# Word Alignment

David Kauchak
CS159 – Fall 2014

Some slides adapted from

Philipp Koehn
School of Informatics
University of Edinburgh

Kevin Knight
USC/Information Sciences Institute
USC/Computer Science Department

Dan Klein
Computer Science Department
UC Berkeley

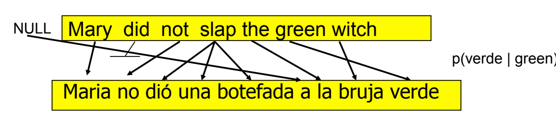
# Admin

Assignment 5

Assignment schedule

# Language translation


# Word models: IBM Model 1




Each foreign word is aligned to exactly one English word

This is the **ONLY** thing we model!

$$p(f_1 f_2 ... f_{|F|}, a_1 a_2 ... a_{|F|} \mid e_1 e_2 ... e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

# Training a word-level model

| | |
|---|---|
| The old man is happy. He has fished many times. | El viejo está feliz porque ha pescado muchos veces. |
| His wife talks to him. | Su mujer habla con él. |
| The sharks await. | Los tiburones esperan. |
| … | … |

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

$p(f_i \mid e_{a_i})$: probability that *e* is translated as *f*

# Thought experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

The sharks await.

Los tiburones esperan.

$p(f_i \mid e_{a_i}) =$ ?

# Thought experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

The sharks await.

Los tiburones esperan.

$$p(f_i \mid e_{a_i}) = \frac{count(f\ aligned\text{-}to\ e)}{count(e)}$$

p(el | the) = 0.5
p(Los | the) = 0.5

Any problems concerns?

# Thought experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

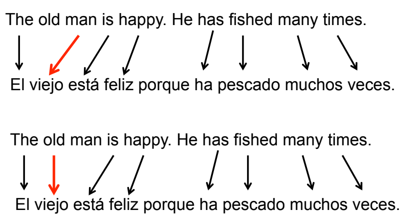
The sharks await.

Los tiburones esperan.

Getting data like this is expensive!

Even if we had it, what happens when we switch to a new domain/corpus

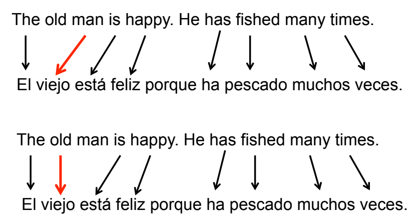
## Thought experiment #2




$$p(f_i | e_{a_i}) = \frac{count(f\ aligned\text{-}to\ e)}{count(e)}$$

What do we do?

## Thought experiment #2




$$p(f_i | e_{a_i}) = \frac{count(f\ aligned\text{-}to\ e)}{count(e)}$$

Use partial counts:
- count(viejo | man) 0.8
- count(viejo | old) 0.2

## Training without alignments

a b

x y

IBM model 1: Each foreign word is aligned to 1 English word (ignore NULL for now)

What are the possible alignments?

## Training without alignments


IBM model 1: Each foreign word is aligned to 1 English word

## Training without alignments


0.01 0.9 0.08 0.01

IBM model 1: Each foreign word is aligned to 1 English word

If I told you how likely each of these were, does that help us with calculating p(f | e)?

## Training without alignments


0.01 0.9 0.08 0.01

IBM model 1: Each foreign word is aligned to 1 English word

$$p(f_i | e_{a_i}) = \frac{count(f \text{ aligned-to } e)}{count(e)}$$

Use partial counts:
- count(y | a) 0.9+0.01
- count(x | a) 0.01+0.08

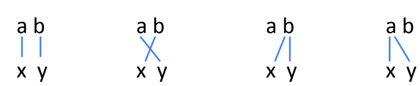
## One the one hand


0.01 0.9 0.08 0.01

If you had the likelihood of each alignment, you could calculate p(f|e)

$$p(f_i | e_{a_i}) = \frac{count(f \text{ aligned-to } e)}{count(e)}$$

## One the other hand




$$p(F, a_1 a_2 \ldots a_{|F|} | E) = \prod_{i=1}^{|F|} p(f_i | e_{a_i})$$

If you had p(f|e) could you calculate the probability of the alignments?

