Alignment in Machine Translation

CMSC 723 / LING 723 / INST 725

MARINE CARPUAT

marine@cs.umd.edu

Figures credit: Matt Post

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: farok crrrok hihok yorok clok kantok ok-yurp

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 wat dat vat eneat .

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

9a. wiwok nok izok kantok ok-yurp .
9b. totat nnat quat 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 .

Centauri/Arcturan [Knight, 1997]

Your assignment, put these words in order: **{ jjat, arrat, mat, bat, oloat, at-yurp }**

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 wat dat vat eneat .

8a. lalok brok anok plok nok .

8b. iat lat pippat rrat nnat .

9a. wiwok nok izok kantok ok-yurp .

9b. totat nnat quat 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 .

Centauri/Arcturian was actually Spanish/English…

Translate: Clients do not sell pharmaceuticals in Europe.

1a. Garcia and associates .
1b. Garcia y asociados .

2a. Carlos Garcia has three associates .
2b. Carlos Garcia tiene tres asociados .

3a. his associates are not strong .
3b. sus asociados no son fuertes .

4a. Garcia has a company also .
4b. Garcia tambien tiene una empresa .

5a. its clients are angry .
5b. sus clientes estan enfadados .

6a. the associates are also angry .
6b. los asociados tambien estan enfadados .

7a. the clients and the associates are enemies .
7b. los clients y los asociados son enemigos .

8a. the company has three groups .
8b. la empresa tiene tres grupos .

9a. its groups are in Europe .
9b. sus grupos estan en Europa .

10a. the modern groups sell strong pharmaceuticals
10b. los grupos modernos venden medicinas fuertes

11a. the groups do not sell zenzanine .
11b. los grupos no venden zanzanina .

12a. the small groups are not modern .
12b. los grupos pequenos no son modernos .

1988

A Statistical Approach to Machine Translation

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin

IBM
Thomas J. Watson Research Center
Yorktown Heights, NY

In this paper, we present a statistical approach to machine translation. We describe the application of our approach to translation from French to English and give preliminary results.

The COLING Paper Review

The validity of statistical (information theoretic) approach to MT has indeed been recognized, as the authors mention, by Weaver as early as 1949. And was universally recognized as mistaken by 1950. (cf. Hutchins, MT: Past, Present, Future, Ellis Horwood, 1986, pp. 30ff. and references therein) The crude force of computers is not science. The paper is simply beyond the scope of COLING.

More about the IBM story: 20 years of bitext workshop

Noisy Channel Model for Machine Translation

- The **noisy channel model** decomposes machine translation into two independent subproblems
  - Language modeling
  - Translation modeling / Alignment

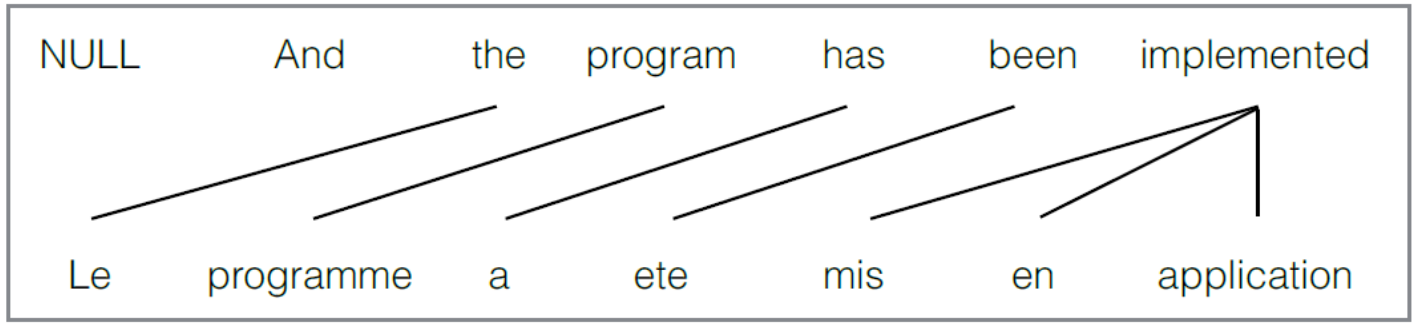
$$\hat{E} = \underset{E \in \text{English}}{\operatorname{argmax}} \overbrace{P(F|E)}^{\text{translation model}} \overbrace{P(E)}^{\text{language model}}$$

