Algorithms for NLP

Machine Translation III

Yulia Tsvetkov – CMU

Slides: Philipp Koehn – JHU; Chris Dyer – DeepMind
Taylor Berg-Kirkpatrick – CMU, Dan Klein – UC Berkeley
Centauri-Arcturan Parallel Text

1a. ok-voon ororok sprok .
1b. at-voon bichat dat .

2a. ok-drubel ok-voon anok plok sprok .
2b. at-drubel at-voon pippat rrat dat .

3a. erok sprok izok hihok ghirok .
3b. totat dat arrat vat hilat .

4a. ok-voon anok drok brok jok .
4b. at-voon krat pippat sat lat .

5a. wiwok farok izok stok .
5b. totat jjat quat cat .

6a. lalok sprok izok jok stok .
6b. wat dat krat quat cat .

7a. lalok farok ororok lalok sprok izok enemok .
7b. wat jjat bichat vat dat vat eneat .

8a. lalok brok anok plok nok .
8b. iat lat pippat rrat nnat .

9a. wiwok nok izok kantok ok-yurp .
9b. totat nnatquat oloat at-yurp .

10a. lalok mok nok yorok ghirok clok .
10b. wat nnat gat mat bat hilat .

11a. lalok nok crrrok hihok yorok zanzanok .
11b. wat nnat arrat mat zanzanat .

12a. lalok rarok nok izok hihok mok .
12b. wat nnat forat arrat vat gat .

Translation challenge: farok crrrok hihok yorok clok kantok ok-yurp
(from Knight (1997): Automating Knowledge Acquisition for Machine Translation)
Lexical Translation

В этом смысле подобные действия частично дискредитируют систему американской демократии

<table>
<thead>
<tr>
<th>IBM Model 1</th>
<th>lexical translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Model 2</td>
<td>adds absolute reordering model</td>
</tr>
<tr>
<td>IBM Model 3</td>
<td>adds fertility model</td>
</tr>
<tr>
<td>IBM Model 4</td>
<td>relative reordering model</td>
</tr>
<tr>
<td>IBM Model 5</td>
<td>fixes deficiency</td>
</tr>
</tbody>
</table>
Phrase-Based System Overview

Sentence-aligned corpus → Word alignments → Phrase table (translation model)

- Morgen → Tomorrow
- fliege → I will fly
- ich → to the conference
- nach Kanada → in Canada

Word alignments:
- cat ||| chat ||| 0.9
- the cat ||| le chat ||| 0.8
- dog ||| chien ||| 0.8
- house ||| maison ||| 0.6
- my house ||| ma maison ||| 0.9
- language ||| langue ||| 0.9
- …
В этом смысле подобные действия частично дискредитируют систему американской демократии.
Noisy Channel Model: Phrase-Based MT

Translation Model $P(f|e)$

- Source phrase
- Target phrase
- Translation features

Reranking Model

Language Model $P(e)$

- Feature weights

$\text{argmax}_e P(f|e)P(e)$
Estimate Translation Probabilities

If we have alignments: Maximum Likelihood Estimation

\[ \hat{p}_{\text{MLE}}(e \mid \text{Haus}) = \begin{cases} 
0.8 & \text{if } e = \text{house}, \\
0.16 & \text{if } e = \text{building}, \\
0.02 & \text{if } e = \text{home}, \\
0.015 & \text{if } e = \text{household}, \\
0.005 & \text{if } e = \text{shell}. 
\end{cases} \]
If we have translation probabilities:

<table>
<thead>
<tr>
<th>das</th>
<th>Haus</th>
<th>ist</th>
<th>klein</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>t(e</td>
<td>f)</td>
<td>e</td>
</tr>
<tr>
<td>the</td>
<td>0.7</td>
<td>house</td>
<td>0.8</td>
</tr>
<tr>
<td>that</td>
<td>0.15</td>
<td>building</td>
<td>0.16</td>
</tr>
<tr>
<td>which</td>
<td>0.075</td>
<td>home</td>
<td>0.02</td>
</tr>
<tr>
<td>who</td>
<td>0.05</td>
<td>household</td>
<td>0.015</td>
</tr>
<tr>
<td>this</td>
<td>0.025</td>
<td>shell</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We can estimate Viterbi alignment

\[ a^* = \arg \max_{a \in [0,1,\ldots,n]^m} p(a \mid e, f) \]
Finding the Viterbi Alignment

\[ a^* = \arg \max_{a \in [0, 1, \ldots, n]^m} p(a | e, f) \]

\[ = \arg \max_{a \in [0, 1, \ldots, n]^m} \frac{p(e, a | f)}{\sum_{a'} p(e, a' | f)} \]

\[ = \arg \max_{a \in [0, 1, \ldots, n]^m} p(e, a | f) \]