$$p(f_i | e_{a_i})$$

## One the other hand

| a b / x y (a–x, b–y) | a b / x y (a–y, b–x) | a b / x y (b–x, b–y) | a b / x y (a–x, a–y) |
|---|---|---|---|
| $p(x \mid a) * p(y \mid b)$ | $p(x \mid b) * p(y \mid a)$ | $p(x \mid b) * p(y \mid b)$ | $p(x \mid a) * p(y \mid a)$ |

$$p(F, a_1 a_2 \ldots a_{|F|} \mid E) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

$$p(f_i \mid e_{a_i})$$

## Have we gotten anywhere?

## Training without alignments

Initially assume a p(f|e) are equally probable

Repeat:

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))
- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

## EM algorithm

(*something from nothing*)

General approach for calculating "hidden variables", i.e. variables without explicit labels in the data

Repeat:

E-step: Calculate the expected probabilities of the hidden variables based on the current model

M-step: Update the model based on the expected counts/probabilities

# EM alignment

E-step
- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step
- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

green house

casa verde

the house

la casa

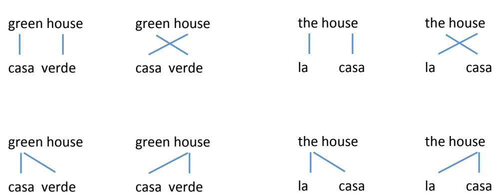
What are the different p(f|e) that make up my model?

| | |
|---|---|
| p( casa \| green) | |
| p( verde \| green) | |
| p( la \| green ) | |

| | |
|---|---|
| p( casa \| house) | |
| p( verde \| house) | |
| p( la \| house ) | |

| | |
|---|---|
| p( casa \| the) | |
| p( verde \| the) | |
| p( la \| the ) | |

Technically, all combinations of foreign and English words

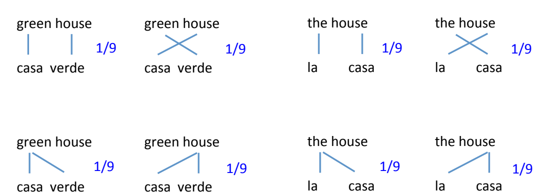



| | |
|---|---|
| p( casa \| green) | 1/3 |
| p( verde \| green) | 1/3 |
| p( la \| green ) | 1/3 |

| | |
|---|---|
| p( casa \| house) | 1/3 |
| p( verde \| house) | 1/3 |
| p( la \| house ) | 1/3 |

| | |
|---|---|
| p( casa \| the) | 1/3 |
| p( verde \| the) | 1/3 |
| p( la \| the ) | 1/3 |

Start with all p(f|e) equally probable



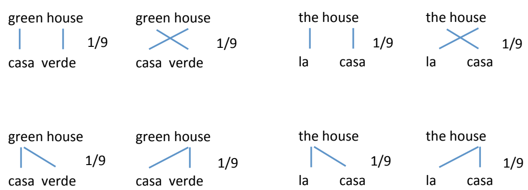

| | |
|---|---|
| p( casa \| green) | 1/3 |
| p( verde \| green) | 1/3 |
| p( la \| green ) | 1/3 |

| | |
|---|---|
| p( casa \| house) | 1/3 |
| p( verde \| house) | 1/3 |
| p( la \| house ) | 1/3 |

| | |
|---|---|
| p( casa \| the) | 1/3 |
| p( verde \| the) | 1/3 |
| p( la \| the ) | 1/3 |

E-step: What are the probabilities of the alignments?

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$




M-step: What are the p(f|e) given the alignments?

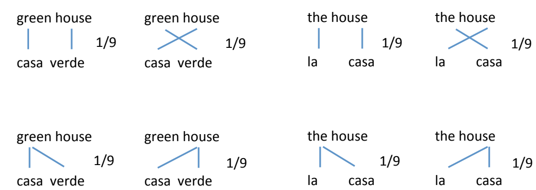
| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | 1/3 | p( casa \| house) | 1/3 | p( casa \| the) | 1/3 |
| p( verde \| green) | 1/3 | p( verde \| house) | 1/3 | p( verde \| the) | 1/3 |
| p( la \| green ) | 1/3 | p( la \| house ) | 1/3 | p( la \| the ) | 1/3 |

c(casa,green) = ?
c(verde,green) = ?
c(la, green) = ?

c(casa,house) ?
c(verde,house) = ?
c(la,house) = ?

c(casa,the) = ?
c(verde,the) = ?
c(la,the) = ?