Word Alignment

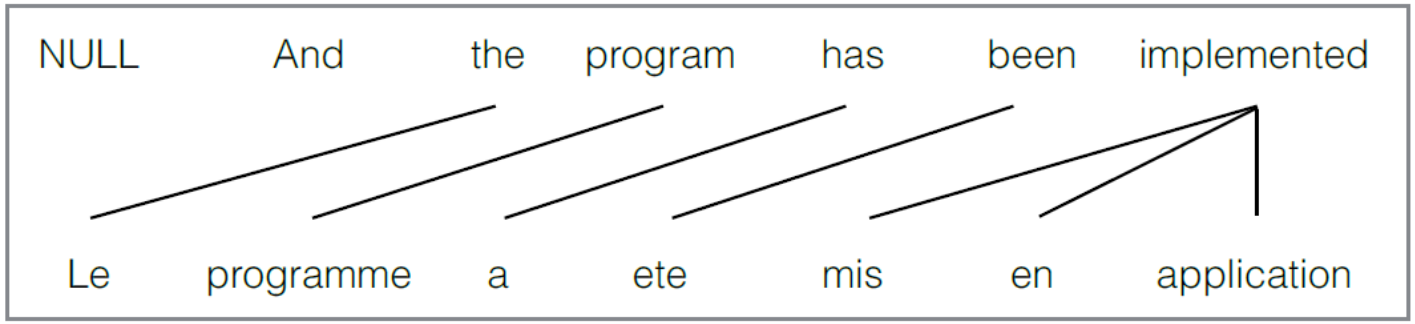
How can we model p(f|e)?

- We'll describe the word alignment models introduced in early 90s at IBM

- Assumption: each French word f is aligned to exactly one English word e
  - Including NULL



Word Alignment
Vector Representation



- Alignment vector a = [2,3,4,5,6,6,6]
  - length of a = length of sentence f
  - a_i = j if French position i is aligned to English position j

Formalizing the connection between word alignments & the translation model

$$p(f_1, f_2, \ldots, f_m \mid e_1, e_2, \ldots, e_l, m)$$
$$= \sum_{a \in A} p(f_1, \ldots, f_m, a_1, \ldots, a_m \mid e_1, \ldots, e_l, m)$$

- We define a conditional model
  - Projecting word translations
  - Through alignment links

How many possible alignments in A?

- How many possible alignments for (f,e) where
  - f is French sentence with m words
  - e is an English sentence with l words

- For each of m French words, we choose an alignment link among (l+1) English words

- Answer: $(l+1)^m$

IBM Model 1: generative story

- Input
  - an English sentence of length l
  - a length m

- For each French position i in $1..\mathrm{m}$
  - Pick an English source index j $$q(j \mid i, l, m) = \frac{1}{l+1}$$
  - Choose a translation $$t(f_i \mid e_{a_i})$$

IBM Model 1: generative story

- Input
  - an English sentence of length l
  - a length m
- For each French position i in 1..m
  - Pick an English source index j
  - Choose a translation

Alignment probabilities are UNIFORM

$$q(j \mid i, l, m) = \frac{1}{l+1}$$

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

Words are translated independently

IBM Model 1: Parameters

- t(f|e)
  - Word translation probability table
  - for all words in French & English vocab

f	e	p(f | e)
le	the	0.42
la	the	0.4
programme	the	0.001
a	has	0.78
…	…	…

IBM Model 1: generative story

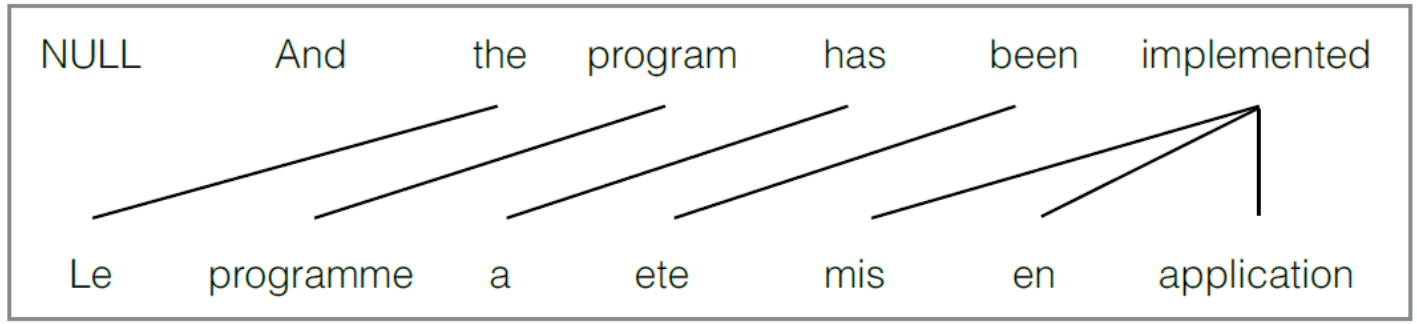
- Input
  - an English sentence of length l
  - a length m
- For each French position i in 1..m
  - Pick an English source index j
  - Choose a translation

