Translation as Weighted Deduction

Adam Lopez
University of Edinburgh
Moses
Koehn et al., ACL 2007

Hiero
Chiang, CL 2007
Moses
Koehn et al., ACL 2007

Hiero
Chiang, CL 2007
Moses
Koehn et al., ACL 2007
30.7

Hiero
Chiang, CL 2007
32.6

Lopez, Coling 2008
Moses
Koehn et al., ACL 2007

- phrase-based
- 15 features
- stack decoding

30.7

Hierro
Chiang, CL 2007

- hierarchical phrase-based
- 5 features
- cube pruning

32.6

Lopez, Coling 2008
Moses
Koehn et al., ACL 2007

- rules
  - phrase-based
  - 15 features
  - stack decoding

Hiero
Chiang, CL 2007

- parameters
  - hierarchical phrase-based
  - 5 features
  - cube pruning

- search

30.7

32.6

Lopez, Coling 2008
rules

parameters

search

- phrase-based
- 15 features
- stack decoding

- hierarchical phrase-based
- 5 features
- cube pruning
rules

phrase-based

5 features

cube pruning

parameters

hierarchical phrase-based

5 features

cube pruning

search
Adam Lopez

rules

parameters

search

phrase-based

5 features

cube pruning

synchronous TAG

5 features

cube pruning
This talk is not about

How to improve your BLEU score by 1.9.
This talk is about

**Building** and **analyzing** translation models and algorithms in a modular way.
rules

parameters

search

phrase-based

15 features

stack decoding
rules

phrase-based

parameters

15 features

search

stack decoding

deductive logic
rules  
parameters  
search  

phrase-based  
15 features  
stack decoding  

deductive logic  
semiring
北 风 呼啸
<table>
<thead>
<tr>
<th>北</th>
<th>风</th>
<th>呼啸</th>
</tr>
</thead>
<tbody>
<tr>
<td>北 /north</td>
<td>风 /wind</td>
<td>呼啸 /whistles</td>
</tr>
<tr>
<td>北 /northerly</td>
<td>风 /winds</td>
<td>呼啸 /strong</td>
</tr>
</tbody>
</table>

- word-to-word translation
- no reordering
north wind whistles  northerly wind whistles
north wind strong   northerly wind strong
north winds whistles northerly winds whistles
north winds strong   northerly winds strong

notice: complexity is $O(2^L)$ for sentence length $L$
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>北</td>
<td>风</td>
<td>呼啸</td>
</tr>
<tr>
<td>north</td>
<td>wind</td>
<td>whistles</td>
</tr>
<tr>
<td>northerly</td>
<td>winds</td>
<td>strong</td>
</tr>
</tbody>
</table>

Complexity is $O(L)$ for sentence length $L$. 

Diagram: 

```
  north  wind  whistle  
  northerly  winds  strong  
```
north northerly winds whistle strong
north

northerly

wind

winds

whistle

strong
\[ R(\text{north} / \text{wind}) \]
$R(\text{north}/\text{wind})$

- [0] north
- [1] wind
- [2] whistle
- [3] strong

Northerly winds whistle strongly.
\[ R(\vec{wind}/wind) \]

\[ R(f_{i+1}/e_j) \]
$$R(\overline{\text{wind}})$$

$$R(f_{i+1}/e_j)$$

0. north

1. northerly winds

2. whistle

3. strong
Determine complexity from inspection

McAllester, Proc. Static Analysis 1999
\[
\begin{bmatrix}
[i] \quad R(f_{i+1}/e_j) \\
[i + 1]
\end{bmatrix}
\]

Viterbi: \(\langle [0, 1], \text{max}, \times \rangle\)  
Boolean: \(\langle \{\top, \bot\}, \cup, \cap \rangle\)  
sum: \(\langle [0, 1], +, \times \rangle\)  
Reverse (outside) values

Compute many quantities on same graph  
Goodman, CL 1999
\[
\begin{bmatrix}
 i \\
\end{bmatrix}
\begin{bmatrix}
 R(f_{i+1}/e_j) \\
\end{bmatrix}
\begin{bmatrix}
 i + 1 \\
\end{bmatrix}
\]

Expectation semiring
Eisner 2002

Approximation semiring
Gimpel & Smith 2009

north
wind
whistle

northerly
winds
strong

Compute many quantities on same graph
Goodman, CL 1999
Basic Idea

• Supply a logic and a semiring, get a complete algorithm.