First, calculate the partial counts




M-step: What are the p(f|e) given the alignments?

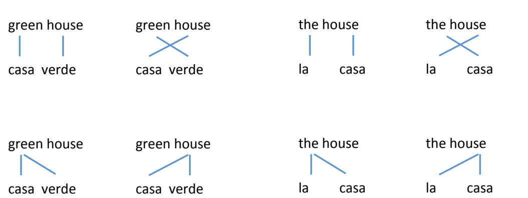
| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | ? | p( casa \| house) | ? | p( casa \| the) | ? |
| p( verde \| green) | ? | p( verde \| house) | ? | p( verde \| the) | ? |
| p( la \| green ) | ? | p( la \| house ) | ? | p( la \| the ) | ? |

c(casa,green) = 1/9+1/9 = 2/9
c(verde,green) = 1/9+1/9 = 2/9
c(la, green) = 0

c(casa,house) = 1/9+1/9+
1/9+1/9 = 4/9
c(verde,house) = 1/9+1/9 = 2/9
c(la,house) = 1/9+1/9 = 2/9

c(casa,the) = 1/9+1/9 = 2/9
c(verde,the) = 0
c(la,the) = 1/9+1/9 = 2/9

Then, calculate the probabilities by normalizing the counts



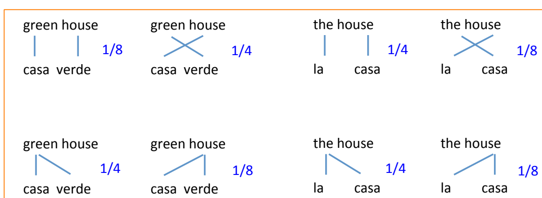

| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | 1/2 | p( casa \| house) | 1/2 | p( casa \| the) | 1/2 |
| p( verde \| green) | 1/2 | p( verde \| house) | 1/4 | p( verde \| the) | 0 |
| p( la \| green ) | 0 | p( la \| house ) | 1/4 | p( la \| the ) | 1/2 |

c(casa,green) = 1/9+1/9 = 2/9
c(verde,green) = 1/9+1/9 = 2/9
c(la, green) = 0

c(casa,house) = 1/9+1/9+
1/9+1/9 = 4/9
c(verde,house) = 1/9+1/9 = 2/9
c(la,house) = 1/9+1/9 = 2/9

c(casa,the) = 1/9+1/9 = 2/9
c(verde,the) = 0
c(la,the) = 1/9+1/9 = 2/9

E-step: What are the probabilities of the alignments?

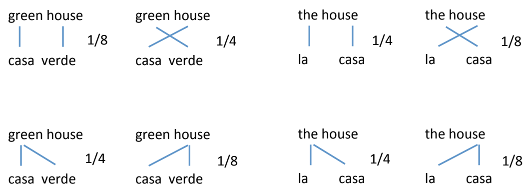



| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | 1/2 | p( casa \| house) | 1/2 | p( casa \| the) | 1/2 |
| p( verde \| green) | 1/2 | p( verde \| house) | 1/4 | p( verde \| the) | 0 |
| p( la \| green ) | 0 | p( la \| house ) | 1/4 | p( la \| the ) | 1/2 |

c(casa,green) = 1/9+1/9 = 1/3
c(verde,green) = 1/9+1/9 = 1/3
c(la, green) = 0

c(casa,house) = 1/9+1/9+
1/9+1/9 = 2/3
c(verde,house) = 1/9+1/9 = 1/3
c(la,house) = 1/9+1/9 = 1/3

c(casa,the) = 1/9+1/9 = 1/3
c(verde,the) = 0
c(la,the) = 1/9+1/9 = 1/3




## M-step: What are the p(f|e) given the alignments?

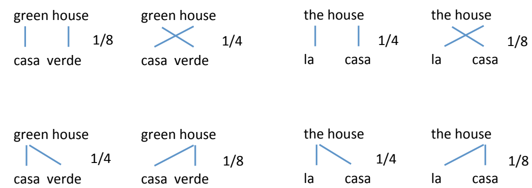
| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | 1/2 | p( casa \| house) | 1/2 | p( casa \| the) | 1/2 |
| p( verde \| green) | 1/2 | p( verde \| house) | 1/4 | p( verde \| the) | 0 |
| p( la \| green ) | 0 | p( la \| house ) | 1/4 | p( la \| the ) | 1/2 |

c(casa,green) = ?
c(verde,green) = ?
c(la, green) = ?