In model 1:

\[ a_i^* = \arg \max_{a_i = 0} \frac{n}{1 + n} p(e_i | f_{a_i}) \]

\[ = \arg \max_{a_i = 0} p(e_i | f_{a_i}) \]
Finding the Viterbi Alignment

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>das</td>
<td>Haus</td>
<td>ist</td>
<td>klein</td>
</tr>
</tbody>
</table>

The home is little
Finding the Viterbi Alignment

0 NULL
1 das
2 Haus
3 ist
4 klein

1 the
2 home
3 is
4 little
Finding the Viterbi Alignment

0
NULL

1
das

2
Haus

3
ist

4
klein

the

1

2

home

3

is

4

little
Finding the Viterbi Alignment

NULL  | 1    | 2   | 3 | 4
---    | ---  | --- | --- | ---
null   | das  | Haus | ist | klein

the    | home | is | little
1 2 3 4
Finding the Viterbi Alignment

NULL 1 das 2 Haus 3 ist 4 klein

the 1 home 2 is 3 little 4
Finding the Viterbi Alignment

null 1 das 2 Haus 3 ist 4 klein

the 1 home 2 is 3 little 4
Finding the Viterbi Alignment

0
NULL

1
das

2
Haus

3
ist

4
klein

1
the

2
home

3
is

4
little
Finding the Viterbi Alignment

NULL  1  2  3  4
das  Haus  ist  klein

the  home  is  little
Finding the Viterbi Alignment

null → das → Haus → ist → klein

the → home → is → little
Finding the Viterbi Alignment

0  1  2  3  4
NULL das Haus ist klein
the home is little
Finding the Viterbi Alignment

0       1       2       3       4
NULL    das     Haus    ist    klein

1       2       3       4
the     home    is      little
Finding the Viterbi Alignment

0
NULL

1
das

the

2
Haus

home

3
ist

is

4
klein

little
Finding the Viterbi Alignment

0
NULL

1
das

2
Haus

3
ist

4
tklein

1
the

2
home

3
is

4
little
Finding the Viterbi Alignment

0  1    2  3  4
    NULL   das  Haus   ist  klein
       /     /     /     /     /
      the  home  is  little
Finding the Viterbi Alignment

null
das  Haus  ist  klein
the  home  is  little
Finding the Viterbi Alignment

null → das → Haus → ist → klein

the → home → is → little
Finding the Viterbi Alignment

0: NULL
1: das
2: Haus
3: ist
4: klein

the

home

is

little
Finding the Viterbi Alignment

0
NULL
1
das
2
Haus
3
ist
4
klein

the
1
home
2
is
3
little
4
Finding the Viterbi Alignment

0
NULL

1
das
the

2
Haus
home

3
ist
is

4
t klein
little
Finding the Viterbi Alignment

0
NULL

1
das

2
Haus

3
ist

4
klein

1
the

2
home

3
is

4
little
Finding the Viterbi Alignment

NULL  1  2  3  4
das  Haus  ist  klein

the  home  is  little
Finding the Viterbi Alignment

NULL  1  2  3  4

0  das  Haus  ist  klein

1  the  home  is  little
Finding the Viterbi Alignment

NULL

das

Haus

ist

klein

the

home

is

little
Finding the Viterbi Alignment

0  1  2  3  4
NULL das Haus ist klein
  the home is little
Finding the Viterbi Alignment

0 1 2 3 4
NULL das Haus ist klein

the home is little
Finding the Viterbi Alignment

0
NULL

1
das

2
Haus

3
ist

4
klein

1
the

2
home

3
is

4
little
Learning Lexical Translation Models

We would like to estimate the lexical translation probabilities $t(e|f)$ from a parallel corpus but we do not have the alignments.

- **Chicken and egg problem**
  - if we had the alignments, we could estimate the parameters of our generative model (MLE)
  - if we had the parameters, we could estimate the alignments

|     | $t(e|f)$ |
|-----|----------|
| klein | 0.4      |
| small | 0.4      |
| little| 0.1      |
| minor | 0.06     |
| petty | 0.04     |
EM Algorithm

- Incomplete data
  - if we had complete data, we could estimate the model
  - if we had the model, we could fill in the gaps in the data

- Expectation Maximization (EM) in a nutshell

  1. initialize model parameters (e.g. uniform, random)
  2. assign probabilities to the missing data
  3. estimate model parameters from complete data
  4. iterate steps 2–3 until convergence
initialize model parameters, e.g. uniform:

