6

speed ++

quality ++

full-integration
cube pruning
tested on our faithful clone of Pharaoh

32 times faster
Huang and Chiang

Forest Rescoring

Speed vs. Search Quality

tested on our faithful clone of Pharaoh

( - log Prob )

average model cost

full-integration
cube pruning

average number of hypotheses per sentence

same parameters

speed ++

quality ++

32 times faster
Speed vs. Translation Accuracy

The graph illustrates the relationship between the average number of hypotheses per sentence and the BLEU score. The curve for Pharaoh shows an increase in BLEU score as the average number of hypotheses increases, reaching a peak and then plateauing. The curves for full-integration and cube pruning also show an initial increase before reaching a plateau, indicating that there is an optimal number of hypotheses for maximizing translation accuracy.

- **speed ++**: Indicates an increase in speed.
- **quality ++**: Indicates an increase in quality.
Huang and Chiang

Forest Rescoring

Speed vs. Translation Accuracy

![Graph showing the relationship between BLEU score and the average number of hypotheses per sentence, with three lines representing different methods: Pharaoh, full-integration, and cube pruning. The graph indicates that Pharaoh is approximately 100 times faster than the other methods.]
Conclusions

- forest-rescoring: cube pruning and cube growing
  - on-the-fly rescoring using $k$-best parsing
  - applicable to both phrase- and syntax-based systems
  - significant speed-up against conventional beam search
- general technique for reducing search spaces
  - effectiveness depends on scale of non-monotonicity
- future work
  - forest-reranking: parsing with non-local features
Thanks!

try out Cubit

a cube pruning decoder for phrase-based translation

www.cis.upenn.edu/~lhuang3/cubit/