Statistical & Phrase Based Machine Translation

Rebecca Knowles
Today

• Word-based and phrase-based models
• A very fast introduction to PBMT/SMT
  – Entire courses can/have been taught on this topic
  – We’ll cover several topics at a high level
• These and similar approaches are among the dominant MT paradigms of the last ~20 years
Outline

• Language models
  – $n$-grams
• Translation models
  – Word-level
  – Alignments
  – EM Algorithm
  – Phrases
• Decoding
Whereas recognition of the inherent dignity and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world,

Whereas disregard and contempt for human rights have resulted in barbarous acts which have outraged the conscience of mankind, and the advent of a world in which human beings shall enjoy freedom of speech and belief and freedom from fear and want has been proclaimed as the highest aspiration of the common people,

Whereas it is essential, if man is not to be compelled to have recourse, as a last resort, to rebellion against tyranny and oppression, that human rights should be protected by the rule of law,

Whereas it is essential to promote the development of friendly relations between nations,


Collect data
Preprocess data
Design model(s)/features
Train model(s)
Tune model(s)
Evaluate model(s)
Run model(s)
Review (from reading)

The next several slides draw on the reading:

A Statistical MT Tutorial Workbook
(Knight, 1999)
Basic Probability

• $P(e)$: *A priori* probability (chance that $e$ happens)
• $P(f|e)$: *Conditional* probability (chance of $f$ given $e$)
• $P(e, f)$: *Joint* probability (chance of $e$ and $f$ both happening)
Basic Probability

• \( P(e) \): What is the chance of \( e \) being an English sentence?
• \( P(f|e) \): What is the chance of the translator producing \( f \) given the English sentence \( e \)?
Review (from reading)

Put these words in the correct English order:

have programming a seen never I language better
Review (from reading)

Put these words in the correct English order:

have programming a seen never I language better
I have never seen a better programming language

• What kind of knowledge are you applying here?
• Do you think a machine could do this job?
• Can you think of a way to automatically test how well a machine is doing, without a lot of human checking?
Language Models (LM)

\[ P(e) \]

\[ P(\text{the cat purrs}) > P(\text{purrs cat the}) \]
Intuition

Decipher this student’s late note:
(one word per column)

<table>
<thead>
<tr>
<th>LIE</th>
<th>CHARM</th>
<th>CODE</th>
<th>SOIL</th>
<th>ROUT</th>
<th>WAKE</th>
<th>HE</th>
<th>US</th>
<th>MOVING</th>
</tr>
</thead>
<tbody>
<tr>
<td>MY</td>
<td>SOLD</td>
<td>CIRCLE</td>
<td>DID</td>
<td>HOT</td>
<td>MAKE</td>
<td>I’LL</td>
<td>IS</td>
<td>HAVING</td>
</tr>
<tr>
<td>GUY</td>
<td>CLAM</td>
<td>SHUTE</td>
<td>RAID</td>
<td>RIOT</td>
<td>SALE</td>
<td>ME</td>
<td>UGH</td>
<td>READING</td>
</tr>
<tr>
<td>BYE</td>
<td>BEAM</td>
<td>CLOCK</td>
<td>RISK</td>
<td>NOT</td>
<td>WANE</td>
<td>BE</td>
<td>UP</td>
<td>MORNING</td>
</tr>
<tr>
<td></td>
<td>ALARM</td>
<td>VISIT</td>
<td>MUST</td>
<td></td>
<td></td>
<td>SEE</td>
<td></td>
<td>LOVING</td>
</tr>
</tbody>
</table>
Intuition

Decipher this student’s late note:
(one word per column)

<table>
<thead>
<tr>
<th>LIE</th>
<th>CHARM</th>
<th>CODE</th>
<th>SOIL</th>
<th>ROUT</th>
<th>WAKE</th>
<th>HE</th>
<th>US</th>
<th>THIS</th>
<th>MOVING</th>
</tr>
</thead>
<tbody>
<tr>
<td>MY</td>
<td>SOLD</td>
<td>CIRCLE</td>
<td>DID</td>
<td>HOT</td>
<td>MAKE</td>
<td>I’LL</td>
<td>IS</td>
<td>TAXIS</td>
<td>HAVING</td>
</tr>
<tr>
<td>GUY</td>
<td>CLAM</td>
<td>SHUT</td>
<td>RAID</td>
<td>RIOT</td>
<td>SALE</td>
<td>ME</td>
<td>UGH</td>
<td>THAI</td>
<td>READING</td>
</tr>
<tr>
<td>BYE</td>
<td>BEAM</td>
<td>CLOCK</td>
<td>RISK</td>
<td>NOT</td>
<td>WANE</td>
<td>BE</td>
<td>UP</td>
<td>TEAR</td>
<td>MORNING</td>
</tr>
<tr>
<td>ALARM</td>
<td>DREAM</td>
<td>VISIT</td>
<td>MUST</td>
<td></td>
<td></td>
<td>SEE</td>
<td></td>
<td></td>
<td>LOVING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
n-grams

