Word Alignment

David Kauchak
CS159 – Fall 2019

Some slides adapted from
Philipp Koehn
University of Edinburgh

Kevin Knight
USC/Information Sciences Institute

Dan Klein
USC/Computer Science Department

School of Informatics
University of Edinburgh

Language translation

Yo quiero
Taco Bell

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

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

$$p(f_1, f_2, \ldots, f_F | a_1, a_2, \ldots, a_E, e_1, e_2, \ldots, e_E) = \prod_{i=1}^{F} p(f_i | e_{a_i})$$
Training a word-level model

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

\[ p(f_1 f_2 \ldots f_{11} | a_1 a_2 \ldots a_{11} e_1 e_2 \ldots e_{11}) = \prod_{i=1}^{11} p(f_i | e_i) \]

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

How do we learn these?

What data would be useful?

Thought experiment

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

El viejo está feliz porque ha pescado muchos veces. Su mujer habla con él. Los tiburones esperan.

\[ p(f_i | e_i) = ? \]

Getting data like this is expensive!

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

Thought experiment

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

\[ p(e | \text{the}) = 0.5 \]

\[ p(Los | \text{the}) = 0.5 \]

Any problems concerns?
Thought experiment #2

The old man is happy. He has fished many times.
El viejo está feliz porque ha pescado muchos veces.

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

What do we do?

Use partial counts:
- \( \text{count(\text{viejo} \mid \text{man})} \) 0.8
- \( \text{count(\text{viejo} \mid \text{old})} \) 0.2

Training without alignments

a b
x y

How should these be aligned?

There is some information!
(Think of the alien translation task last time)
Training without alignments

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

What are the possible alignments?

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

$p(f \mid e_a) = \frac{\text{count}(f \text{ aligned-to } e)}{\text{count}(e)}$ Use partial counts and sum:
- $\text{count}(y \mid a)$ 0.9+0.01
- $\text{count}(x \mid a)$ 0.01+0.01
One the one hand

\[
\begin{array}{cccc}
a & b & x & y \\
\hline
0.01 & 0.9 & 0.08 & 0.01
\end{array}
\]

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

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

One the other hand

\[
\begin{array}{cccc}
a & b & \ X & \ Y \\
\hline
x & y & x & y \\
\hline
\end{array}
\]

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

\[
p(F, a_1, a_2, \ldots, a_n | E) = \prod_i p(f_i | e_i)
\]

One the other hand

We want to calculate the probability of the alignment, e.g.

\[
p(\text{alignment} | F, E) = p(A | F, E)
\]

We can calculate \( p(A_0, F | E) \) using the word probabilities.

\[
p(F, a_1, a_2, \ldots, a_n | E) = \prod_i p(f_i | e_i)
\]

One the other hand

We want to calculate the probability of the alignment, e.g.

\[
p(\text{alignment} | F, E) = p(A | F, E)
\]

We can calculate \( p(A_0, F | E) \) using the word probabilities.

\[
p(A, F | E) = p(A | F, E)
\]

How are these two probabilities related?
Our friend the chain rule

\[ p(A_1, F|E) = p(A_1|F, E) \times p(F|E) \]

\[ p(A_1|F, E) = \frac{p(A_1, F|E)}{p(F|E)} \]

What is \( p(F|E) \)?

Hint: how do we go from \( p(A_1, F|E) \) to \( p(F|E) \)?

---

One the other hand

\[ p(x|a) \times p(y|b) \]

\[ p(F, a|E) \]

\[ p(F, a|E) \]

\[ p(F, a|E) \]

\[ p(F, a, a, \ldots a, a|E) = \prod_{\alpha} p(f_{(\alpha)}|e_{(\alpha)}) \]

Normalize

\[ p(a, |E, F) = \frac{p(x|a) \times p(y|b)}{\sum_{a} p(F, a|E)} \]
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?
1. calculate: \( p(f_1, f_2, \ldots, f_l|a_1, a_2, \ldots, a_l, e_1, e_2, \ldots, e_l) = \prod_{i=1}^{l} p(f_i|e_i) \)

2. normalize: \( p(a|E) = \frac{p(F, a|E)}{\sum_{a} p(F, a|E)} \)
E-step: What are the probabilities of the alignments?

