

# CSE 842

## Natural Language Processing

### Lecture 22: Machine Translation (2)

## Topics to be covered

- Alignment in MT:
  - IBM Model 1 and Model 2
- Phrase-based models
- Decoding

## The Noisy Channel Model

- Goal: translation system from French to English

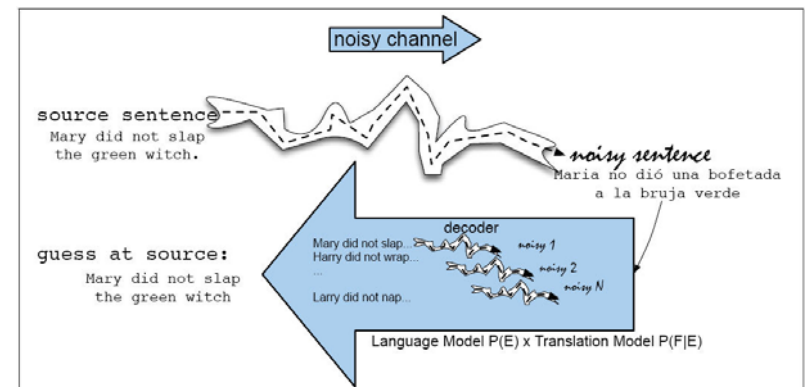
$$P(\mathbf{e} | \mathbf{f}) = \frac{P(\mathbf{e}, \mathbf{f})}{P(\mathbf{f})} = \frac{P(\mathbf{e})P(\mathbf{f} | \mathbf{e})}{\sum_{\mathbf{e}} P(\mathbf{e})P(\mathbf{f} | \mathbf{e})}$$

and

$$\arg \max_{\mathbf{e}} P(\mathbf{e} | \mathbf{f}) = \arg \max_{\mathbf{e}} P(\mathbf{e})P(\mathbf{f} | \mathbf{e})$$

- A Noisy Channel Model has two components:
  - $P(\mathbf{e})$  the language model: could be a trigram model, estimated from any data (parallel corpus not needed)
  - $P(\mathbf{f} | \mathbf{e})$  the translation model: trained from a parallel corpus of French/English pairs

## The Noisy Channel Model



# IBM Model 1: Alignment

- How do we model  $P(\mathbf{f}|\mathbf{e})$
- English sentence  $\mathbf{e}$  has  $l$  words:  $e_1 \dots e_l$ ,  
French sentence  $\mathbf{f}$  has  $m$  words  $f_1 \dots f_m$ .
- An alignment  $\mathbf{a}$  identifies which English word each French word originated from
- Formally, an alignment  $\mathbf{a}$  is  $\{a_1, \dots, a_m\}$ , where each  $a_j \in \{0 \dots l\}$
- There are  $(l+1)^m$  possible alignments.
  - Each French word is originated from exactly one English word (including NULL).

# IBM Model 1: Alignment

- e.g.,  $l = 6, m = 7$   
 $\mathbf{e} =$  And the program has been implemented  
 $\mathbf{f} =$  Le programme a ete mis en application
- One alignment is  
 $\{2, 3, 4, 5, 6, 6, 6\}$

Another (bad) alignment is  
 $\{1, 1, 1, 1, 1, 1, 1\}$

# IBM Model 1: Alignment

- In IBM model 1, all alignment  $\mathbf{a}$  is equally likely

$$P(\mathbf{a} | \mathbf{e}) = C \times \frac{1}{(l+1)^m}, \mathbf{a} \in A$$

where  $C = \text{prob}(\text{length}(\mathbf{f}) = m)$  is a constant

- This is a **major** simplifying assumption, but it gets things start.