• Does it work for most translation models?
\[
\frac{[i'', V]}{[i', V \lor 0^i 1^{i'-i} 0^{I-i}]} \frac{R(f_{i+1} \ldots f_{i''}/e_{j} \ldots e_{j'})}{V \land 0^i 1^{i'-i} 0^{I-i}'} = 0^I, |i - i''| \leq d
\]
previous coverage vector phrase pair distortion limit

\[ V \lor 0^i 1^{i'} - i 0^{I - i'} = 0^I, \quad |i - i''| \leq d \]
Phrase-based Models
Phrase-based Models

Max distortion $d$
see, e.g. Moore & Quirk 2007

Window length $d$
Moses (Hoang & Koehn, pc)

First $d$ uncovered
see, e.g. Tillman & Ney 2003,
Zens & Ney 2004
Phrase-based Models

Max distortion $d$
see, e.g. Moore & Quirk 2007

\[
\begin{align*}
[i, V] & \frac{R(f_{i+1} \ldots f_i/e_j \ldots e_{j'})}{[i', V \lor 0^i1'i'0^{-}i'-1]} \land 0^i1'i'0^{-}i' = 0^I, |i - i''| \leq d \\
[i', C] & \frac{R(f_{i+1} \ldots f_i/e_j \ldots e_{j'})}{[i', C \ll i' - i]} \land 1'i' - i0^d - i' + i = 0^d, i' - i \leq d \\
[i', C] & \frac{R(f_{i+1} \ldots f_i/e_j \ldots e_{j'})}{[i, C \lor 0^i1'i'0^{-}i'-10^d-i''+i]} \land 0^i1'i'' - i0^d-i'' + i = 0^d, i'' - i \leq d \\
\end{align*}
\]

Window length $d$
Moses (Hoang & Koehn, pc)

\[
\begin{align*}
[i, C] & \frac{R(f_{i+1} \ldots f_i/e_j \ldots e_{j'})}{[i, C \ll i' - i]} \land 1'i'' - i0^d - i' + i = 0^d, i'' - i \leq d \\
[i, C] & \frac{R(f_{i+1} \ldots f_i/e_j \ldots e_{j'})}{[i, C \lor 0^i1'i''0^{-}i'-10^d-i''+i]} \land 0^i1'i - i0^d-i'' + i = 0^d, i'' - i \leq d \\
\end{align*}
\]

First $d$ uncovered
see, e.g. Tillman & Ney 2003, Zens & Ney 2004

\[
\begin{align*}
[i, U] & \frac{R(f_{i+1} \ldots f_i/e_j \ldots e_{j'})}{[i', U - [i', i'' \lor [i'', i'' + d - |U - [i', i'']|]]]} i' > i, f_{i+1} \in U \\
[i, U] & \frac{R(f_{i+1} \ldots f_i/e_j \ldots e_{j'})}{[i, U - [i', i''] \lor [\max(U \lor i) + 1, \max(U \lor i) + 1 + d - |U - [i', i'']|]]} i' < i, [f_i, f_{i'}] \subset U \\
\end{align*}
\]
Phrase-based Models

Max distortion $d$
see, e.g. Moore & Quirk 2007

Window length $d$
Moses (Hoang & Koehn, pc)

First $d$ uncovered
see, e.g. Tillman & Ney 2003,
Zens & Ney 2004
Phrase-based Models

Max distortion $d$

see, e.g. Moore & Quirk 2007

Window length $d$

Moses (Hoang & Koehn, pc)

First $d$ uncovered

see, e.g. Tillman & Ney 2003,
Zens & Ney 2004
Phrase-based Models

Max distortion \( d \)

see, e.g. Moore & Quirk 2007

Window length \( d \)

Moses (Hoang & Koehn, pc)

First \( d \) uncovered

see, e.g. Tillman & Ney 2003, Zens & Ney 2004
Phrase-based Models

Max distortion $d$
see, e.g. Moore & Quirk 2007

Window length $d$
Moses (Hoang & Koehn, pc)

First $d$ uncovered
see, e.g. Tillman & Ney 2003,
Zens & Ney 2004

d = 3
Phrase-based Models

Max distortion $d$
see, e.g. Moore & Quirk 2007

Window length $d$
Moses (Hoang & Koehn, pc)

First $d$ uncovered
see, e.g. Tillman & Ney 2003,
Zens & Ney 2004
Phrase-based Models

Max distortion $d$
see, e.g. Moore & Quirk 2007

Window length $d$
Moses (Hoang & Koehn, pc)

First $d$ uncovered
see, e.g. Tillman & Ney 2003, Zens & Ney 2004

$d = 3$

$O(n^3 d^2)$

$O(n d^2 2^d)$

$O(n d \binom{n}{d+1})$
Phrase-based Models

These models are not the same.

- Each can generate translations that the other cannot (regardless of \( d \)).
- Different complexities.
- Reported results will be impossible to replicate with your (different) strategy.
Good News

- Most translation models are a few lines of deductive logic.
- Computation of any semiring for free.
Good News

- Most translation models are a few lines of deductive logic.
- Computation of any semiring for free.
- You might conclude: give a logic and a semiring, get a complete algorithm.
Result