2. normalize: \[ p(\alpha, F) = \frac{p(F; \alpha, E)}{\sum_{\beta = 1}^{N} p(F; \beta, E)} \]

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

First, calculate the partial counts

Then, calculate the probabilities by normalizing the counts
E-step: 1. what are the $p(A,F|E)$?

E-step: 2. what are the alignments, i.e. normalize?

$$p(a_i|E) = \frac{p(F,a_i|E)}{\sum_{a_j} p(F,a_j|E)}$$
M-step: What are the p(f|e) given the alignments?

First, calculate the partial counts.

\[
\begin{align*}
c(casa, green) &= \frac{1}{6} + \frac{1}{3} = \frac{3}{6} \\
c(verde, green) &= \frac{1}{3} + \frac{1}{3} = \frac{4}{6} \\
c(la, green) &= 0 \\
c(casa, house) &= \frac{1}{3} + \frac{1}{6} + \frac{1}{3} + \frac{1}{6} = \frac{6}{6} \\
c(verde, house) &= \frac{1}{6} + \frac{1}{6} = \frac{2}{6} \\
c(la, house) &= \frac{1}{6} + \frac{1}{6} = \frac{2}{6} \\
c(casa, the) &= \frac{1}{6} + \frac{1}{3} = \frac{3}{6} \\
c(verde, the) &= \frac{1}{3} = \frac{2}{6} \\
c(la, the) &= \frac{1}{3} + \frac{1}{3} = \frac{2}{3} \\
\end{align*}
\]

Then, calculate the probabilities by normalizing the counts.

\[
\begin{align*}
p(casa | green) &= \frac{\frac{1}{6} + \frac{1}{3}}{\frac{3}{6} + \frac{4}{6} + 0} = \frac{3}{7} \\
p(verde | green) &= \frac{\frac{1}{3} + \frac{1}{3}}{\frac{3}{6} + \frac{4}{6} + 0} = \frac{4}{7} \\
p(la | green) &= 0 \\
p(casa | house) &= \frac{\frac{1}{3} + \frac{1}{6} + \frac{1}{3} + \frac{1}{6}}{\frac{6}{6} + \frac{2}{6} + \frac{2}{6}} = \frac{3}{5} \\
p(verde | house) &= \frac{\frac{1}{6} + \frac{1}{6}}{\frac{6}{6} + \frac{2}{6} + \frac{2}{6}} = \frac{1}{5} \\
p(la | house) &= \frac{\frac{1}{6} + \frac{1}{6}}{\frac{6}{6} + \frac{2}{6} + \frac{2}{6}} = \frac{1}{5} \\
p(casa | the) &= \frac{\frac{1}{6} + \frac{1}{3}}{\frac{3}{6} + \frac{2}{6} + \frac{2}{3}} = \frac{3}{7} \\
p(verde | the) &= \frac{\frac{1}{3} + \frac{2}{6} + \frac{2}{3}}{\frac{3}{6} + \frac{2}{6} + \frac{2}{3}} = \frac{0}{7} \\
p(la | the) &= \frac{\frac{1}{3} + \frac{2}{3}}{\frac{3}{6} + \frac{2}{6} + \frac{2}{3}} = \frac{4}{7} \\
\end{align*}
\]
E-step: 1. what are the \( p(A|F|E) \)?

