Translation Model
Search Spaces

Adam Lopez
University of Edinburgh

Joint work with Michael Auli,
Hieu Hoang and Philipp Koehn
idea
idea  implementation
idea -> implementation -> experiment

32.6
idea -> implementation -> experiment

32.6
Development Cycle for MT Research

1. Define goal
2. Build prototype
3. Error Analysis
4. Try to fix 'next' error
5. Is it better?
   - NO: Ignore it
   - YES: Use it

Moses
Koehn et al., ACL 2007

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

Hieroo
Chiang, CL 2007

30.7 32.6

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

- phrase-based
- 15 features
- stack decoding
- C++

30.7

Hiero
Chiang, CL 2007

- hierarchical phrase-based
- 5 features
- cube pruning
- python/pyrex

32.6

Lopez, Coling 2008
This talk is *not* about

How to improve your BLEU score by 1.9.
This talk is about **Weighted deduction**: a language for **building** and **analyzing** models and algorithms in a modular way.
This talk is about

**Weighted deduction**: a language for **building** and **analyzing** models and algorithms in a modular way.

Lopez, *Translation as Weighted Deduction*, EACL 2009
Insights

• Your idea of phrase-based translation is different from your neighbor’s.

• We can derive a general recipe for approximate search.
Empirical Results

- Common pruning heuristics are very bad for your system’s upper bound accuracy.
- Practical phrase-based and hierarchical phrase-based systems have nearly identical search spaces.
北 风 呼啸
北 風 呼嘯

- word-to-word translation
- no reordering
- no language model
<table>
<thead>
<tr>
<th></th>
<th>Chinese</th>
<th>English</th>
<th>Pinyin</th>
</tr>
</thead>
<tbody>
<tr>
<td>北风呼啸</td>
<td>北/north (.8)</td>
<td>风/wind (.6)</td>
<td>呼啸/whistles (.7)</td>
</tr>
<tr>
<td>北/northerly (.2)</td>
<td>风/winds (.4)</td>
<td>呼啸/strong (.3)</td>
<td></td>
</tr>
</tbody>
</table>

- word-to-word translation
- no reordering
- no language model
北 风 呼啸

北 / north (.8) 风 / wind (.6) 呼啸 / whistles (.7)
北 / northerly (.2) 风 / winds (.4) 呼啸 / strong (.3)

north wind whistles (.34) northerly wind whistles (.08)
north wind strong (.14) northerly winds whistles (.06)
north winds whistles (.22) northerly winds strong (.024)
north winds strong (.096) northerly wind strong (.04)
北 风 呼啸

北 / north (.8)  风 / wind (.6)  呼啸 / whistles (.7)
北 / northerly (.2)  风 / winds (.4)  呼啸 / strong (.3)

north wind whistles (.34)  northerly wind whistles (.08)
north wind strong (.14)  northerly winds whistles (.06)
north winds whistles (.22)  northerly winds strong (.024)
north winds strong (.096)  northerly wind strong (.04)

notice: complexity is $O(2^L)$ for sentence length $L$
北 风 呼啸

北/north (.8)  风/wind (.6)  呼啸/whistles (.7)
北/northerly (.2)  风/winds (.4)  呼啸/strong (.3)

north (.8)  wind (.6)  whistle (.7)
northerly (.2)  winds (.4)  strong (.3)
北 风 呼啸

北 / north (.8)  风 / wind (.6)  呼啸 / whistles (.7)

北 / northerly (.2)  风 / winds (.4)  呼啸 / strong (.3)

complexity is $O(L)$ for sentence length $L$
complexity is $O(L)$ for sentence length $L$
north (.8) northerly (.2)
wind (.6) winds (.4)
whistle (.7) strong (.3)
north (0.8) wind (0.6) whistle (0.7)

northerly (0.2) winds (0.4) strong (0.3)
north (.8)  wind (.6)  whistle (.7)

northerly (.2)  winds (.4)  strong (.3)
\[ i \ R(f_{i+1}/e_j) \]
\[ R(f_{i+1}/e_{j}) \]

Diagram:
- Node [0]: north (0.8)
- Node [1]: wind (0.6)
- Node [2]: whistle (0.7)
- Node [3]:

- Edge: northerly (0.2)
- Edge: winds (0.4)
- Edge: strong (0.3)
\[
\begin{align*}
[i] & \quad R(f_{i+1}/e_j) \\
[i + 1] & \quad [1] \quad R(\overline{\Delta}/\text{wind}) \\
[0] & \text{north (.8)} \\
[1] & \text{wind (.6)} \\
[2] & \quad \text{whistle (.7)} \\
[3] & \text{northerly (.2)} \\
& \text{winds (.4)} \\
& \text{strong (.3)}
\end{align*}
\]
Why is this good?
Why is this good?

\[
\begin{align*}
\left[ i \right] & \quad \frac{R(f_{i+1}/e_j)}{[i + 1]} \\
\end{align*}
\]
Why is this good?