• Given:
  • A logic
  • A semiring

• Get: a complete algorithm
Bad News

- Our models use non-local features.
- We need approximate search algorithms (and we need to be able to tweak them).
Non-local features
Non-local features
Non-local features

\[
\begin{bmatrix}
e_{q}, \ldots, e_{q+n-2}
\end{bmatrix}
R
\begin{bmatrix}
e_{q}, \ldots, e_{q+n-1}
\end{bmatrix}
\begin{bmatrix}
e_{q+1}, \ldots, e_{q+n-1}
\end{bmatrix}
\]
Non-local features
Non-local features
Non-local features
Non-local features
Non-local features

minimal logic

\[
[i] \quad R(f_{i+1} \ldots f_{i'}/e_j \ldots e_{j'})
\]
\[\quad [i']\]
Non-local features

minimal logic

$$\left[ i \right] \frac{R(f_{i+1}...f_{i'}/e_{j}...e_{j'})}{[i']}$$

complete logic

$$\left[ i, e_{j-n+1}, ..., e_{j-1} \right] \frac{R(f_{i+1}...f_{i'}/e_{j}...e_{j'})R(e_{j-n+1}, ..., e_{j})...R(e_{j'-n+1}...e_{j'})}{[i', e_{j'-n+2}...e_{j'}]}$$
Non-local features

minimal logic

\[ R(f_{i+1}...f_i'/e_j...e_{j'}) \]

complete logic

\[ R(f_{i+1}...f_i'/e_j...e_{j'})R(e_{j-n+1}, ..., e_j)...R(e_{j'-n+1}...e_{j'}) \]
Deductive logics provide useful tools to manipulate search algorithms

PRODUCT (Cohen et al. ICLP 2009)

Fold-Unfold (Eisner & Blatz 2006; Johnson 2007)
Result

- Given:
  - A complete logic
  - A semiring
- Get: a complete algorithm
Result

- Given:
  - A complete logic
  - A semiring

- Get: a complete algorithm

- Problem: how to deal with approximate search?
Search
Search

stack decoding
Koehn 2004
Search
Search
Search
Search
Search
Search
Result

Given:
- A complete logic
- A semiring
- A stack predicate
- Pruning parameters

Get: a complete algorithm
Stack Pruning Effects

Window length $d$

number of items

sentence length
Stack Pruning Effects

- Sentence length
- Number of items retained in stacks

Window length $d$
Stack Pruning Effects

First $d$ uncovered
retained in stacks

number of items

sentence length
Search
$R(f_{i+1}/e_j)^{0.7}$

0.4

0.3

0.2
Search

\[ R(f_{i+1}/e_j) \]

\[ 0.7 \quad 0.2 \quad 0.1 \]

<table>
<thead>
<tr>
<th>0.4</th>
<th>0.28</th>
<th>0.08</th>
<th>0.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.21</td>
<td>0.06</td>
<td>0.03</td>
</tr>
<tr>
<td>0.2</td>
<td>0.14</td>
<td>0.04</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Search

\[ R(f_{i+1}/e_j) = \begin{pmatrix} 0.28 & 0.08 & 0.04 \\ 0.21 & 0.06 & 0.03 \\ 0.14 & 0.04 & 0.02 \end{pmatrix} \times \begin{pmatrix} 0.5 & 0.4 & 0.5 \\ 0.9 & 0.3 & 0.6 \\ 0.5 & 0.3 & 0.4 \end{pmatrix} \]
$R(f_{i+1}/e_j) = 0.7$

$R(e_{j-n+1}, \ldots, e_j)$

\[
\begin{array}{ccc}
0.4 & 0.28 & 0.08 & 0.04 \\
0.3 & 0.21 & 0.06 & 0.03 \\
0.2 & 0.14 & 0.04 & 0.02 \\
\end{array}
\]

\[
\begin{array}{ccc}
0.5 & 0.4 & 0.5 \\
0.9 & 0.3 & 0.6 \\
0.5 & 0.3 & 0.4 \\
\end{array}
\]

\[
= \begin{array}{ccc}
0.14 & 0.03 & 0.02 \\
0.18 & 0.02 & 0.02 \\
0.7 & 0.01 & 0.01 \\
\end{array}
\]
Cube Pruning
Chiang, 2007; Huang & Chiang, 2007

\[ R(f_{i+1}/e_j) \]

\[ R(e_{j-n+1}, \ldots, e_j) \]

\[
\begin{array}{ccc}
0.4 & 0.28 & 0.08 & 0.04 \\
0.3 & 0.21 & 0.06 & 0.03 \\
0.2 & 0.14 & 0.04 & 0.02 \\
\end{array}
\times
\begin{array}{ccc}
0.5 & 0.4 & 0.5 \\
0.9 & 0.3 & 0.6 \\
0.5 & 0.3 & 0.4 \\
\end{array}
= \begin{array}{ccc}
0.14 & 0.03 & 0.02 \\
0.18 & 0.02 & 0.02 \\
0.7 & 0.01 & 0.01 \\
\end{array}
\]
Cube Pruning
Chiang, 2007;
Huang & Chiang, 2007
Cube Pruning
Chiang, 2007;
Huang & Chiang, 2007
Result

- Given:
  - A minimal logic
  - A complete logic
  - A semiring
  - Pruning parameters
- Get: a complete algorithm
Conclusion

- Translation can easily be cast in the deductive framework.
- Analysis reveals inconsistencies.
- Modify models with logic transforms.
- Easy to describe non-local features.
- Search strategies can be incorporated into deductive systems.
Future Work

- Other approximate search strategies.
- Modular implementation.
- Exploration of novel models.
Thanks

Phil Blunsom
Chris Callison-Burch
Chris Dyer
Hieu Hoang
Martin Kay
Philipp Koehn
Josh Schroeder
Lane Schwartz