- \( p(\text{casa} | \text{green}) = 3/7 \)
- \( p(\text{verde} | \text{green}) = 4/7 \)
- \( p(\text{la} | \text{green}) = 0 \)

- \( p(\text{casa} | \text{house}) = 3/5 \)
- \( p(\text{verde} | \text{house}) = 1/5 \)
- \( p(\text{la} | \text{house}) = 1/5 \)

- \( p(\text{casa} | \text{the}) = 3/7 \)
- \( p(\text{verde} | \text{the}) = 0 \)
- \( p(\text{la} | \text{the}) = 4/7 \)

E-step: 2. what are the alignments, i.e. normalize?

- \( p(\text{casa} | \text{green}) = 0.108 \)
- \( p(\text{verde} | \text{green}) = 0.432 \)
- \( p(\text{la} | \text{green}) = 0.151 \)

- \( p(\text{casa} | \text{house}) = 0.432 \)
- \( p(\text{verde} | \text{house}) = 0.151 \)
- \( p(\text{la} | \text{house}) = 0.300 \)

- \( p(\text{casa} | \text{the}) = 0.360 \)
- \( p(\text{verde} | \text{the}) = 0.640 \)
- \( p(\text{la} | \text{the}) = 0.640 \)

\[
\begin{align*}
\text{c(\text{casa,green})} & = 0.108 + 0.309 = 0.417 \\
\text{c(\text{verde,green})} & = 0.432 + 0.309 = 0.741 \\
\text{c(\text{la,green})} & = 0
\end{align*}
\]

\[
\begin{align*}
\text{c(\text{casa,house})} & = 0.432 + 0.151 = 0.583 \\
\text{c(\text{verde,house})} & = 0.432 + 0.151 = 0.583 \\
\text{c(\text{la,house})} & = 0.300 + 0.151 = 0.451
\end{align*}
\]
**Iterate...**

<table>
<thead>
<tr>
<th>5 iterations</th>
<th>10 iterations</th>
<th>100 iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p(\text{casa} \mid \text{green}) )</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>( p(\text{verde} \mid \text{green}) )</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>( p(\text{la} \mid \text{green}) )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( p(\text{casa} \mid \text{house}) )</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td>( p(\text{verde} \mid \text{house}) )</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>( p(\text{la} \mid \text{house}) )</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>( p(\text{casa} \mid \text{the}) )</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>( p(\text{verde} \mid \text{the}) )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( p(\text{la} \mid \text{the}) )</td>
<td>0.76</td>
<td>0.76</td>
</tr>
</tbody>
</table>

**EM alignment**

**E-step**
- Enumerate all possible alignments
- Calculate how probable the alignments are under the current model (i.e. \( p(\text{f} \mid \text{e}) \))

**M-step**
- Recalculate \( p(\text{f} \mid \text{e}) \) using counts from all alignments, weighted by how probable they are

*Why does it work?*

**Intuitively:**
- **E-step**: Calculate how probable the alignments are under the current model (i.e. \( p(\text{f} \mid \text{e}) \))
- **M-step**: Recalculate \( p(\text{f} \mid \text{e}) \) using counts from all alignments, weighted by how probable they are

Things that co-occur will have higher probabilities

Alignments that contain things with higher \( p(\text{f} \mid \text{e}) \) 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)?
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)?

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

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

MLE example
Can we do any better?

\[
p(\text{heads}) = 0.5
\]

\[
\log(0.50^{60} \times 0.50^{40}) = -69.3
\]

\[
p(\text{heads}) = 0.7
\]

\[
- \log(0.70^{60} \times 0.30^{40}) = -69.5
\]

EM alignment: the math
The EM algorithm tries to find parameters of the model \(p(f|e)\) that maximize the likelihood of the data

In our case:

\[
p(f_1, f_2, \ldots, f_l | e_1, e_2, \ldots, e_m) = \sum_{e_1} \sum_{e_2} \ldots \sum_{e_m} p(f_i | e_a)
\]

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, 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

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.

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

\[
p(\text{el | the}) = 0.5
p(\text{Los | the}) = 0.5
\]
Thought experiment #2

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

El viejo está feliz porque ha pescado muchos veces.