<table>
<thead>
<tr>
<th>e</th>
<th>f</th>
<th>initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>das</td>
<td>0.25</td>
</tr>
<tr>
<td>book</td>
<td>das</td>
<td>0.25</td>
</tr>
<tr>
<td>house</td>
<td>das</td>
<td>0.25</td>
</tr>
<tr>
<td>the</td>
<td>buch</td>
<td>0.25</td>
</tr>
<tr>
<td>book</td>
<td>buch</td>
<td>0.25</td>
</tr>
<tr>
<td>a</td>
<td>buch</td>
<td>0.25</td>
</tr>
<tr>
<td>book</td>
<td>ein</td>
<td>0.25</td>
</tr>
<tr>
<td>a</td>
<td>ein</td>
<td>0.25</td>
</tr>
<tr>
<td>the</td>
<td>haus</td>
<td>0.25</td>
</tr>
<tr>
<td>house</td>
<td>haus</td>
<td>0.25</td>
</tr>
</tbody>
</table>
EM for Model 1

- initialize model parameters, e.g. uniform:
- repeat until convergence:
  - compute “expected” alignments

\[ p(a|e, f) = \frac{p(e, a|f)}{p(e|f)} \]

\[
= \frac{\frac{\epsilon}{(l_f+1)l_e} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)l_e} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)}
\]

\[
= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}
\]

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>das</td>
<td>0.25</td>
</tr>
<tr>
<td>book</td>
<td>das</td>
<td>0.25</td>
</tr>
<tr>
<td>house</td>
<td>das</td>
<td>0.25</td>
</tr>
<tr>
<td>the</td>
<td>buch</td>
<td>0.25</td>
</tr>
</tbody>
</table>

see simplification trick in the previous lecture
EM for Model 1

- initialize model parameters, e.g. uniform:
- repeat until convergence:
  - compute “expected” alignments
- keep track of the expected number of times \( f \) translates into \( e \) throughout the whole corpus

\[
c(e|f; e, f) = \sum_a p(a|e, f) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})
\]

Initialize \( c(e|f) = 0 \) for all \( e, f \) in vocab
for every sentence pair \( e, f \):
  for every alignment \( a \) do
    for \( j = 1 \ldots \text{len}(e) \) do
      \( c(e_j|f_{a(j)}) += p(a|e, f) \)
EM for Model 1

- initialize model parameters, e.g. uniform:
- repeat until convergence:
  - compute “expected” alignments
  - keep track of the expected number of times $f$ translates into $e$ throughout the whole corpus

\[
p(a|e, f)\]

\[
c(e|f; e, f) = \sum_a p(a|e, f) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_a(j))\]

- apply MLE to estimate new model parameters
EM for Model 1

- initialize model parameters, e.g. uniform:
- repeat until convergence:

**E-step**
- compute “expected” alignments

\[ p(a|e, f) \]

**M-step**
- keep track of the expected number of times \( f \) translates into \( e \) throughout the whole corpus

\[
c(e|f; e, f) = \sum_{a} p(a|e, f) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})
\]

- apply MLE to estimate new model parameters

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>initial</th>
<th>1st it.</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>das</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>book</td>
<td>das</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>house</td>
<td>das</td>
<td>0.25</td>
<td>1.0</td>
</tr>
<tr>
<td>the</td>
<td>buch</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>
IBM Model 1 and EM

<table>
<thead>
<tr>
<th></th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(\text{the}</td>
<td>\text{la})$</td>
</tr>
<tr>
<td>$p(\text{the}</td>
<td>\text{maison})$</td>
</tr>
<tr>
<td>$p(\text{house}</td>
<td>\text{la})$</td>
</tr>
<tr>
<td>$p(\text{house}</td>
<td>\text{maison})$</td>
</tr>
</tbody>
</table>
IBM Model 1 and EM

t-table

Probabilities

\[
p(\text{the} | \text{la}) = 0.7 \quad p(\text{house} | \text{la}) = 0.05
\]
\[
p(\text{the} | \text{maison}) = 0.1 \quad p(\text{house} | \text{maison}) = 0.8
\]

Alignments

la \rightarrow \text{the} \quad \text{maison} \rightarrow \text{house}

la \rightarrow \text{the} \quad \text{maison} \rightarrow \text{house}

la \rightarrow \text{the} \quad \text{maison} \rightarrow \text{house}

la \rightarrow \text{the} \quad \text{maison} \rightarrow \text{house}
IBM Model 1 and EM

t-table

Probabilities

\[ p(\text{the}|\text{la}) = 0.7 \]  \[ p(\text{house}|\text{la}) = 0.05 \]
\[ p(\text{the}|\text{maison}) = 0.1 \]  \[ p(\text{house}|\text{maison}) = 0.8 \]