$$q(j \mid i, l, m) = \frac{1}{l+1}$$

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

$$\boxed{p(f_1 \ldots f_m, a_1 \ldots a_m | e_1 \ldots e_l, m) = \prod_{i=1}^{m} q(a_i | i, l, m) t(f_i | e_{a_i})}$$

IBM Model 1: Example



- Alignment vector a = [2,3,4,5,6,6,6]
- P(f,a|e)?

Improving on IBM Model 1: IBM Model 2

- Input
  - an English sentence of length l
  - a length m

- For each French position i in $1..\mathrm{m}$
  - Pick an English source index j
  - Choose a translation

Remove assumption that q is uniform

$$q(j \mid i, l, m)$$

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

IBM Model 2: Parameters

- q(j|i,l,m)
  - now a table
  - not uniform as in IBM1

- How many parameters are there?

j	q(j | 1, 6, 7)
1	0.27
2	0.14
…	…
48	1E-75

2 Remaining Tasks

Inference

- Given
  - a sentence pair (e,f)
  - an alignment model with parameters t(f|e) and q(j|i,l,m)
- What is the most probable alignment a?

Parameter Estimation

- Given
  - training data (lots of sentence pairs)
  - a model definition
- how do we learn the parameters t(f|e) and q(j|i,l,m)?

Inference

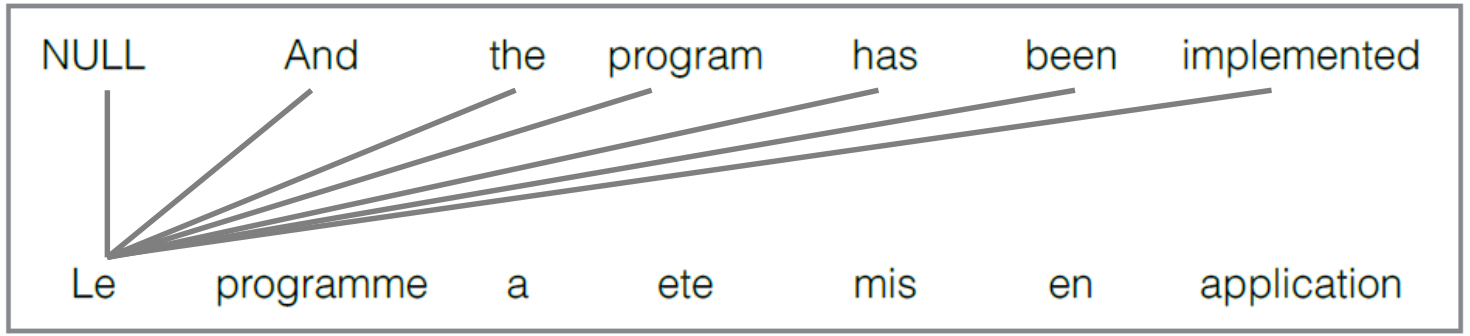
- Inputs
  - Model parameter tables for t and q
  - A sentence pair

NULL	And	the	program	has	been	implemented
Le	programme	a	ete	mis	en	application

- How do we find the alignment a that maximizes P(f,a|e)?
  - Hint: recall independence assumptions!