# IBM Model 1: Translation Probabilities

- Next step: come up with an estimate for  
 $P(\mathbf{f} | \mathbf{a}, \mathbf{e})$
- In model 1, this is:

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

## IBM Model 1: Translation Probabilities

e.g.,  $l = 6, m = 7$

$\mathbf{e}$  = And the program has been implemented

$\mathbf{f}$  = Le programme a ete mis en application

$\mathbf{a} = \{2,3,4,5,6,6,6\}$

$$P(\mathbf{f} | \mathbf{a}, \mathbf{e}) = P(\text{Le} | \text{the}) \times P(\text{programme} | \text{program}) \times \\ P(\text{a} | \text{has}) \times P(\text{ete} | \text{been}) \times \\ P(\text{mis} | \text{implemented}) \times \\ P(\text{en} | \text{implemented}) \times \\ P(\text{application} | \text{implemented})$$

## IBM Model 1: The Generative Process

To generate a French string  $\mathbf{f}$  from an English string  $\mathbf{e}$

- Step 1: Pick the length of  $\mathbf{f}$  (all lengths equally probable,  $C$ )
- Step 2: Pick an alignment  $\mathbf{a}$  with probability  $\frac{1}{(l+1)^m}$

- Step 3: Pick the French words with probability

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

- The final result:

$$P(\mathbf{f}, \mathbf{a} | \mathbf{e}) = P(\mathbf{a} | \mathbf{e}) \times P(\mathbf{f} | \mathbf{a}, \mathbf{e}) = \frac{C}{(l+1)^m} \prod_{j=1}^m P(f_j | e_{a_j})$$

## IBM Model 1:

- We have

$$P(\mathbf{f}, \mathbf{a} | \mathbf{e}) = \frac{C}{(l+1)^m} \prod_{j=1}^m P(f_j | e_{a_j})$$

- And:

$$P(\mathbf{f} | \mathbf{e}) = \sum_{\mathbf{a} \in A} \frac{C}{(l+1)^m} \prod_{j=1}^m P(f_j | e_{a_j})$$

Where  $A$  is the set of all possible alignments

## IBM Model 2:

- Only difference: we now introduce alignment or distortion parameters

$D(i | j, l, m)$  = Probability that  $j$ 'th French word is connected to  $i$ 'th English word, given sentence lengths of  $\mathbf{e}$  and  $\mathbf{f}$  are  $l$  and  $m$  respectively

- Define  $P(\mathbf{a} = \{a_1, \dots, a_m\} | \mathbf{e}, l, m) = \prod_{j=1}^m D(a_j | j, l, m)$

- Gives  $P(\mathbf{f}, \mathbf{a} | \mathbf{e}, l, m) = \prod_{j=1}^m D(a_j | j, l, m) T(f_j | e_{a_j})$

Model 1 is a special case of Model 2, where

$$D(i | j, l, m) = \frac{1}{l+1} \text{ for all } i, j.$$

## An Example

$l = 6, m = 7$

**e** = And the Program has been implemented

**f** = Le programme a ete mis en application

**a** = {2,3,4,5,6,6,6}  $P(\mathbf{a} | \mathbf{e}, l = 6, m = 7) = D(i = 2 | j = 1, l = 6, m = 7) \times$   
 $D(i = 3 | j = 2, l = 6, m = 7) \times$   
 $D(i = 4 | j = 3, l = 6, m = 7) \times$   
 $D(i = 5 | j = 4, l = 6, m = 7) \times$   
 $D(i = 6 | j = 5, l = 6, m = 7) \times$   
 $D(i = 6 | j = 6, l = 6, m = 7) \times$   
 $D(i = 6 | j = 7, l = 6, m = 7) \times$

## An Example

e.g.,  $l = 6, m = 7$

**e** = And the program has been implemented

**f** = Le programme a ete mis en application

**a** = {2,3,4,5,6,6,6}

$P(\mathbf{f} | \mathbf{a}, \mathbf{e}) = T(\text{Le} | \text{the}) \times T(\text{programme} | \text{program}) \times$   
 $T(\text{a} | \text{has}) \times T(\text{ete} | \text{been}) \times$   
 $T(\text{mis} | \text{implemented}) \times$   
 $T(\text{en} | \text{implemented}) \times$   
 $T(\text{application} | \text{implemented})$