Alignments

\[ p(\text{e}, \text{a}|\text{f}) = 0.56 \]  \[ p(\text{e}, \text{a}|\text{f}) = 0.035 \]  \[ p(\text{e}, \text{a}|\text{f}) = 0.08 \]  \[ p(\text{e}, \text{a}|\text{f}) = 0.005 \]
**IBM Model 1 and EM**

<table>
<thead>
<tr>
<th>t-table</th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p(\text{the}</td>
</tr>
<tr>
<td></td>
<td>( p(\text{the}</td>
</tr>
<tr>
<td></td>
<td>( p(\text{house}</td>
</tr>
<tr>
<td></td>
<td>( p(\text{house}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>la → the</td>
</tr>
<tr>
<td>maison → house</td>
</tr>
<tr>
<td>la → house</td>
</tr>
<tr>
<td>la → maison → house</td>
</tr>
<tr>
<td>la → the</td>
</tr>
<tr>
<td>maison → house</td>
</tr>
<tr>
<td>la → house</td>
</tr>
<tr>
<td>la → maison → house</td>
</tr>
</tbody>
</table>

\[
p(e, a|f) = 0.56 \quad p(e, a|f) = 0.035 \quad p(e, a|f) = 0.08 \quad p(e, a|f) = 0.005
\]

Applying the chain rule:

\[
p(a|e,f) = \frac{p(e,a|f)}{p(e|f)}
\]

\[
p(e,a) = p(e)p(a|e)
\]
IBM Model 1 and EM: Expectation Step

**t-table**

Probabilities

\[
p(\text{the}|\text{la}) = 0.7 \quad p(\text{house}|\text{la}) = 0.05
\]
\[
p(\text{the}|\text{maison}) = 0.1 \quad p(\text{house}|\text{maison}) = 0.8
\]

Alignments

\[
\begin{align*}
\text{la} & \rightarrow \text{the} \\
\text{maison} & \rightarrow \text{house}
\end{align*}
\]
\[
\begin{align*}
p(\text{e}, \text{a}|\text{f}) &= 0.56 \\
p(\text{e}, \text{a}|\text{f}) &= 0.035 \\
p(\text{e}, \text{a}|\text{f}) &= 0.08 \\
p(\text{e}, \text{a}|\text{f}) &= 0.005
\end{align*}
\]

**E-step**

\[
p(a|\text{e}, \text{f}) = 0.824 \\
p(a|\text{e}, \text{f}) = 0.052 \\
p(a|\text{e}, \text{f}) = 0.118 \\
p(a|\text{e}, \text{f}) = 0.007
\]

\[
p(a|\text{e}, \text{f}) = \frac{p(\text{e}, \text{a}|\text{f})}{p(\text{e}|\text{f})}
\]
IBM Model 1 and EM: Maximization Step

**t-table**

Probabilities

\[
\begin{align*}
    p(\text{the}|\text{la}) &= 0.7 \\
    p(\text{the}|\text{maison}) &= 0.1 \\
    p(\text{house}|\text{la}) &= 0.05 \\
    p(\text{house}|\text{maison}) &= 0.8
\end{align*}
\]

Alignments

\[
\begin{align*}
    \text{la} &\rightarrow \text{the} & \text{la} &\rightarrow \text{the} & \text{la} &\rightarrow \text{the} & \text{la} &\rightarrow \text{the} \\
    \text{maison} &\rightarrow \text{house} & \text{maison} &\rightarrow \text{house} & \text{maison} &\rightarrow \text{house} & \text{maison} &\rightarrow \text{house}
\end{align*}
\]

\[
\begin{align*}
    p(e, a|f) &= 0.56 \\
    p(e, a|f) &= 0.035 \\
    p(e, a|f) &= 0.08 \\
    p(e, a|f) &= 0.005
\end{align*}
\]

**E-step**

\[
\begin{align*}
    p(a|e, f) &= 0.824 \\
    p(a|e, f) &= 0.052 \\
    p(a|e, f) &= 0.118 \\
    p(a|e, f) &= 0.007
\end{align*}
\]

**M-step**

Counts

\[
\begin{align*}
    c(\text{the}|\text{la}) &= 0.824 + 0.052 \\
    c(\text{house}|\text{la}) &= 0.052 + 0.007 \\
    c(\text{the}|\text{maison}) &= 0.118 + 0.007 \\
    c(\text{house}|\text{maison}) &= 0.824 + 0.118
\end{align*}
\]
IBM Model 1 and EM: Maximization Step