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

Determine complexity from inspection

McAllester, Proc. Static Analysis 1999
Why is this good?

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

\( [i + 1] \)

\( i \) ranges over sentence length

Determine complexity from inspection

McAllester, Proc. Static Analysis 1999
Why is this good?

$\left[ i \right] \frac{R(f_{i+1}/e_j)}{[i + 1]}$
Why is this good?

$$\begin{array}{c}
\frac{R(f_{i+1}/e_j)}{i+1} \\
[\text{Compute many quantities on same graph}]
\end{array}$$

Goodman, CL 1999
Why is this good?

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

Viterbi: \langle [0, 1], \text{max, } \times \rangle  

sum: \langle [0, 1], +, \times \rangle  

Boolean: \langle \{\top, \bot\}, \cup, \cap \rangle  

Compute many quantities on same graph  

Goodman, CL 1999
Why is this good?

\[
\begin{array}{c}
[i] \quad R(f_{i+1}/e_j) \\
[i + 1]
\end{array}
\]

Expectation semirings  (Eisner, ACL 2002)
Approximation semirings  (Gimpel & Smith, EACL 2009)

Compute many quantities on same graph
Goodman, CL 1999
Why is this good?

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

Separate search space from search logic

Kay, 1986
Phrase-based Models

\[
\begin{align*}
[i'', V] & \quad R(f_{i+1} \ldots f_{i'}/e_j \ldots e_{j'}) \\
[i', V \lor 0^i] & \quad r_{1' - i} 0^{I - i'} = 0^I, |i - i''| \leq d
\end{align*}
\]

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

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

\[
\begin{align*}
[i, U] & \quad R(f_{i'} \ldots f_{i''}/e_j \ldots e_{j'}) \\
[i'', U - [i', i''] \lor [i'', i'' + d - |U - [i', i'']|]] & \quad i' > i, f_{i+1} \in U
\end{align*}
\]

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

without distortion

item form: \[ \{0, 1\}^I \]

\[
\begin{align*}
\left[ V \right] & \frac{R(f_{i+1} \ldots f_{i'}/e_j \ldots e_{j'})}{\left[ V \lor 0^i1^{i'-i}0^{I-i'} \right]} \left[ V \land 0^i1^{i'-i}0^{I-i'} \right] = 0^I \\
\text{complexity: } & O(2^II^2)
\end{align*}
\]
Phrase-based Models

What does the distortion limit do?
Phrase-based Models

What does the distortion limit do?
Phrase-based Models

What does the distortion limit do?

dl=3
Phrase-based Models

What does the distortion limit do?

dl=3

Maximum Distortion $d$ Strategy

see, e.g. Moore & Quirk, MT Summit 2007

complexity: $O(I^3d^2)$
Phrase-based Models

What does the distortion limit do?
Phrase-based Models

What does the distortion limit do?

dl=3
Phrase-based Models

What does the distortion limit do?

\[ dl = 3 \]

Window Length \( d \) Strategy
implemented in Moses

item form: \([i, \{0, 1\}^d]\)  complexity: \(O(d^2 2^d I)\)
Phrase-based Models

What does the distortion limit do?
Phrase-based Models

What does the distortion limit do?

dl=3
Phrase-based Models

What does the distortion limit do?

dl=3

First $d$ Uncovered Words Strategy

see, e.g. Tillman & Ney, CL 2003, Zens & Ney, HLT-NAACL 2004

complexity: $O\left(dI\left(\frac{I}{d+1}\right)\right)$
Phrase-based Models

These models are not the same.

• Each can generate translations that the other cannot (regardless of \(d\)).

• Complexities are different; implications for search.

