Language translation

Mary did not slap the green witch
Maria no dio una botefada a la bruja verde

Each foreign word is aligned to exactly one English word
This is the ONLY thing we model!

\[ p(f_1, f_2, \ldots, f_F, a_1, a_2, \ldots, a_M | e_1, e_2, \ldots, e_E) = \prod_{i=1}^{M} p(f_i | e_i) \]
Training a word-level model

The old man is happy. He has fished many times.
His wife talks to him.
The sharks await.

\[
p(f_1, f_2, \ldots, f_F | a_1, a_2, \ldots, a_E) = \prod_{i=1}^{F} p(f_i | e_a)
\]

\[p(f_i | e_a): \text{probability that } e_i \text{ is translated as } f_i\]

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.

\[p(f_i | e_a) = ?\]
Thought experiment #2

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

El viejo está feliz porque ha pescado muchos veces.

\[ p(f|e) = \frac{\text{count}(f \text{ aligned-to } e)}{\text{count}(e)} \]

What do we do?

Training without alignments

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
</tbody>
</table>

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

What are the possible alignments?

Use partial counts:
- \( \text{count(viejo | man)} 0.8 \)
- \( \text{count(viejo | old)} 0.2 \)
Training without alignments

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

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>x</th>
<th>y</th>
<th>a</th>
<th>b</th>
<th>x</th>
<th>y</th>
<th>a</th>
<th>b</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.9</td>
<td>0.08</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

One the one hand

\[
p(f_i | e_a) = \frac{\text{count}(f \text{-aligned-to } e)}{\text{count}(e)}
\]

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

One the other hand

\[
p(F, a_1, a_2, \ldots | E) = \prod_{i=1}^{n} p(f_i | e_a)
\]

If you had \( p(f | e) \) could you calculate the probability of the alignments?
**One the other hand**

\[
p(F, a_1 a_2 a_3 | E) = \prod_{i=1}^{n} p(f_i | e_{a_i})
\]

**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

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

Technically, all combinations of foreign and English words

Start with all \( p(f|e) \) equally probable

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

\[
p(f_1, f_2, \ldots, f_N | \epsilon_1, \epsilon_2, \ldots, \epsilon_N) = \prod_{i=1}^{N} p(f_i | \epsilon_i)
\]
E-step: What are the probabilities of the alignments?

M-step: What are the p(\text{fe}) given the alignments?

First, calculate the partial counts

Then, calculate the probabilities by normalizing the counts
First, calculate the partial counts:

<table>
<thead>
<tr>
<th></th>
<th>green house</th>
<th>green house</th>
<th>the house</th>
<th>the house</th>
</tr>
</thead>
<tbody>
<tr>
<td>casa</td>
<td>1/8</td>
<td>1/4</td>
<td>1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>verde</td>
<td>1/4</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
</tbody>
</table>

M-step: What are the \( p(la | green) \)?

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

\( c(la, green) = ? \) \( c( verde, green) = ? \) \( c(la, verde) = ? \) \( c(la, house) = ? \)

\( p(la | casa) = 1/4 + 1/4 = 1/2 \)

\( p(la | green) = 1/8 + 1/8 = 1/4 \)

Then, calculate the probabilities by normalizing the counts:

<table>
<thead>
<tr>
<th></th>
<th>green house</th>
<th>green house</th>
<th>the house</th>
<th>the house</th>
</tr>
</thead>
<tbody>
<tr>
<td>casa</td>
<td>1/8</td>
<td>1/4</td>
<td>1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>verde</td>
<td>1/4</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
</tbody>
</table>

M-step: What are the \( p(la | the) \)?

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

\( c(la, the) = ? \) \( c( verde, the) = ? \) \( c(la, verde, the) = ? \) \( c(la, house, the) = ? \)

\( p(la | the) = 1/4 + 1/4 + 1/2 = 3/4 \)

\( p(la | house) = 1/8 + 1/8 = 1/4 \)

\( p(la | green) = 1/8 + 1/8 = 1/4 \)

\( p(la | green) = 1/8 + 1/8 = 1/4 \)

\( p(la | the) = 1/4 + 1/4 + 1/2 = 3/4 \)

\( p(la | house) = 1/8 + 1/8 = 1/4 \)

\( p(la | green) = 1/8 + 1/8 = 1/4 \)
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?

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?

\[p(\text{head}) = 0.60\]

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)?

\[
\text{likelihood(data)} = \prod p(x_i)
\]

\[
\log(0.60^{60} \times 0.40^{40}) = -67.3
\]
MLE example

Can we do any better?

\[ \text{likelihood(data)} = \prod p(x_i) \]

\[ p(\text{heads}) = 0.5 \]
\[ \log(0.5^{60} \times 0.5^{40}) = -69.3 \]

\[ p(\text{heads}) = 0.7 \]
\[ - \log(0.7^{60} \times 0.3^{40}) = -69.5 \]

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:

\[ p(f_1, f_2, \ldots | e_1, e_2, \ldots) = \sum_a \sum_e p(f_i | e_a) \]

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, \textbf{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, \textbf{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

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.

\[
p(f_i \mid e_j) = \frac{\text{count}(f_i \text{ aligned-to } e_j)}{\text{count}(e_j)}
\]

\( p(\text{el} \mid \text{the}) = 0.5 \)

\( p(\text{Los} \mid \text{the}) = 0.5 \)
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)

Are these equivalent?

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

\[ p(f \rightarrow e) \]: probability that \( f \) is aligned to \( e \) \textit{in this pair}

\[
\begin{align*}
  a & \quad b & \quad c \\
  y & \quad z
\end{align*}
\]

What is \( p(y \rightarrow a) \)?

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

\[ p(y | a) \]

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

\[ p(y | a) + p(y | b) + p(y | c) \]

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

---

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

for \((E, F)\) in corpus:
  for \( e \) in \( E \):
    for \( f \) in \( F \):
      \[ p(f \rightarrow e) = \frac{p(f | e)}{\text{sum}_{e \in E} p(f | e)} \]
      \[ \text{count}(e,f) += p(f \rightarrow e) \]
      \[ \text{count}(e) += p(f \rightarrow e) \]
  for all \((e,f)\) in \(\text{count}\):
    \[ p(f|e) = \frac{\text{count}(e,f)}{\text{count}(e)} \]
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

Word alignment

100 iterations

<table>
<thead>
<tr>
<th>F</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>casa</td>
<td>green</td>
</tr>
<tr>
<td>verde</td>
<td>green</td>
</tr>
<tr>
<td>(la</td>
<td>green)</td>
</tr>
<tr>
<td>casa</td>
<td>house</td>
</tr>
<tr>
<td>verde</td>
<td>house</td>
</tr>
<tr>
<td>(la</td>
<td>house)</td>
</tr>
<tr>
<td>casa</td>
<td>the</td>
</tr>
<tr>
<td>verde</td>
<td>the</td>
</tr>
<tr>
<td>(la</td>
<td>the)</td>
</tr>
</tbody>
</table>

Why?

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 \( p(f|e) \)

\[
a_i = \arg_{f \\ E} \max \ p(f_i|e_i)
\]
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?

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!

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