## IBM Model 2: The Generative Process

To generate a French string **f** from an English string **e**

- Step 1: Pick the length of **f** (all lengths equally probable,  $C$ )
- Step 2: Pick an alignment **a** = { $a_1, a_2, \dots, a_m$ } with probability

$$\prod_{j=1}^m D(a_j | j, l, m)$$

- Step 3: Pick the French words with probability

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

- The final result:

$$P(\mathbf{f}, \mathbf{a} | \mathbf{e}) = P(\mathbf{a} | \mathbf{e}) \times P(\mathbf{f} | \mathbf{a}, \mathbf{e}) = C \prod_{j=1}^m D(a_j | j, l, m) T(f_j | e_{a_j})$$

## IBM Model 2:

- We have

$$P(\mathbf{f}, \mathbf{a} | \mathbf{e}) = C \prod_{j=1}^m D(a_j | j, l, m) T(f_j | e_{a_j})$$

- And:

$$P(\mathbf{f} | \mathbf{e}) = \sum_{\mathbf{a} \in A} C \prod_{j=1}^m D(a_j | j, l, m) T(f_j | e_{a_j})$$

Where  $A$  is the set of all possible alignments

# A Hidden Variable Problem

- Training data is a set of  $(\mathbf{f}_i, \mathbf{e}_i)$  pairs, the log likelihood of data is:

$$\sum_i \log P(\mathbf{f}_i | \mathbf{e}_i) = \sum_i \log \sum_{\mathbf{a} \in A} P(\mathbf{a} | \mathbf{e}_i) P(\mathbf{f}_i | \mathbf{a}, \mathbf{e}_i)$$

Where A is the set of all possible alignments

- We need to find model parameters (i.e., translation probabilities) to maximize the log likelihood function
- EM can be used for this problem: initialize translation probabilities randomly, and at each iteration choose

$$\Theta^t = \arg \max_{\Theta} \sum_i \sum_{\mathbf{a} \in A} P(\mathbf{a} | \mathbf{e}_i, \mathbf{f}_i, \Theta^{t-1}) \log(\mathbf{f}_i | \mathbf{a}, \mathbf{e}_i, \Theta)$$

where  $\Theta^t$  are the parameter values at the t'th iteration

# The EM Algorithm

- Initialize the model parameters to some arbitrary values  $\theta_i^0$
- Iterate the E-step and the M-step until convergence. During step k
  - Compute the expected values of the hidden data based on the current parameter estimates  $\theta_i^k$  (E-step)
  - Derive  $\theta_i^{k+1}$  as an ML estimate using the values of the hidden data computed in the E-step (M-step)
- EM always converges, but convergence may be to a *local* maximum.

# Simplification of Model 1 and 2

- We have  $\mathbf{f} = \{f_1 \dots f_m\}$ ,  $\mathbf{a} = \{a_1 \dots a_m\}$ , and

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

where

$$P(a_j, f_j | \mathbf{e}, l, m) = D(a_j | j, l, m) T(f_j | e_{a_j})$$

- We can think of the  $m$   $(f_j, a_j)$  pairs as being generated independently
- How to approach this problem?