• Reported results may be impossible to replicate with your (different) strategy.
Language Models
Language Models

This won’t work:

north (.8) wind (.6) whistle (.7)
northerly (.2) winds (.4) strong (.3)
Language Models

This won’t work:

north (.8) wind (.6) whistle (.7)
northerly (.2) winds (.4) strong (.3)

We solve this using the PRODUCT transform
Cohen et al., ICLP 2009
Language Models

Step 1: Define a logic for the language model.

\[
\begin{bmatrix}
e_q, \ldots, e_{q+n-2}
\end{bmatrix} \cdot R\left(\begin{bmatrix}
e_q, \ldots, e_{q+n-1}
\end{bmatrix}\right)
\begin{bmatrix}
e_{q+1}, \ldots, e_{q+n-1}
\end{bmatrix}
\]
Language Models

Step 1: Define a logic for the language model.

\[
\begin{align*}
[e_q, \ldots, e_{q+n-2}] & \quad R(e_q, \ldots, e_{q+n-1}) \\
[e_{q+1}, \ldots, e_{q+n-1}] & 
\end{align*}
\]

Step 2: Modify translation logic to write words.

\[
\begin{align*}
[i, u_e \bullet] & \quad R(f_{i+1} \ldots f_{i'} / e_j v_e) \\
[i', e_j \bullet v_e] & \\
[i, u_e \bullet e_j v_e] & \\
[i, u_e e_j \bullet v_e] & 
\end{align*}
\]
Language Models

Step 3: Apply the PRODUCT Transform

\[
[i, u_e \bullet, e_q, \ldots, e_{q+n-2}] \quad R(f_i \ldots f_i' / e_j u_e) \quad R(e_q, \ldots, e_{q+n-1})
\]

\[
[i', e_j \bullet u_e, e_{q+1}, \ldots, e_{q+n-1}]
\]

\[
[i, u_e \bullet e_j v_e, e_q, \ldots, e_{q+n-2}] \quad R(e_q, \ldots, e_{q+n-1})
\]

\[
[i, u_e e_j \bullet v_e, e_{q+1}, \ldots, e_{q+n-1}]
\]
Language Models

Step 4: Constrain the result

\[
\begin{align*}
&[i, u_e \bullet, e_{j-n+1}, \ldots, e_{j-1}] \quad R(f_i \ldots f_{i'} / e_j v_e) \quad R(e_{j-n+2} \ldots e_j) \\
&[i', e_j \bullet v_e, e_{j-n+2}, \ldots, e_j] \\
&[i, u_e \bullet e_{i+n-1} v_e, e_i, \ldots, e_{i+n-2}] \quad R(e_{j-n+2} \ldots e_j) \\
&[i + 1, u_e e_j \bullet v_e, e_{j-n+2}, \ldots, e_j]
\end{align*}
\]
Step 4: Constrain the result

\[ [i, u_e \bullet, e_{j-n+1}, \ldots, e_{j-1}] \quad R(f_i \ldots f_{i'}/e_j v_e) \quad R(e_{j-n+2} \ldots e_j) \]

\[ [i', e_j \bullet v_e, e_{j-n+2}, \ldots, e_j] \]

\[ [i, u_e \bullet e_{i+n-1} v_e, e_i, \ldots, e_{i+n-2}] \quad R(e_{j-n+2} \ldots e_j) \]

\[ [i + 1, u_e e_j \bullet v_e, e_{j-n+2}, \ldots, e_j] \]

Notice: this logic is more articulated than in most models
Language Models

(optional) Step 5: unfold the logic

Eisner & Blatz, Formal Grammar 2006

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

\[
[i', e_{j'-n+2} \ldots e_{j'}]
\]
(optional) Step 5: unfold the logic

\[ [i, e_{j-n+1}, \ldots, e_{j-1}] \ R(f_{i+1} \ldots f_{i'}/e_{j} \ldots e_{j'})R(e_{j-n+1}, \ldots, e_{j})\ldots R(e_{j'-n+1} \ldots e_{j'}) \]

\[ [i', e_{j'-n+2} \ldots e_{j'}] \]

Result is the complete logic.
Language Models

(optional) Step 5: unfold the logic

Eisner & Blatz, Formal Grammar 2006

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

\[
[i', e_{j'-n+2} \ldots e_{j'}]
\]

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

\[
[i']
\]
Language Models

(optional) Step 5: *unfold* the logic

Eisner & Blatz, Formal Grammar 2006

\[
\begin{align*}
&[i, e_{j-n+1}, \ldots, e_{j-1}] \\
&R(f_{i+1} \ldots f_{i'} / e_{j} \ldots e_{j'}) R(e_{j-n+1}, \ldots, e_{j}) \ldots R(e_{j'-n+1} \ldots e_{j'}) \\
&[i', e_{j'-n+2} \ldots e_{j'}]
\end{align*}
\]