A sequence of n words is an n-gram.
unigram: “word”
bigram: “a bigram”
trigram: “read a trigram”
4-gram: “this is an example”
...
Trigram Language Model

\[ b(z|x, y) = \frac{\text{count}("x y z")}{\text{count}("x y")} \]

\[ P(\text{the cat purrs}) \approx b(\text{the} | \text{BOS, BOS}) \]
\[ \times b(\text{cat} | \text{the, BOS}) \]
\[ \times b(\text{purrs} | \text{the, cat}) \]
\[ \times b(\text{EOS} | \text{cat, purrs}) \]

We can estimate counts from a large monolingual corpus!
What am I glossing over?

- Smoothing (see A Statistical MT Tutorial Workbook (Knight, 1999) section 11)
- Markov assumption
- Chain rule of probability
- Neural LMs
- See these slides for much more information: http://statmt.org/book/slides/07-language-models.pdf
Translation Models

\[ P(f|e) \]

\[ P(\text{Haus}|\text{house}) > P(\text{Haus}|\text{building}) > P(\text{Haus}|\text{the}) \]
\[ P(f|e) \]

• How can we find out \( P(f|e) \)?
P(f|e)

• How can we find out P(f|e)? Alignments.

das Haus ist klein
the house is small

Alignments

• If we had word-aligned text, we could easily estimate $P(f|e)$.
  – But we don’t usually have alignments, and they are expensive to produce by hand…
• If we had $P(f|e)$ we could produce alignments automatically.
• Chicken vs. Egg…
IBM Model 1 (1993)

- Lexical Translation Model
- Word Alignment Model

\[
P(a, f | e) = \prod_{j=1}^{m} t(f_j | e_{a_j})
\]

\[
P(f | e) = \sum_a P(a, f | e)
\]
Simplified IBM 1

• We’ll work through an example with a simplified version of IBM Model 1
• Figures and examples are drawn from A Statistical MT Tutorial Workbook, Section 27, (Knight, 1999)
• Simplifying assumption: each source word must translate to exactly one target word
Data

Two sentence pairs:

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>b c</td>
<td>x y</td>
</tr>
<tr>
<td>b</td>
<td>y</td>
</tr>
</tbody>
</table>
Possible Alignments

b c
\|--|--|
x y

b c
\|--|--|
x y

b
\|--|
y
EM Algorithm

A. Initialize model parameters.
B. **Expectation:** Assign probabilities to data.
C. **Maximization:** Re-estimate model from data.
D. Repeat B and C until convergence.

This is what we might describe as “training” or “learning” the model’s parameters.
EM Step 1: Initialize

Set parameter values uniformly.
All translations have an equal chance of happening.

\[
\begin{align*}
  t(x|b) &= 1/2 \\
  t(y|b) &= 1/2 \\
  t(x|c) &= 1/2 \\
  t(y|c) &= 1/2
\end{align*}
\]
Step 2: Compute \( P(a, f|e) \)

For all alignments, compute \( P(a, f|e) \)

Remember:

\[
P(a, f|e) = \prod_{j=1}^{m} t(f_j|e_{a_j})
\]

- \( t(x|b) = 1/2 \)
- \( t(y|b) = 1/2 \)
- \( t(x|c) = 1/2 \)
- \( t(y|c) = 1/2 \)

\[
P(a, f|e) = 1/2
\]
Step 2: Compute $P(a,f|e)$

For all alignments, compute $P(a,f|e)$

Remember:

$$P(a, f|e) = \prod_{j=1}^{m} t(f_j|e_{a_j})$$

$t(x|b) = 1/2$

$t(y|b) = 1/2$

$t(x|c) = 1/2$

$t(y|c) = 1/2$

$P(a, f|e) = 1/2 \times 1/2 = 1/4$

$P(a, f|e) = 1/2 \times 1/2 = 1/4$

$P(a, f|e) = 1/2$
Step 3: Compute $P(a|e,f)$

We can compute $P(a|e,f)$ using $P(a,f|e)$: $\frac{P(a,f|e)}{\sum_a P(a,f|e)}$
Step 3: Compute $P(a|e,f)$