# A Crucial Step in the EM Algorithm

- We have the following  $(\mathbf{e}, \mathbf{f})$  pair:
  - $\mathbf{e}$  = And the program has been implemented
  - $\mathbf{f}$  = Le programme a ete mis en application
- Given that  $\mathbf{f}$  was generated according to Model 2, what is the probability that  $a_1 = 2$ ?

$$P(a_1 = 2 | \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}: a_1=2} P(\mathbf{a} | \mathbf{f}, \mathbf{e}, l, m)$$

## A Crucial Step in the EM Algorithm

$$P(a_1 = 2 | f, e) = \sum_{a: a_1=2} P(a | f, e, l, m) = \frac{D(a_1 = 2 | j = 1, l = 6, m = 7)T(le | the)}{\sum_{i=0}^l D(a_1 = i | j = 1, l = 6, m = 7)T(le | e_i)}$$

Follows directly because the  $(f_j, a_j)$  pairs are independent

$$P(a_1 = 2 | f, e, l, m) = \frac{P(a_1 = 2, f_1 = Le | f_2 \dots f_m, e, l, m)}{P(f_1 = Le | f_2 \dots f_m, e, l, m)} \quad (1)$$

$$= \frac{P(a_1 = 2, f_1 = Le | e, l, m)}{P(f_1 = Le | e, l, m)} \quad (2)$$

$$= \frac{P(a_1 = 2, f_1 = Le | e, l, m)}{\sum_i P(a_1 = i, f_1 = Le | e, l, m)}$$

## A Crucial Step in the EM Algorithm

### A General Result

$$P(a_j = i | f, e) = \sum_{a: a_j=i} P(a | f, e, l, m) = \frac{D(a_j = i | j, l = 6, m = 7)T(f_j | e_i)}{\sum_{i'=0}^l D(a_j = i' | j, l = 6, m = 7)T(f_j | e_{i'})}$$

## Alignment Probabilities

e.g.,  $l = 6, m = 7$

**e** = And the program has been implemented

**f** = Le programme a ete mis en application

Probability of “mis” being connected to “the”

$$P(a_5 = 2 | f, e) = \frac{D(a_5 = 2 | j = 5, l = 6, m = 7)T(mis | the)}{Z}$$

where

$$Z = D(a_5 = 0 | j = 5, l = 6, m = 7)T(mis | NULL)$$

$$+ D(a_5 = 1 | j = 5, l = 6, m = 7)T(mis | And)$$

$$+ D(a_5 = 2 | j = 5, l = 6, m = 7)T(mis | the)$$

$$+ D(a_5 = 3 | j = 5, l = 6, m = 7)T(mis | program)$$

+ ...

## The EM Algorithm for Model 2

### • Define

- $e[k]$  for  $k = 1 \dots n$  is the  $k$ 'th English sentence
- $f[k]$  for  $k = 1 \dots n$  is the  $k$ 'th French sentence
- $l[k]$  is the length of  $e[k]$
- $m[k]$  is the length of  $f[k]$
- $e[k, i]$  is the  $i$ 'th word in  $e[k]$
- $f[k, j]$  is the  $j$ 'th word in  $f[k]$

### • Current parameters $\theta^{t-1}$ are

$$T(f | e) \text{ for all } f \in F, e \in E$$

$$D(i | j, l, m)$$

### • EM algorithm is used to re-estimated the T and D parameters

## Step 1: Calculate the Alignment Probabilities

Calculate an array of alignment probabilities  
(for  $k = 1 \dots n, j = 1 \dots m[k], i = 0 \dots l[k]$ )

$$a[i, j, k] = P(a_j = i \mid e[k], f[k], \theta^{t-1}) \\ = \frac{D(a_j = i \mid j, l, m)T(f_{kj} \mid e_{ki})}{\sum_{i'=0}^l D(a_j = i' \mid j, l, m)T(f_{kj} \mid e_{ki'})}$$

where

$$e_{ki} = e[k, i], f_{kj} = f[k, j], \text{ and } l = l[k], m = m[k]$$

i.e., the probability of  $f[k, j]$  being aligned to  $e[k, i]$ .