Notice:

\[
\begin{align*}
&[i] \\
&R(f_{i+1} \ldots f_{i'}/e_{j} \ldots e_{j'}) \\
&[i']
\end{align*}
\]
Search Algorithms

- Cube pruning (Chiang 2007) uses minimal logic to drive complete logic.
- Stack pruning uses predicates on items to group sets of items for pruning.
- Dijkstra/Knuth: exact algorithms.
ruleset

parameters

search

- hierarchical phrase-based
- 5 features
- cube pruning
model error

exact search yields an incorrect answer

Germann et al., ACL 2001
ruleset
parameters
search

hierarchical phrase-based
5 features
cube pruning

model error
search error

inexact search misses model optimum

Germann et al., ACL 2001
ruleset
parameters
search

hierarchical phrase-based
5 features
cube pruning

model error
search error

???
ruleset
does not contain correct answer

parameters
5 features

search
cube pruning

hierarchical phrase-based

induction error
model error
search error
Induction Error

Question: Given a ruleset and a sentence pair, does the search space contain the sentence pair?
Induction Error

Question: Given a ruleset and a sentence pair, does the search space contain the sentence pair?

Assumption: Reference is ground truth.
Induction Error

Question: Given a ruleset and a sentence pair, does the search space contain the sentence pair?

Assumption: Reference is ground truth.

This assumption is too strong, but it correlates with metrics that we care about. If a model systematically fails to generate reference, model is probably no good.
Experiments

- GIZA++, grow-diag-final, MERT
- Train: French-English Europarl (40Mw)
- Dev: WMT 2008 dev (2K sentences)
- Test: WMT 2008 test (2K sentences)
- Systems: Moses, Hiero
Experiment 1

source

Moses output

Hiero output

variables: distortion limit, translation option limit
Experiment 1

variables: distortion limit, translation option limit
Results

coverage (%) vs. translation option limit

distortion limit

coverage (%) [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
translation option limit [20, 50, 100, 200, 400, 800, ∞]
Analysis

Rank of translations of *problème* according to $p(e \mid f)$.

- problem (1)
- challenge (25)
- obstacle (44)
- dilemma (105)
Example

... il n'peut y avoir de délai transitoire en matière de respect des règles démocratiques.
Example

... il n’peut y avoir de délai transitoire en matière de respect des règles démocratiques.
	here can be no transitional period for complying with democratic rules.
Example

... il n'peut y avoir de délai transitoire en matière de respect des règles démocratiques.

there can be no transitional period for complying with democratic rules.

there can be no transitional period in the field of democratic rules.
**Conclusion:** pruning target phrases on the basis of local features severely harms upper bound accuracy.
Model Differences

Literature posits that differences are **structural**:

- Discontinuous vs. continuous phrases
- Lexically informed vs. arbitrary reordering
- Non-local vs. local deletion of words.
- Language theoretic (SCFG vs. FST)

In other words, a function of rule sets and logics.
Experiment 2

- source
- reference
- Moses output
- Hiero output
Experiment 2

source

Moses

reference

Moses output

Hiero output
Experiment 2

Moses

source

reference

Moses output

Hiero output

Zollman et al., Coling 2008
Experiment 2

source

Hiero

Moses output

Hiero output

reference
Results

For French-English, German-English, respectively

- Hiero → Moses (99.4, 97.7)
- Moses → Hiero (97.6, 97.6)
Conclusion

These differences matter less than we think:

- Discontinuous vs. continuous phrases
- Lexically informed vs. arbitrary reordering
- Non-local vs. local deletion of words.
- Language theoretic (SCFG vs. FST)

These decoders simply rank hypotheses differently.

for counterpoint see: Lopez, Coling 2008
Looking Ahead

Modular architecture and decomposable error analysis for rapid system development

- model
- parameters
- search

- phrase-based
- 15 features
- stack decoding
- C++

induction error
model error
search error
speed
Looking Ahead

The Statistical Machine Translation Algorithms Library: implementation of generic algorithms including a reference decoder.

point your browser to:
http://github.com/alopez
Weighted deduction: a language for building and analyzing models and algorithms in a modular way.
Insights

• Your idea of phrase-based translation is different from your neighbor’s.

• We can derive a general recipe for approximate search.
Empirical Results

- Common pruning heuristics are very bad for your system’s upper bound accuracy.
- Practical phrase-based and hierarchical phrase-based systems have nearly identical search spaces.
More Information


• Translation as Weighted Deduction. To appear in EACL 2009.

• Papers available from my website — http://homepages.inf.ed.ac.uk/alopez
Thanks

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