$$P(a|e,f) = P(a,f|e) / \sum_a P(a,f|e)$$

- $P(a,f|e) = 1/2 \times 1/2 = 1/4$
- $P(a,f|e) = 1/2 \times 1/2 = 1/4$
- $P(a,f|e) = 1/2$

$$P(a|e,f) = \frac{1/4}{2/4} = 1/2$$

$$P(a|e,f) = \frac{1/4}{2/4} = 1/2$$

$$P(a|e,f) = \frac{1/2}{1/2} = 1$$
Step 4: Collect Counts

\[ P(a|e,f) = \frac{1}{2/4} = \frac{1}{2} \]

\[ tc(x|b) = \frac{1}{2} \]

\[ tc(y|b) = \frac{1}{2} + 1 = \frac{3}{2} \]

\[ tc(x|c) = \frac{1}{2} \]

\[ tc(y|c) = \frac{1}{2} \]
Step 5: Normalize

\[ tc(x|b) = 1/2 \]
\[ tc(y|b) = 1/2 + 1 = 3/2 \]
\[ tc(x|c) = 1/2 \]
\[ tc(y|c) = 1/2 \]

\[ t(x|b) = \frac{1/2}{4/2} = 1/4 \]
\[ t(y|b) = \frac{3/2}{4/2} = 3/4 \]
\[ t(x|c) = \frac{1/2}{1} = 1/2 \]
\[ t(y|c) = \frac{1/2}{1} = 1/2 \]
Repeat Step 2

Compute $P(a, f | e)$ using new parameters:

$$P(a, f | e) = \prod_{j=1}^{m} t(f_j | e_{a_j})$$

- $t(x | b) = \frac{1/2}{4/2} = 1/4$
- $t(y | b) = \frac{3/2}{4/2} = 3/4$
- $t(x | c) = \frac{1/2}{1} = 1/2$
- $t(y | c) = \frac{1/2}{1} = 1/2$

$$P(a, f | e) = \frac{3}{4}$$
Repeat Step 2

Compute \( P(a, f|e) \) using new parameters:

\[
P(a, f|e) = \prod_{j=1}^{m} t(f_j|e_{a_j})
\]

\[
t(x|b) = \frac{1/2}{4/2} = 1/4
\]

\[
t(y|b) = \frac{3/2}{4/2} = 3/4
\]

\[
t(x|c) = \frac{1/2}{1} = 1/2
\]

\[
t(y|c) = \frac{1/2}{1} = 1/2
\]

\[
P(a, f|e) = 1/4 \times 1/2 = 1/8
\]

\[
P(a, f|e) = 3/2 \times 1/2 = 3/8
\]

\[
P(a, f|e) = 3/4
\]
Repeat Step 3: Compute $P(a|e, f)$

$$P(a|e, f) = P(a, f|e) / \sum_a P(a, f|e)$$

\[
\begin{align*}
\text{b} & \quad \text{c} \\
\text{x} & \quad \text{y} \\
\text{b} & \quad \text{c} \\
\text{x} & \quad \text{y} \\
\text{b} & \quad \text{c} \\
\text{y} \\

P(a, f|e) &= 1/4 \times 1/2 = 1/8 \\
P(a, f|e) &= 3/2 \times 1/2 = 3/8 \\
P(a, f|e) &= 3/4 \\

\text{b} & \quad \text{c} \\
\text{x} & \quad \text{y} \\
\text{b} & \quad \text{c} \\
\text{x} & \quad \text{y} \\
\text{b} & \quad \text{y} \\

P(a|e, f) &= \frac{1/8}{4/8} = 1/4 \\
\text{b} & \quad \text{c} \\
\text{x} & \quad \text{y} \\
\text{b} & \quad \text{c} \\
\text{x} & \quad \text{y} \\
\text{b} & \quad \text{y} \\

P(a|e, f) &= \frac{3/8}{4/8} = 3/4 \\

\text{b} & \quad \text{y} \\
\text{b} & \quad \text{y} \\

P(a|e, f) &= 1
Repeat Steps 4 & 5

Collect counts:

\[ tc(x|b) = \frac{1}{4} \]
\[ tc(y|b) = \frac{3}{4} + 1 = \frac{7}{4} \]
\[ tc(x|c) = \frac{3}{4} \]
\[ tc(y|c) = \frac{1}{4} \]

Normalize counts:

\[ t(x|b) = \frac{1}{8} \]
\[ t(y|b) = \frac{7}{8} \]
\[ t(x|c) = \frac{3}{4} \]
\[ t(y|c) = \frac{1}{4} \]
What is happening to $t()$?