Inference

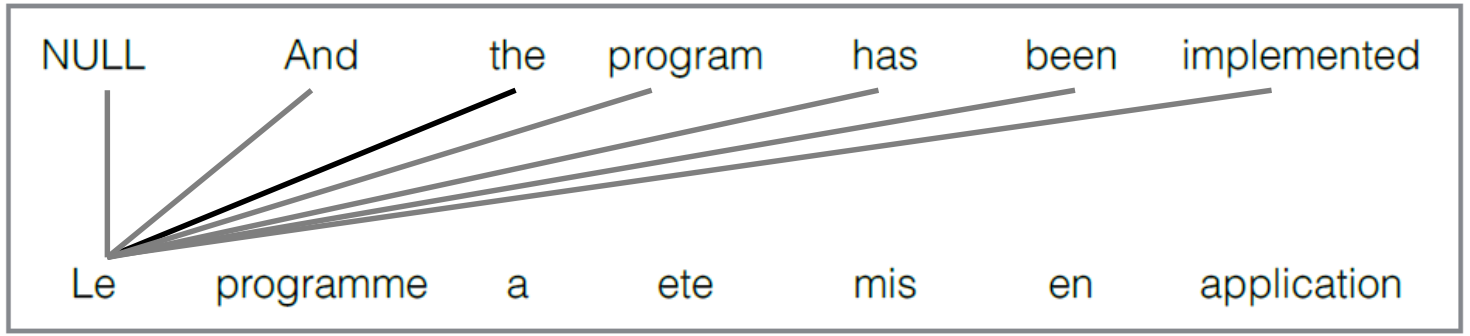
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

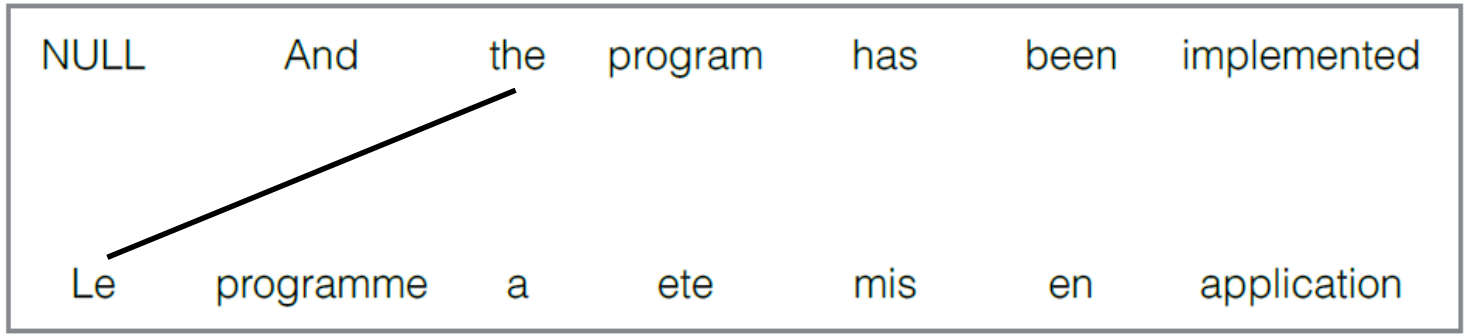
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

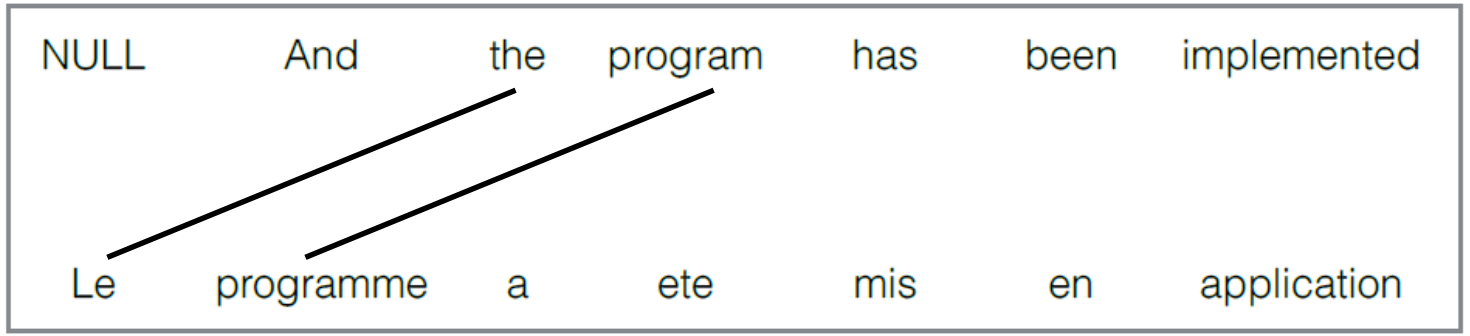
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