**t-table**

**Probabilities**

\[
p(\text{the}|\text{la}) = 0.7 \\
p(\text{the}|\text{maison}) = 0.1 \\
p(\text{house}|\text{la}) = 0.05 \\
p(\text{house}|\text{maison}) = 0.8
\]

**E-step**

**Alignments**

\[
p(a|e, f) = 0.824 \\
p(a|e, f) = 0.052 \\
p(a|e, f) = 0.118 \\
p(a|e, f) = 0.007
\]

**M-step**

**Counts**

\[
c(\text{the}|\text{la}) = 0.824 + 0.052 \\
c(\text{the}|\text{maison}) = 0.118 + 0.007 \\
c(\text{house}|\text{la}) = 0.052 + 0.007 \\
c(\text{house}|\text{maison}) = 0.824 + 0.118
\]

**Update t-table:**

\[
p(\text{the}|\text{la}) = \frac{c(\text{the}|\text{la})}{c(\text{la})}
\]
IBM Model 1 and EM: Pseudocode

- implementation of IBM Model 1 after applying the simplification trick (in previous lecture)
IBM Model 2

\[ p(e, a|f) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \text{align.prob}(a(j)|j, l_e, l_f) \]

how many alignment parameters?
EM for Model 2

- Now we have two sets of parameters:
  - initialize t-table parameters uniformly or carry over from trained Model 1
  - initialize alignment probabilities uniformly

```
for each l_e do:
  for each l_f do:
    for i = 0..l_f do
      for j = 1..l_e do
        align_prob(i|j, l_e, l_f) = 1/(l_f+1)
```
EM for Model 2

- initialize model parameters:
- repeat until convergence:
  - compute “expected” alignments

\[ p(a|e, f) = p(e, a|f)/p(e|f) \]

\[
p(e, a|f) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \text{align-prob}(a(j)|j, l_e, l_f) \]

\[
p(e|f) = \epsilon \prod_{i=0}^{l_e} \sum_{j=1}^{l_f} t(e_j|f_i) \text{align-prob}(i|j, l_e, l_f) \]
EM for Model 2

- initialize model parameters, e.g. uniform:

- repeat until convergence:
  - compute “expected” alignments

\[
p(a|e, f)
\]

- keep track of the expected number of times \( f \) translates into \( e \) throughout the whole corpus

\[
c(e|f; e, f) = \sum_a p(a|e, f) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})
\]

Initialize \( c(e|f) = 0 \) for all \( e, f \) in vocab
for every sentence pair \( e, f \):
  for every alignment \( a \) do
    for \( j = 1..\text{len}(e) \) do
      \( c(e_j|f_{a(j)}) += p(a|e, f) \)
EM for Model 2

- initialize model parameters, e.g. uniform:
- repeat until convergence:
  - compute “expected” alignments
  - keep track of the expected number of times $f$ translates into $e$ throughout the whole corpus
  \[
  c(e|f; e, f) = \sum_a p(a|e, f) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})
  \]
- apply MLE to estimate new model parameters

<table>
<thead>
<tr>
<th>$e$</th>
<th>$f$</th>
<th>initial</th>
<th>1st it.</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>das</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>book</td>
<td>das</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>house</td>
<td>das</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>the</td>
<td>buch</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

same as in Model 1 for t-table parameters
EM for Model 2

- keep track of the expected number of times $f$ translates into $e$ throughout the whole corpus

\[ c(e|f; e, f) = \sum_a p(a|e, f) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)}) \]

- apply MLE to estimate new model parameters

\[ t(e|f; e, f) = \frac{\sum_{(e,f)} c(e|f; e, f)}{\sum_e \sum_{(e,f)} c(e|f; e, f)} \]

- same as in Model 1 for t-table parameters

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>initial</th>
<th>1st it.</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>das</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>book</td>
<td>das</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>house</td>
<td>das</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>the</td>
<td>buch</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

\[ c(i|j, l_e, l_f; e, f) = \frac{t(e_j|f_i) \text{align-prob}(a(j)|j, l_e, l_f)}{\sum_{i'=0}^{l_f} t(e_{j'}|f_{i'}) \text{align-prob}(a(i')|j, l_e, l_f)} \]
IBM Model 2 and EM

Input: set of sentence pairs \((e, f)\)

Output: probability distributions \(t\) (lexical translation)

and \(a\) (alignment)

1: carry over \(t(e|f)\) from Model 1

2: initialize \(a(i, j, l_e, l_f) = 1/(l_f+1)\) for all \(i, j, l_e, l_f\)

3: while not converged do

4: // initialize

5: count\((e|f)\) = 0 for all \(e, f\)