## Step 2: Calculate the Expected Counts

Calculate the translation counts

$$tcount(e, f) = \sum_{\substack{i, j, k: \\ e[k, i]=e, \\ f[k, j]=f}} a[i, j, k]$$

$tcount(e, f)$  is expected number of times that  $e$  is aligned with  $f$  in the corpus

## Step 2: Calculate the Expected Counts

Calculate the alignment counts

$$acount(i, j, l, m) = \sum_{k: l[k]=l, m[k]=m} a[i, j, k]$$

$acount(i, j, l, m)$  is expected number of times that  $e_i$  is aligned to  $f_j$  in English/French sentences of length  $l$  and  $m$  respectively

## Step 3: Re-estimating the Parameters

New translation probabilities are then defined as

$$P(f \mid e) = \frac{tcount(e, f)}{\sum_f tcount(e, f)}$$

New alignment probabilities are defined as

$$P(a_j = i \mid j, l, m) = \frac{acount(i, j, l, m)}{\sum_i acount(i, j, l, m)}$$

this defines the mapping from  $\theta^{t-1}$  to  $\theta^t$ .

# The Special Case of Model 1

- Start with parameters  $\theta^{t-1}$  as

$$T(f|e) \text{ for all } f \in F, e \in E$$

- Calculate **alignment probabilities** under current parameters

$$a[i, j, k] = \frac{T(f_{kj} | e_{ki})}{\sum_{i'=0}^l T(f_{kj} | e_{ki'})}$$

$$(\because D(a_j = i | j, l, m) = \frac{1}{(l+1)^m} \text{ for all } i, j, l, m)$$

- Calculate **expected counts**  $tcount(e, f)$
- Re-estimate  $T(f|e)$  from the expected counts

# A Summary of the EM Procedure

- Start with parameters  $\theta^{t-1}$  as

$$T(f|e) \text{ for all } f \in F, e \in E$$

$$D(i | j, l, m)$$

- Calculate **alignment probabilities** under current parameters

$$a[i, j, k] = \frac{D(a_j = i | j, l, m) T(f_{kj} | e_{ki})}{\sum_{i'=0}^l D(a_j = i' | j, l, m) T(f_{kj} | e_{ki'})}$$

- Calculate **expected counts**  $tcount(e, f)$  and  $acount(i, j, l, m)$  from the alignment probabilities.
- Re-estimate  $T(f|e)$  and  $D(i|j, l, m)$  from the expected counts

# An Example of training Models 1 & 2

Example will use the following translations

$e[1] = \text{the dog}$

$f[1] = \text{le chien}$

$e[2] = \text{the cat}$

$f[2] = \text{le chat}$

$e[3] = \text{the bus}$

$f[3] = \text{l' autobus}$

No use of NULL word as  $e_0$

**Initial (random) parameters:**

$e$	$f$	$T(f e)$
the	le	0.23
the	chien	0.2
the	chat	0.11
the	l'	0.25
the	autobus	0.21
dog	le	0.2
dog	chien	0.16
dog	chat	0.33
dog	l'	0.12
dog	autobus	0.18
cat	le	0.26
cat	chien	0.28
cat	chat	0.19
cat	l'	0.24
cat	autobus	0.03
bus	le	0.22
bus	chien	0.05
bus	chat	0.26
bus	l'	0.19
bus	autobus	0.27

**Alignment probabilities:**

i	j	k	a(i,j,k)
1	1	0	0.526423237959726
2	1	0	0.473576762040274
1	2	0	0.552517995605817
2	2	0	0.447482004394183
1	1	1	0.466532602066533
2	1	1	0.533467397933467
1	2	1	0.356364544422507
2	2	1	0.643635455577493
1	1	2	0.571950438336247
2	1	2	0.428049561663753
1	2	2	0.439081311724508
2	2	2	0.560918688275492

**Expected counts:**

<i>e</i>	<i>f</i>	<i>tcount(e, f)</i>
the	le	0.99295584002626
the	chien	0.552517995605817
the	chat	0.356364544422507
the	l'	0.571950438336247
the	autobus	0.439081311724508
dog	le	0.473576762040274
dog	chien	0.447482004394183
dog	chat	0
dog	l'	0
dog	autobus	0
cat	le	0.533467397933467
cat	chien	0
cat	chat	0.643635455577493
cat	l'	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l'	0.428049561663753
bus	autobus	0.560918688275492