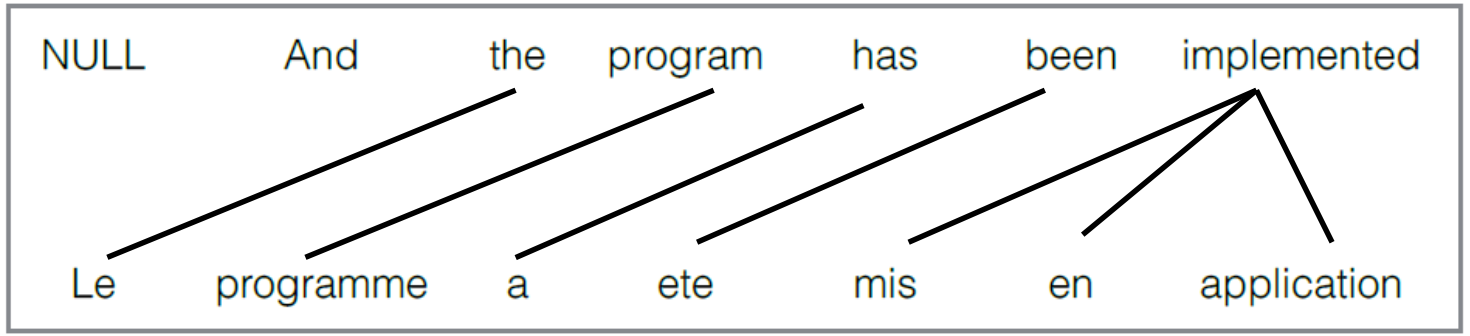
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

2 Remaining Tasks

Inference

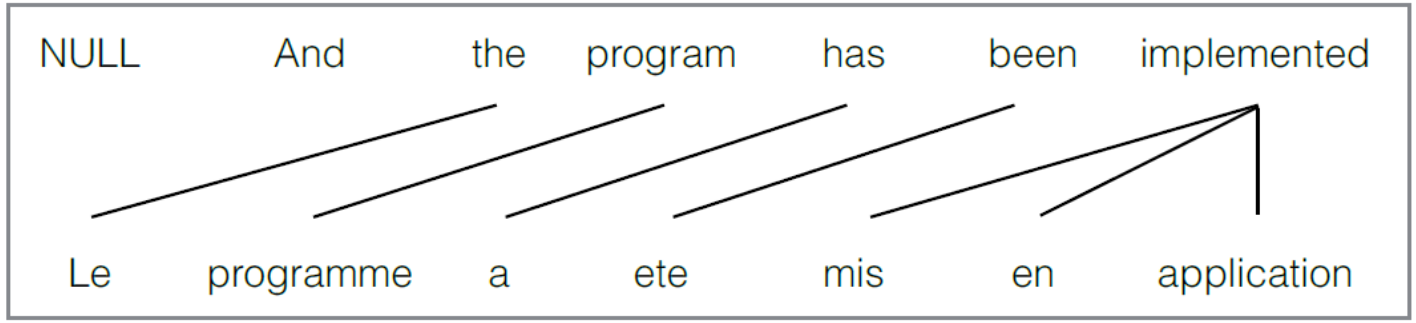
- Given
  - a sentence pair (e,f)
  - an alignment model with parameters t(f|e) and q(j|i,l,m)
- What is the most probable alignment a?

Parameter Estimation

- Given
  - training data (lots of sentence pairs)
  - a model definition
- how do we learn the parameters t(f|e) and q(j|i,l,m)?

Parameter Estimation (warm-up)

- Inputs
  - Model definition (t and q)
  - A corpus of sentence pairs, with word alignment



- How do we build tables for t and q?
  - Use counts, just like for n-gram models!

Parameter Estimation: hard EM

Algorithm 1 (hard EM)

```
initialize parameters t and q to something
repeat until convergence
    for every sentence
        for every target position j
            for every source position i
                if aligned(i, j)
                    count(f_j | e_i) += 1
                    count(e_i) += 1
                    count(j, i, l, m) += 1
                    count(i, l, m) += 1
    t(f | e) = count(f, e) / count(e)
    q(j | i, l, m) = count(j, i, l, m) / count(i, l, m)
```

Parameter Estimation

- Problem
  - Parallel corpus gives us **(e,f)** pairs only, **a is hidden**
- We know how to
  - estimate **t** and **q**, given **(e,a,f)**
  - compute **p(f,a|e)**, given t and q
- Solution: Expectation-Maximization algorithm (EM)
  - E-step: given hidden variable, estimate parameters
  - M-step: given parameters, update hidden variable