6: total\((f)\) = 0 for all \(f\)

7: count\(_a\)(\(i, j, l_e, l_f\)) = 0 for all \(i, j, l_e, l_f\)

8: total\(_a\)(\(j, l_e, l_f\)) = 0 for all \(j, l_e, l_f\)

9: for all sentence pairs \((e, f)\) do

10: \(l_e = \text{length}(e)\), \(l_f = \text{length}(f)\)

11: // compute normalization

12: for \(j = 1 \ldots l_e\) do // all word positions in \(e\)

13: s-total\(_e\)(\(e_j\)) = 0

14: for \(i = 0 \ldots l_f\) do // all word positions in \(f\)

15: s-total\(_f\)(\(e_j|f_i\)) += \(t(e_j|f_i) \times a(i, j, l_e, l_f)\)

16: end for

17: end for

18: // collect counts

19: for \(j = 1 \ldots l_e\) do // all word positions in \(e\)

20: for \(i = 0 \ldots l_f\) do // all word positions in \(f\)

21: \(c = t(e_j|f_i) \times a(i, j, l_e, l_f) / \text{s-total}(e_j)\)

22: count\(_e\)(\(e_j|f_i\)) += \(c\)

23: total\(_f\)(\(f_i\)) += \(c\)

24: count\(_a\)(\(i, j, l_e, l_f\)) += \(c\)

25: total\(_a\)(\(j, l_e, l_f\)) += \(c\)

26: end for

27: end for

28: end for

29: // estimate probabilities

30: \(t(e|f) = 0\) for all \(e, f\)

31: \(a(i, j, l_e, l_f) = 0\) for all \(i, j, l_e, l_f\)

32: for all \(e, f\) do

33: \(t(e|f) = \text{count}(e|f) / \text{total}(f)\)

34: end for

35: for all \(i, j, l_e, l_f\) do

36: \(a(i, j, l_e, l_f) = \text{count}_a(i, j, l_e, l_f) / \text{total}_a(j, l_e, l_f)\)

37: end for

38: end while

- implementation of IBM Model 2 after applying the simplification trick (in previous lecture)
A: Thank you, I shall do so gladly.

E:  

1 2 3 4 5 6 7 8 9

A:  

1 3 7 6 8 8 8 8 9

F: Gracias, lo haré de muy buen grado.

Model Parameters

Emissions: $P( F_1 = \text{Gracias} | E_{A1} = \text{Thank})$

Transitions: $P( A_2 = 3)$
Localization effect in aligning the words
Thank you, I shall do so gladly.

Gracias, lo haré de muy buen grado.

**Model Parameters**

*Emissions:* $P(F_1 = \text{Gracias} | E_{A1} = \text{Thank})$

*Transitions:* $P(A_2 = 3 | A_1 = 1)$
The HMM Model

- Model 2 preferred global monotonicity
- We want local monotonicity:
  - Most jumps are small
- HMM model (Vogel 96)

\[
P(f, a | e) = \prod_j P(a_j | a_{j-1}) P(f_j | e_i)
\]

- Re-estimate using the forward-backward algorithm
- Handling nulls requires some care

| f        | t(f | e) |
|----------|--------|
| nationale| 0.469  |
| national | 0.418  |
| nationaux| 0.054  |
| nationales| 0.029 |
Given a sentence pair, which words correspond to each other?

<table>
<thead>
<tr>
<th>Michael</th>
<th>geht</th>
<th>davon</th>
<th>aus</th>
<th>dass</th>
<th>er</th>
<th>im</th>
<th>haus</th>
<th>bleibt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>assumes</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>that</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>he</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>will</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>stay</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>in</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>the house</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Is the English word **does** aligned to the German **wohnt** (verb) or **nicht** (negation) or neither?
How do the idioms *kicked the bucket* and *biss ins grass* match up? Outside this exceptional context, *bucket* is never a good translation for *grass*.
IBM Models create a many-to-one mapping
- words are aligned using an alignment function
- a function may return the same value for different input (one-to-many mapping)
- a function cannot return multiple values for one input (no many-to-one mapping)

Real word alignments have many-to-many mappings
Symmetrization

**english to spanish**

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>daba</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**spanish to english**

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>daba</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**intersection**

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>daba</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Growing Heuristics

- Add alignment points from union based on heuristics
- Popular method: grow-diag-final-and
Alignment Error Rate

Possible links $P$

Sure links $S$

Precision($A, P$) = $\frac{|P \cap A|}{|A|}$

Recall($A, S$) = $\frac{|S \cap A|}{|S|}$

\[ \text{AER}(A, P, S) = 1 - \frac{|S \cap A| + |P \cap A|}{|S'| + |A|} \]
## AER for HMMs