c(casa,house) = ?
c(verde,house) = ?
c(la,house) = ?

c(casa,the) = ?
c(verde,the) = ?
c(la,the) = ?

First, calculate the partial counts



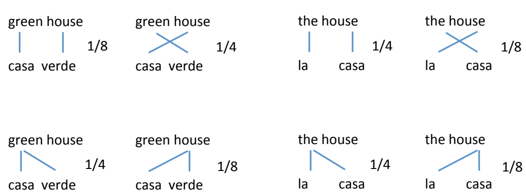

| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | 1/2 | p( casa \| house) | 1/2 | p( casa \| the) | 1/2 |
| p( verde \| green) | 1/2 | p( verde \| house) | 1/4 | p( verde \| the) | 0 |
| p( la \| green ) | 0 | ( la \| house ) | 1/4 | p( la \| the ) | 1/2 |

c(casa,green) = 1/8+1/4 = 3/8
c(verde,green) = 1/4+1/4 = 1/2
c(la, green) = 0

c(casa,house) = 1/4+1/8+
1/4+1/8 = 3/4
c(verde,house) = 1/8+1/8 = 1/4
c(la,house) = 1/8+1/8 = 1/4

c(casa,the) = 1/8+1/4 = 3/8
c(verde,the) = 0
c(la,the) = 1/4+1/4 = 1/2

Then, calculate the probabilities by normalizing the counts



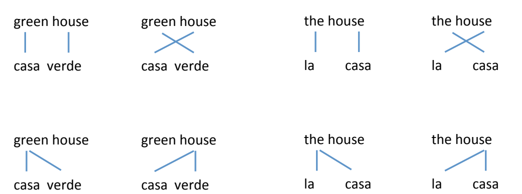

## M-step: What are the p(f|e) given the alignments?

| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | ? | p( casa \| house) | ? | p( casa \| the) | ? |
| p( verde \| green) | ? | p( verde \| house) | ? | p( verde \| the) | ? |
| p( la \| green ) | ? | p( la \| house ) | ? | p( la \| the ) | ? |

c(casa,green) = 1/8+1/4 = 3/8
c(verde,green) = 1/4+1/4 = 1/2
c(la, green) = 0

c(casa,house) = 1/4+1/8+
1/4+1/8 = 3/4
c(verde,house) = 1/8+1/8 = 1/4
c(la,house) = 1/8+1/8 = 1/4

c(casa,the) = 1/8+1/4 = 3/8
c(verde,the) = 0
c(la,the) = 1/4+1/4 = 1/2




| | | | | | |
|---|---|---|---|---|---|
| p( casa \| green) | 3/7 | p( casa \| house) | 3/5 | p( casa \| the) | 3/7 |
| p( verde \| green) | 4/7 | p( verde \| house) | 1/5 | p( verde \| the) | 0 |
| p( la \| green ) | 0 | p( la \| house ) | 1/5 | p( la \| the ) | 4/7 |

c(casa,green) = 1/8+1/4 = 3/8
c(verde,green) = 1/4+1/4 = 1/2
c(la, green) = 0

c(casa,house) = 1/4+1/8+
1/4+1/8 = 3/4
c(verde,house) = 1/8+1/8 = 1/4
c(la,house) = 1/8+1/8 = 1/4

c(casa,the) = 1/8+1/4 = 3/8
c(verde,the) = 0
c(la,the) = 1/4+1/4 = 1/2

green house / casa verde — 3/7 * 1/5 = 3/35 (.086)
green house / casa verde — 4/7 * 3/5 = 12/35 (.34)
the house / la casa — 4/7 * 3/5= 12/35 (.34)
the house / la casa — 3/7 * 1/5 = 3/35 (.086)