\[
\begin{align*}
t(x|b) &= 1/2 \\
t(y|b) &= 1/2 \\
t(x|c) &= 1/2 \\
t(y|c) &= 1/2
\end{align*}
\]

\[
\begin{align*}
t(x|b) &= \frac{1/2}{4/2} = 1/4 \\
t(y|b) &= \frac{3/2}{4/2} = 3/4 \\
t(x|c) &= \frac{1/2}{1} = 1/2 \\
t(y|c) &= \frac{1/2}{1} = 1/2
\end{align*}
\]

\[
\begin{align*}
t(x|b) &= 1/8 \\
t(y|b) &= 7/8 \\
t(x|c) &= 3/4 \\
t(y|c) &= 1/4
\end{align*}
\]
What does that mean?

Which alignments are more likely to be correct?

\[
\begin{align*}
\frac{b}{c} & \quad \frac{b}{c} \\
\mid & \quad \times \\
\frac{x}{y} & \quad \frac{x}{y} \\
\frac{b}{y} & \\
\mid & \\
\frac{y}{c} &
\end{align*}
\]

\[
\begin{align*}
t(x|b) &= 1/8 \\
t(y|b) &= 7/8 \\
t(x|c) &= 3/4 \\
t(y|c) &= 1/4
\end{align*}
\]
What would happen to t()...

...if we repeated steps 2-5 many many times?
Repeat Steps 2-5 Many Times

\[ t(x|b) = 0.0001 \]
\[ t(y|b) = 0.9999 \]
\[ t(x|c) = 0.9999 \]
\[ t(y|c) = 0.0001 \]
Review of IBM Model 1 & EM

- Iteratively learned an alignment/translation model from sentence-aligned text (without “gold standard” alignments)
- Model can now be used for alignment and/or word-level translation
- We explored a simplified version of this; IBM Model 1 allows more types of alignments
What am I glossing over?

- Other IBM models, less strict alignments: *A Statistical MT Tutorial Workbook* (Knight, 1999)
Why is Model 1 insufficient?

• Why won’t this produce great translations?
• What might we also want to have in such a model?
### Phrases

#### English to Spanish

<table>
<thead>
<tr>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>María no daba una</td>
</tr>
<tr>
<td>did</td>
<td>bofetada</td>
</tr>
<tr>
<td>not</td>
<td>ala</td>
</tr>
<tr>
<td>slap</td>
<td>bruja</td>
</tr>
<tr>
<td>the</td>
<td>verde</td>
</tr>
<tr>
<td>green</td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
</tr>
</tbody>
</table>

#### Spanish to English

<table>
<thead>
<tr>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>María no daba una</td>
</tr>
<tr>
<td>did</td>
<td>bofetada</td>
</tr>
<tr>
<td>not</td>
<td>ala</td>
</tr>
<tr>
<td>slap</td>
<td>bruja</td>
</tr>
<tr>
<td>the</td>
<td>verde</td>
</tr>
<tr>
<td>green</td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
</tr>
</tbody>
</table>

#### Intersection

<table>
<thead>
<tr>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>María no daba una</td>
</tr>
<tr>
<td>did</td>
<td>bofetada</td>
</tr>
<tr>
<td>not</td>
<td>ala</td>
</tr>
<tr>
<td>slap</td>
<td>bruja</td>
</tr>
<tr>
<td>the</td>
<td>verde</td>
</tr>
<tr>
<td>green</td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
</tr>
</tbody>
</table>
Phrases

Mary did not slap the green witch

Maria no daba una bofetada a la bruja verde
What am I glossing over?

• Lots and lots of approaches to phrase extraction!
Decoding

What have we done so far?

• We can score alignments.
• We can score translations.

How do we *generate* translations?

• Decoding!
Decoding

Why can’t we just score all possible translations?

What do we do instead?
Decoding

We may have many translation options for a word or phrase. Decoding is NP-complete (can verify solutions in polynomial time, but can’t locate solutions efficiently). We use heuristics to limit the search space.

The role of the decoder is to:

• Choose “good” translation options
• Arrange them in a “good” order
How can this go wrong?

• Search doesn’t find the best translation
  – Need to fix the search
• The best translation found is not good
  – Need to fix the model
Example
Example
Search Graph

Er hat seit Monaten geplant.
What am I glossing over?

Most of the details of decoding!

Try these resources:

Textbook slides:  

Demo: http://mt-class.org/jhu/stack-decoder/
Phrase-Based SMT Overview

• Translate phrases (not just words)
• Combine information from language models, translation models, alignments, possibly more
• Need to restrict search space with heuristics (distortion limits, recombining similar hypotheses, limiting number of options to consider, etc.)