<table>
<thead>
<tr>
<th>Model</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1 INT</td>
<td>19.5</td>
</tr>
<tr>
<td>HMM E→F</td>
<td>11.4</td>
</tr>
<tr>
<td>HMM F→E</td>
<td>10.8</td>
</tr>
<tr>
<td>HMM AND</td>
<td>7.1</td>
</tr>
<tr>
<td>HMM INT</td>
<td>4.7</td>
</tr>
<tr>
<td>GIZA M4 AND</td>
<td>6.9</td>
</tr>
</tbody>
</table>
Phrase-Based MT
Phrase-Based Translation Overview

**Input:**    lo haré | rápidamente |.

**Translations:**    I'll do it | quickly |.
                     quickly | I'll do it |.

**Objective:**    \[
\arg \max_e \left[ P(f|e) \cdot P(e) \right] \\
\arg \max_e \left[ \prod_{\langle \tilde{e}, f \rangle} P(f|\tilde{e}) \cdot \prod_{i=1}^{|e|} P(e_i|e_{i-1}, e_{i-2}) \right]
\]

*The decoder...* tries different segmentations, translates phrase by phrase, and considers reorderings.
Phrase Translations Example

- Phrase translations for *den Vorschlag* learned from the Europarl corpus:

| English               | $\phi(\tilde{e}|f)$ | English               | $\phi(\tilde{e}|f)$ |
|-----------------------|----------------------|-----------------------|----------------------|
| the proposal          | 0.6227               | the suggestions       | 0.0114               |
| 's proposal           | 0.1068               | the proposed          | 0.0114               |
| a proposal            | 0.0341               | the motion            | 0.0091               |
| the idea              | 0.0250               | the idea of           | 0.0091               |
| this proposal         | 0.0227               | the proposal ,        | 0.0068               |
| proposal              | 0.0205               | its proposal          | 0.0068               |
| of the proposal       | 0.0159               | it                    | 0.0068               |
| the proposals         | 0.0159               | ...                   | ...                  |

- lexical variation (*proposal* vs *suggestions*)
- morphological variation (*proposal* vs *proposals*)
- included function words (*the, a, ...*)
- noise (*it*)
Linguistic Phrases?

- Model is not limited to linguistic phrases (noun phrases, verb phrases, prepositional phrases, ...)
- Example non-linguistic phrase pair
  
  spass am → fun with the

- Prior noun often helps with translation of preposition
- Experiments show that limitation to linguistic phrases hurts quality
Decoder design is important: [Koehn et al. 03]
extract phrase pair consistent with word alignment:

assumes that / geht davon aus, dass
Consistent

All words of the phrase pair have to align to each other.
Phrase Pair Extraction

Smallest phrase pairs:
- michael — michael
- assumes — geht davon aus / geht davon aus ,
- that — dass / , dass
- he — er
- will stay — bleibt
- in the — im
- house — haus

unaligned words (here: German comma) lead to multiple translations
michael assumes — michael geht davon aus / michael geht davon aus,
assumes that — geht davon aus, dass ; assumes that he — geht davon aus, dass er
that he — dass er /, dass er ; in the house — im haus
michael assumes that — michael geht davon aus, dass
michael assumes that he — michael geht davon aus, dass er
michael assumes that he will stay in the house — michael geht davon aus, dass er im haus bleibt
assumes that he will stay in the house — geht davon aus, dass er im haus bleibt
that he will stay in the house — dass er im haus bleibt ; dass er im haus bleibt,
he will stay in the house — er im haus bleibt ; will stay in the house — im haus bleibt
The Pharaoh “Model”

[Koehn et al, 2003]

\[
P(e|g) = P(\{\bar{g}_i\}|g) \prod_i \phi(\bar{e}_i|\bar{g}_i) d(a_i - b_{i-1})
\]

Segmentation  Translation  Distortion
Distance-Based Reordering

![Diagram showing reordering process with phrases in foreign and English languages.]

<table>
<thead>
<tr>
<th>phrase</th>
<th>translates</th>
<th>movement</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-3</td>
<td>start at beginning</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>skip over 4-5</td>
<td>+2</td>
</tr>
<tr>
<td>3</td>
<td>4-5</td>
<td>move back over 4-6</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>skip over 6</td>
<td>+1</td>
</tr>
</tbody>
</table>

Scoring function: $d(x) = \alpha^{|x|} \text{ — exponential with distance}$
The Pharaoh “Model”

\[ P(f|e) = P(\{\bar{e}_i\}|e) \prod_i \phi(f_i|\bar{e}_i) d(a_i - b_{i-1}) \]

\[ \frac{1}{K} \]

\[ \frac{\text{count}(f_i, \bar{e}_i)}{\text{count}(\bar{e}_i)} \]

\[ \alpha |a_i - b_{i-1}| \]

Where do we get these counts?
Phrase Weights

How the MT community estimates $P(\bar{f}|\bar{e})$

**Parallel training sentences**

Gracias, lo haré de muy buen grado.
Thank you, I shall do so gladly.