green house / casa verde — 3/7* 4/7= 12/49 (.24)
green house / casa verde — 3/5* 1/5= 3/25 (.12)
the house / la casa — 4/7 * 3/7 = 12/49 (.24)
the house / la casa — 1/5 * 3/5 = 3/25 (.12)

| | |
|---|---|
| p( casa \| green) | 3/7 |
| p( verde \| green) | 4/7 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | 3/5 |
| p( verde \| house) | 1/5 |
| p( la \| house ) | 1/5 |

| | |
|---|---|
| p( casa \| the) | 3/7 |
| p( verde \| the) | 0 |
| p( la \| the ) | 4/7 |

c(casa,green) = 1/8+1/4 = 3/8
c(verde,green) = 1/4+1/4 = 1/2
c(la, green) = 0

c(casa,house) = 1/4+1/8+ 1/4+1/8 = 3/4
c(verde,house) = 1/8+1/8 = 1/4
c(la,house) = 1/8+1/8 = 1/4

c(casa,the) = 1/8+1/4 = 3/8
c(verde,the) = 0
c(la,the) = 1/4+1/4 = 1/2

---

green house / casa verde — 3/7 * 1/5 = 3/35 (.086)
green house / casa verde — 4/7 * 3/5 = 12/35 (.343)
the house / la casa — 4/7 * 3/5= 12/35 (.343)
the house / la casa — 3/7 * 1/5 = 3/35 (.086)

green house / casa verde — 3/7* 4/7= 12/49 (.245)
green house / casa verde — 3/5* 1/5= 3/25 (.12)
the house / la casa — 4/7 * 3/7 = 12/49 (.245)
the house / la casa — 1/5 * 3/5 = 3/25 (.12)

| | |
|---|---|
| p( casa \| green) | 3/7 |
| p( verde \| green) | 4/7 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | 3/5 |
| p( verde \| house) | 1/5 |
| p( la \| house ) | 1/5 |

| | |
|---|---|
| p( casa \| the) | 3/7 |
| p( verde \| the) | 0 |
| p( la \| the ) | 4/7 |

c(casa,green) = .086+.245=0.331
c(verde,green) = .343+0.245 = 0.588
c(la, green) = 0

c(casa,house) = .343+.12+ .343+.12=0.926
c(verde,house) = .086+.12=0.206
c(la,house) = .086+.12=0.206

c(casa,the) = .086+.245=0.331
c(verde,the) = 0
c(la,the) = .343+.245=0.588

---

green house / casa verde
green house / casa verde
the house / la casa
the house / la casa

green house / casa verde
green house / casa verde
the house / la casa
the house / la casa

| | |
|---|---|
| p( casa \| green) | 0.36 |
| p( verde \| green) | 0.64 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | 0.69 |
| p( verde \| house) | 0.15 |
| p( la \| house ) | 0.15 |

| | |
|---|---|
| p( casa \| the) | 0.36 |
| p( verde \| the) | 0 |
| p( la \| the ) | 0.64 |

c(casa,green) = .086+.245=0.331
c(verde,green) = .343+0.245 = 0.588
c(la, green) = 0

c(casa,house) = .343+.12+ .343+.12=0.926
c(verde,house) = .086+.12=0.206
c(la,house) = .086+.12=0.206

c(casa,the) = .086+.245=0.331
c(verde,the) = 0
c(la,the) = .343+.245=0.588

---

# Iterate…

**5 iterations**

| | |
|---|---|
| p( casa \| green) | 0.24 |
| p( verde \| green) | 0.76 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | 0.84 |
| p( verde \| house) | 0.08 |
| p( la \| house ) | 0.08 |

| | |
|---|---|
| p( casa \| the) | 0.24 |
| p( verde \| the) | 0 |
| p( la \| the ) | 0.76 |

**10 iterations**

| | |
|---|---|
| p( casa \| green) | 0.1 |
| p( verde \| green) | 0.9 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | 0.98 |
| p( verde \| house) | 0.01 |
| p( la \| house ) | 0.01 |

| | |
|---|---|
| p( casa \| the) | 0.1 |
| p( verde \| the) | 0 |
| p( la \| the ) | 0.9 |

**100 iterations**

| | |
|---|---|
| p( casa \| green) | 0.005 |
| p( verde \| green) | 0.995 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | ~1.0 |
| p( verde \| house) | ~0.0 |
| p( la \| house ) | ~0.0 |

| | |
|---|---|
| p( casa \| the) | 0.005 |
| p( verde \| the) | 0 |
| p( la \| the ) | 0.995 |

## EM alignment

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

Why does it work?