**Old and new parameters:**

<i>e</i>	<i>f</i>	old	new
the	le	0.23	0.34
the	chien	0.2	0.19
the	chat	0.11	0.12
the	l'	0.25	0.2
the	autobus	0.21	0.15
dog	le	0.2	0.51
dog	chien	0.16	0.49
dog	chat	0.33	0
dog	l'	0.12	0
dog	autobus	0.18	0
cat	le	0.26	0.45
cat	chien	0.28	0
cat	chat	0.19	0.55
cat	l'	0.24	0
cat	autobus	0.03	0
bus	le	0.22	0
bus	chien	0.05	0
bus	chat	0.26	0
bus	l'	0.19	0.43
bus	autobus	0.27	0.57

<i>e</i>	<i>f</i>						
the	le	0.23	0.34	0.46	0.56	0.64	0.71
the	chien	0.2	0.19	0.15	0.12	0.09	0.06
the	chat	0.11	0.12	0.1	0.08	0.06	0.04
the	l'	0.25	0.2	0.17	0.15	0.13	0.11
the	autobus	0.21	0.15	0.12	0.1	0.08	0.07
dog	le	0.2	0.51	0.46	0.39	0.33	0.28
dog	chien	0.16	0.49	0.54	0.61	0.67	0.72
dog	chat	0.33	0	0	0	0	0
dog	l'	0.12	0	0	0	0	0
dog	autobus	0.18	0	0	0	0	0
cat	le	0.26	0.45	0.41	0.36	0.3	0.26
cat	chien	0.28	0	0	0	0	0
cat	chat	0.19	0.55	0.59	0.64	0.7	0.74
cat	l'	0.24	0	0	0	0	0
cat	autobus	0.03	0	0	0	0	0
bus	le	0.22	0	0	0	0	0
bus	chien	0.05	0	0	0	0	0
bus	chat	0.26	0	0	0	0	0
bus	l'	0.19	0.43	0.47	0.47	0.47	0.48
bus	autobus	0.27	0.57	0.53	0.53	0.53	0.52

$e$	$f$	
the	le	0.94
the	chien	0
the	chat	0
the	l'	0.03
the	autobus	0.02
dog	le	0.06
dog	chien	0.94
dog	chat	0
dog	l'	0
dog	autobus	0
cat	le	0.06
cat	chien	0
cat	chat	0.94
cat	l'	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l'	0.49
bus	autobus	0.51

After 20 iterations:

$e$	$f$	$T(f   e)$
the	le	0.67
the	chien	0
the	chat	0
the	l'	0.33
the	autobus	0
dog	le	0
dog	chien	1
dog	chat	0
dog	l'	0
dog	autobus	0
cat	le	0
cat	chien	0
cat	chat	1
cat	l'	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l'	0
bus	autobus	1

Model 2 has several local maxima – good one:

$e$	$f$	$T(f   e)$
the	le	0
the	chien	0.4
the	chat	0.3
the	l'	0
the	autobus	0.3
dog	le	0.5
dog	chien	0.5
dog	chat	0
dog	l'	0
dog	autobus	0
cat	le	0.5
cat	chien	0
cat	chat	0.5
cat	l'	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l'	0.5
bus	autobus	0.5

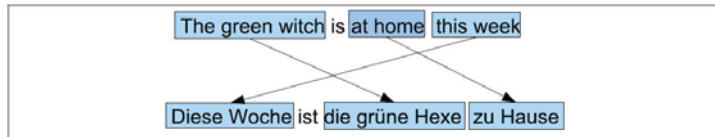
Model 2 has several local maxima – bad one:

$e$	$f$	$T(f   e)$
the	le	0
the	chien	0.33
the	chat	0.33
the	l'	0
the	autobus	0.33
dog	le	1
dog	chien	0
dog	chat	0
dog	l'	0
dog	autobus	0
cat	le	1
cat	chien	0
cat	chat	0
cat	l'	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l'	1
bus	autobus	0

another bad one:

# The Phrase-based Translation Model

- Modern SMT considers a better way to compute translation model  $P(\mathbf{f}|\mathbf{e})$  is based on **phrases**.