**provide phrase pair counts.**

lo haré $\iff$ I shall do so
44 times in the corpus

**All phrase pairs are counted,**

Gracias, lo haré de muy buen grado.
Thank you, I shall do so gladly.

**and counts are normalized.**

$$P(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\text{count}(\bar{e})}$$
<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
<td></td>
</tr>
<tr>
<td>did not</td>
<td>a slap</td>
<td>by</td>
<td></td>
<td>green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>slap</td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---------- slap ---------- the witch
Many translation options to choose from

- in Europarl phrase table: 2727 matching phrase pairs for this sentence
- by pruning to the top 20 per phrase, 202 translation options remain
The machine translation decoder does not know the right answer
- picking the right translation options
- arranging them in the right order

→ Search problem solved by heuristic beam search
Decoding: Start with Initial Hypothesis

initial hypothesis: no input words covered, no output produced
Decoding: Hypothesis Expansion

pick any translation option, create new hypothesis
Decoding: Hypothesis Expansion

create hypotheses for all other translation options
Decoding: Hypothesis Expansion

also create hypotheses from created partial hypothesis
Decoding: Find Best Path

backtrack from highest scoring complete hypothesis
Computational Complexity

- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Reduction of search space:
  - recombination (risk-free)
  - pruning (risky)
Recombination

- Two hypothesis paths lead to two matching hypotheses
  - same foreign words translated
  - same English words in the output

- Worse hypothesis is dropped
Recombination

- Two hypothesis paths lead to hypotheses indistinguishable in subsequent search
  - same foreign words translated
  - same last two English words in output (assuming trigram language model)
  - same last foreign word translated

- Worse hypothesis is dropped
Pruning: Beams + Forward Costs

Problem: easy partial analyses are cheaper
  - Solution 1: use beams per foreign subset
  - Solution 2: estimate forward costs (A*-like)

Maria no dio una bofetada a la bruja verde

- **e: Mary did not**
  - **f:** *-------*
  - **p:** 0.154

- **e: the**
  - **f:** -----**--*
  - **p:** 0.354

**better**

**partial**

**translation**

**covers**

**easier part**

**--> lower cost**
Parameter Tuning
Learning weights has been tried, several times:
  - [Marcu and Wong, 02]
  - [DeNero et al, 06]
  - … and others

Seems not to work well, for a variety of partially understood reasons

Main issue: big chunks get all the weight, obvious priors don’t help
  - Though, [DeNero et al 08]
Phrases do help
  - But they don’t need to be long
  - Why should this be?

\[ \text{BLEU} \]

- max7
- max5
- max4
- max3
- max2
\[ \phi(\overline{f_i} | \overline{e_i}) = \frac{\text{count}(\overline{f_i}, \overline{e_i})}{\text{count}(\overline{e_i})} p_w(\overline{f_i} | \overline{e_i}) \]

\[ p_w(\overline{f} | \overline{e}, a) = p_w(f_1 f_2 f_3 | e_1 e_2 e_3, a) = w(f_1 | e_1) \times \frac{1}{2}(w(f_2 | e_2) + w(f_2 | e_3)) \times w(f_3 | \text{NULL}) \]
Features encapsulate lots of information
- Basic MT systems have around 6 features
- $P(e|f)$, $P(f|e)$, lexical weighting, language model

How to tune feature weights?

Idea 1: Use your favorite classifier
Why Tuning is Hard

- Problem 1: There are latent variables
  - Alignments and segmentations

\[ \begin{array}{c}
x: \text{le parlement, adopte, la, résolution, législative} \\
h: \\
y: \text{parliament, has, adopted, the, resolution} \end{array} \]
Why Tuning is Hard

- Problem 3: Computational constraints
  - Discriminative training involves repeated decoding
  - Very slow! So people tune on sets much smaller than those used to build phrase tables
Minimum Error Rate Training

- Standard method: minimize BLEU directly (Och 03)
  - MERT is a discontinuous objective
  - Only works for max ~10 features, but works very well then
  - Here: k-best lists, but forest methods exist (Machery et al 08)
  - Recently, lots of alternatives being explored for more features
MERT

Model Score

BLEU Score