## EM alignment

E-ste

–

–  der the

M-st

–  nents,


Why does it work?

## EM alignment

Intuitively:

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

Things that co-occur will have higher probabilities

E-step

- Calculate how probable the alignments are under the current model (i.e. p(f|e))

Alignments that contain things with higher p(f|e) will be scored higher

## An aside: estimating probabilities

What is the probability of "the" occurring in a sentence?

$$\frac{\text{number of sentences with "the"}}{\text{total number of sentences}}$$

Is this right?

## Estimating probabilities

What is the probability of "the" occurring in a sentence?

$$\frac{\text{number of sentences with "the"}}{\text{total number of sentences}}$$

No. This is an *estimate* based on our data

This is called the maximum likelihood estimation. Why?

## Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation picks the values for the model parameters that maximize the likelihood of the training data

You flip a coin 100 times. 60 times you get heads.

What is the MLE for heads?

p(head) = 0.60

## Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation picks the values for the model parameters that maximize the likelihood of the training data

You flip a coin 100 times. 60 times you get heads.

What is the likelihood of the data under this model (each coin flip is a data point)?

## MLE example

You flip a coin 100 times. 60 times you get heads.

MLE for heads: p(head) = 0.60

What is the likelihood of the data under this model (each coin flip is a data point)?

$$likelihood(data) = \prod_i p(x_i)$$

$\log(0.60^{60} * 0.40^{40}) = -67.3$

# MLE example

Can we do any better?

$$likelihood(data) = \prod_i p(x_i)$$

p(heads) = 0.5

log($0.50^{60}$ * $0.50^{40}$) =-69.3

p(heads) = 0.7

– log($0.70^{60}$ * $0.30^{40}$)=-69.5

# EM alignment: the math

The EM algorithm tries to find parameters to the model (in our case, p(f|e)) that maximize the likelihood of the data

In our case:

$$p(f_1 f_2 \ldots f_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \sum_{a_1} \sum_{a_2} \ldots \sum_{a_{|F|}} p(f_i \mid e_{a_i})$$

Each iteration, we increase (or keep the same) the likelihood of the data

# Implementation details

Any concerns/issues?
Anything underspecified?

Repeat:

- E-step
  - Enumerate all possible alignments
  - Calculate how probable the alignments are under the current model (i.e. p(f|e))
- M-step
  - Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

# Implementation details

When do we stop?

Repeat:

- E-step
  - Enumerate all possible alignments
  - Calculate how probable the alignments are under the current model (i.e. p(f|e))
- M-step
  - Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

## Implementation details

- Repeat for a fixed number of iterations
- Repeat until parameters don't change (much)
- Repeat until likelihood of data doesn't change much

Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

## Implementation details

For |E| English words and |F| foreign words, how many alignments are there?

Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

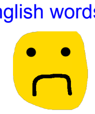
M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

## Implementation details

Each foreign word can be aligned to any of the English words (or NULL)

$(|E|+1)^{|F|}$



Repeat:

E-step

- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. p(f|e))

M-step

- Recalculate p(f|e) using counts from **all** alignments, **weighted** by how probable they are

## Thought experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

The sharks await.