- Generative Story:
  - Group English sentence  $\mathbf{e}$  into phrases,  $\bar{e}_1, \bar{e}_2, \dots, \bar{e}_i$
  - Translate each English phrase  $\bar{e}_i$  into a French phrase  $\bar{f}_i$
  - Optionally reorder each of the French phrases.

# The Phrase-based Translation Model

$$P(\mathbf{f} | \mathbf{e}) = \prod_{i=1}^I \phi(\bar{f}_i, \bar{e}_i) d(a_i - b_{i-1})$$

$\phi(\bar{f}_i, \bar{e}_i)$ : translation probability

$d(a_i - b_{i-1})$ : distortion probability:

$a_i$ : starting position of the foreign word (French) generated by  $\bar{e}_i$

$b_{i-1}$ : end position of the foreign word generated by  $\bar{e}_{i-1}$

$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|}$$

Parameter estimation using parallel training data. But we don't have large, hand labeled phrase-aligned training sets.

We can extract phrases based on word alignment

# Extracting Phrases

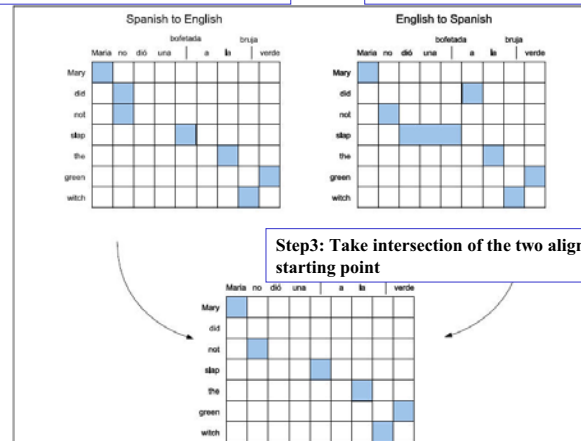
- A training example of Spanish/English sentence pair (from Koehn and Knight Tutorial):
  - Spanish: Maria no daba una bofetada a la bruja verde
  - English: Mary did not slap the green witch
- Some phrase pairs (not all) extracted from this example:
  - (Maria ↔ Mary), (bruja ↔ witch), (verde ↔ green), (no ↔ did not)
  - (no daba una bofetada ↔ did not slap), (daba una bofetada a la ↔ slap the)
- The phrases can be extracted using alignment from the IBM models (e.g., from IBM Model 2)
  - Once we trained the model, for any  $(\mathbf{f}, \mathbf{e})$  pair, we can identify the most likely alignment  $\mathbf{a}^*$  under the model:

$$\mathbf{a}^* = \arg \max_{\mathbf{a}} P(\mathbf{a} | \mathbf{f}, \mathbf{e}) = \arg \max_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} | \mathbf{e})$$

# Finding Alignment Matrices

Step1: Train IBM model 2 for  $P(\mathbf{e}|\mathbf{f})$ , come up with most likely alignment for each  $(\mathbf{e}, \mathbf{f})$  pair

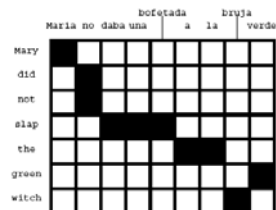
Step2: Train IBM model 2 for  $P(\mathbf{f}|\mathbf{e})$ , come up with most likely alignment for each  $(\mathbf{e}, \mathbf{f})$  pair