Without the alignments

if f aligned to e:
   count(e,f) += 1
   count(e) += 1

p(f → e) : probability that f is aligned to e in this pair
   count(e,f) += p(f → e)
   count(e) += p(f → e)

Key: use expected counts (i.e., how likely based on the current model), rather than actual counts

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 in this sentence, how likely is it to be a?

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

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

\[ \begin{array}{ccc}
  a & b & c \\
  y & z \\
\end{array} \]

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

\[ p(y | a) = \frac{p(y | a) \cdot p(y | b) \cdot p(y | c)}{p(y | a) + p(y | b) + p(y | c)} \]

EM: without the alignments

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

for some number of iterations:
  for (E, F) in corpus:
    for e in E:
      for f in F:
        \[ p(f \rightarrow e) = \frac{p(f | e)}{\sum_{f \in F} p(f | e)} \]
        count(e, f) += p(f \rightarrow e)
        count(e) += p(f \rightarrow e)

for all (e, f) in count:
  \[ p(f | e) = \frac{\text{count}(e, f)}{\text{count}(e)} \]
**EM: without the alignments**

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

for some number of iterations:
  for \((E, F)\) in corpus:
      for \(e\) in \(E\):
          for \(f\) in \(F\):
              \[ p(f \rightarrow e) = \frac{p(f|e)}{\sum_{e' \in e} p(f|e')} \]
              count\((e, f)\) += \(p(f \rightarrow e)\)

for all \((e, f)\) in count:
  \(p(f|e) = \frac{\text{count}(e, f)}{\text{count}(e)}\)

Where are the E and M steps?

**EM: without the alignments**

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

for some number of iterations:
  for \((E, F)\) in corpus:
      for \(e\) in \(E\):
          for \(f\) in \(F\):
              \[ p(f \rightarrow e) = \frac{p(f|e)}{\sum_{e' \in e} p(f|e')} \]
              count\((e, f)\) += \(p(f \rightarrow e)\)

for all \((e, f)\) in count:
  \(p(f|e) = \frac{\text{count}(e, f)}{\text{count}(e)}\)

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

**NULL**

Sometimes foreign words don't have a direct correspondence to an English word

Adding a NULL word allows for \(p(f \mid \text{NULL})\), i.e. words that appear, but are not associated explicitly with an English word

Implementation: add “NULL” (or some unique string representing NULL) to each of the English sentences, often at the beginning of the sentence

\[
\begin{array}{c|c}
\text{p(casa | NULL)} & 2/3 \\
\text{p(verde | NULL)} & 2/3 \\
\text{p(la | NULL)} & 2/3 \\
\end{array}
\]
Benefits of word-level model

Rarely used in practice for modern MT system

Mary did not slap the green witch
Maria no dio 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

<table>
<thead>
<tr>
<th>Word</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>casa</td>
<td>0.005</td>
</tr>
<tr>
<td>verde</td>
<td>0.995</td>
</tr>
<tr>
<td>la</td>
<td>0</td>
</tr>
<tr>
<td>casa</td>
<td>0.005</td>
</tr>
<tr>
<td>verde</td>
<td>0.005</td>
</tr>
<tr>
<td>la</td>
<td>0.995</td>
</tr>
</tbody>
</table>

green house
casa verde

How should these be aligned?

the house
al casa

Word-level alignment

$alignment(E,F) = \arg \max_A p(A,F \mid E)$

Which for IBM model 1 is:

$alignment(E,F) = \arg \max \prod_{i=1}^{n} p(f_i \mid e_i)$

Given a model (i.e. trained $p(f_i \mid e_i)$), how do we find this?

Align each foreign word ($f$ in $F$) to the English word ($e$ in $E$) with highest $p(f_i \mid e_i)$

$a_i = \arg \max_{A(i)} p(f_i \mid e_i)$
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?

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!

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