Los tiburones esperan.

$$p(f_i \mid e_{a_i}) = \frac{count(f\ aligned\text{-}to\ e)}{count(e)}$$

p(el | the) = 0.5
p(Los | the) = 0.5

## If we had the alignments…

Input: corpus of English/Foreign sentence pairs along with alignment

```
for (E, F) in corpus:
    for aligned words (e, f) in pair (E,F):
        count(e,f) += 1
        count(e) += 1

for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

## If we had the alignments…

Input: corpus of English/Foreign sentence pairs along with alignment

```
for (E, F) in corpus:
    for e in E:
        for f in F:
            if f aligned-to e:
                count(e,f) += 1
                count(e) += 1

for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

## If we had the alignments…

Input: corpus of English/Foreign sentence pairs along with alignment

```
for (E, F) in corpus:
    for aligned words (e, f) in pair (E,F):
        count(e,f) += 1
        count(e) += 1
```

```
for (E, F) in corpus
    for e in E:
        for f in F:
            if f aligned-to e:
                count(e,f) += 1
                count(e) += 1
```

Are these equivalent?

```
for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

## Without the alignments

Input: corpus of English/Foreign sentence pairs along with alignment

```
for (E, F) in corpus:
    for e in E:
        for f in F:
            p(f -> e): probability that f is aligned to e in this pair
            count(e,f) += p( f -> e)
            count(e) += p(f -> e)

for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

## Without alignments

p(f -> e): probability that f is aligned to e *in this pair*

a b c

y z

What is p(y -> a)?

Put another way, of all things that y could align to, how likely is it to be a?

## Without alignments

p(f -> e): probability that f is aligned to e *in this pair*

a b c

y z

Of all things that y could align to, how likely is it to be a:

p(y | a)

Does that do it?

No! p(y | a) is how likely y is to align to a over the whole data set.

## Without alignments

p(f -> e): probability that f is aligned to e *in this pair*

a b c

y z

Of all things that y could align to, how likely is it to be a:

$$\frac{p(y \mid a)}{p(y \mid a) + p(y \mid b) + p(y \mid c)}$$

## Without the alignments

Input: corpus of English/Foreign sentence pairs along with alignment

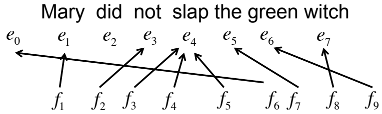
```
for (E, F) in corpus:
    for e in E:
        for f in F:
            p(f -> e) = p(f | e) / ( sum_(e in E) p( f | e) )
            count(e,f) += p( f -> e)
            count(e) += p(f -> e)

for all (e,f) in count:
    p(f|e) = count(e,f) / count(e)
```

# Benefits of word-level model

Rarely used in practice for modern MT system

Mary did not slap the green witch




Maria no dió una botefada a la bruja verde

Two key side effects of training a word-level model:

- Word-level alignment
- p(f | e): translation dictionary

How do I get this?

# Word alignment

100 iterations

| | |
|---|---|
| p( casa \| green) | 0.005 |
| p( verde \| green) | 0.995 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | ~1.0 |
| p( verde \| house) | ~0.0 |
| p( la \| house ) | ~0.0 |

| | |
|---|---|
| p( casa \| the) | 0.005 |
| p( verde \| the) | 0 |
| p( la \| the ) | 0.995 |

green house

casa verde

How should these be aligned?

the house

la casa

# Word alignment

100 iterations

| | |
|---|---|
| p( casa \| green) | 0.005 |
| p( verde \| green) | 0.995 |
| p( la \| green ) | 0 |

| | |
|---|---|
| p( casa \| house) | ~1.0 |
| p( verde \| house) | ~0.0 |
| p( la \| house ) | ~0.0 |

| | |
|---|---|
| p( casa \| the) | 0.005 |
| p( verde \| the) | 0 |
| p( la \| the ) | 0.995 |

green house

casa verde

Why?

the house

la casa

# Word-level alignment

$$alignment(E,F) = \arg_A \max p(A,F \mid E)$$

Which for IBM model 1 is:

$$alignment(E,F) = \arg_A \max \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

Given a model (i.e. trained p(f|e)), how do we find this?

Align each foreign word (f in F) to the English word (e in E) with highest p(f|e)

$$a_i = \arg_{j:1-|E|} \max p(f_i \mid e_j)$$

## Word-alignment Evaluation

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

How good of an alignment is this?
How can we quantify this?

## Word-alignment Evaluation

System:

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Human

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

How can we quantify this?

## Word-alignment Evaluation

System:

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Human

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Precision and recall!

## Word-alignment Evaluation

System:

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Human

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Precision: $\frac{6}{7}$ Recall: $\frac{6}{10}$