Step3: Take intersection of the two alignments as a starting point

## Finding Alignment Matrices

- Apply heuristics for growing alignments
- Only explore alignment in union of  $P(f|e)$  and  $P(e|f)$
- Detailed operation can be found in Och and Ney (2003)



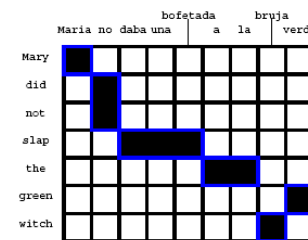
- Allow many-to-one alignment in both directions
- Collect all phrase pairs that are consistent with the word
  - A phrase alignment has to contain all alignment points for all words it covers

4/20/2009

CSE842, Spring 2009, MSU

45

## Word Alignment Induced Phrases (Knight & Koehn Tutorial, 2003)



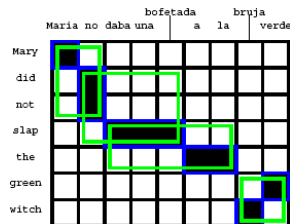
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch),  
(verde, green)

4/20/2009

CSE842, Spring 2009, MSU

46

## Word Alignment Induced Phrases (Knight & Koehn Tutorial, 2003)



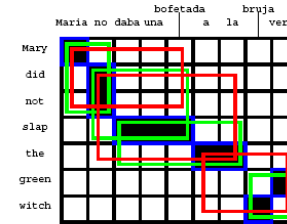
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch),  
(verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap),  
(daba una bofetada a la, slap the), (bruja verde, green witch)

4/20/2009

CSE842, Spring 2009, MSU

47

## Word Alignment Induced Phrases (Knight & Koehn Tutorial, 2003)



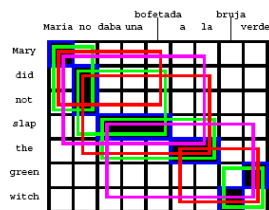
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch),  
(verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap),  
(daba una bofetada a la, slap the), (bruja verde, green witch),  
(Maria no daba una bofetada, Mary did not slap),  
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

4/20/2009

CSE842, Spring 2009, MSU

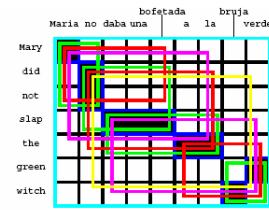
48

# Word Alignment Induced Phrases (Knight & Koehn Tutorial, 2003)



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch),  
 (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap),  
 (daba una bofetada a la, slap the), (bruja verde, green witch),  
 (Maria no daba una bofetada, Mary did not slap),  
 (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),  
 (Maria no daba una bofetada a la, Mary did not slap the),  
 (daba una bofetada a la bruja verde, slap the green witch)

# Word Alignment Induced Phrases (Knight & Koehn Tutorial, 2003)



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch),  
 (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap),  
 (daba una bofetada a la, slap the), (bruja verde, green witch),  
 (Maria no daba una bofetada, Mary did not slap),  
 (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),  
 (Maria no daba una bofetada a la, Mary did not slap the),  
 (daba una bofetada a la bruja verde, slap the green witch),  
 (no daba una bofetada a la bruja verde, did not slap the green witch),  
 (Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)

## Phrase Translation Table

We can store each phrase  $(\bar{f}, \bar{e})$ , together with its probability  $\phi(\bar{f}, \bar{e})$  in a large phrase translation table, where :

$$\phi(\bar{f}, \bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\sum_f \text{count}(\bar{f}, \bar{e})}$$

## Decoding for Phrase-based Models

Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	not	give	a	slap	to	the	witch	green
	did not		a slap		to		green witch	
	no		slap		to the			
	did not give				to			
					the			
			slap			the	witch	

The lattice of all possible English translations for words and phrases in a Spanish sentence

Decoding:

The goal is to identify the English sentence that maximizes the translation and language model probabilities

A\* search (check P. 890-894)