Parameter Estimation: EM

Algorithm 1 (soft EM)

initialize parameters t and q to something
repeat until convergence
    for every sentence
        for every target position j
            for every source position i
                $\text{count}(f_j, e_i) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
                $\text{count}(e_i) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
                $\text{count}(j, i, l, m) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
                $\text{count}(i, l, m) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
    $t(f \mid e) = \text{count}(f, e) / \text{count}(e)$
    $q(j \mid i, l, m) = \text{count}(j, i, l, m) / \text{count}(i, l, m)$

Use "Soft" values instead of binary counts

Parameter Estimation: soft EM

- Soft EM considers all possible alignment links
- Each alignment link now has a weight

$$P(a_i = j \mid e_i, f_j) = \frac{q(j \mid i, l, m) \cdot t(f_i \mid e_j)}{\sum_{j'=1}^{l} q(j' \mid i, l, m) \cdot t(f_i \mid e_{j'})}$$

EM for IBM Model 1

- Expectation (E)-step:
  - Compute expected counts for parameters (t) based on summing over hidden variable
- Maximization (M)-step:
  - Compute the maximum likelihood estimate of t from the expected counts

EM example: initialization

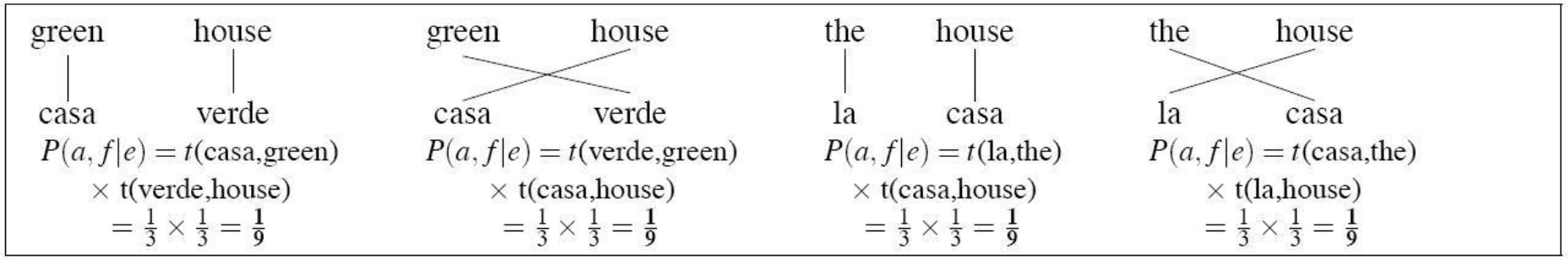
In this example:
Source language F = Spanish
Target language E = English

green house	the house
casa verde	la casa

$t(casa\|green) = \frac{1}{3}$	$t(verde\|green) = \frac{1}{3}$	$t(la\|green) = \frac{1}{3}$
$t(casa\|house) = \frac{1}{3}$	$t(verde\|house) = \frac{1}{3}$	$t(la\|house) = \frac{1}{3}$
$t(casa\|the) = \frac{1}{3}$	$t(verde\|the) = \frac{1}{3}$	$t(la\|the) = \frac{1}{3}$

EM example: E-step

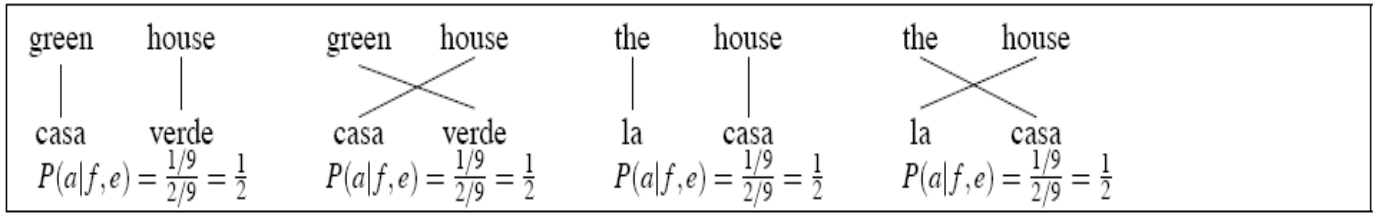
(a) compute probability of each alignment p(a,f|e)



Note: we're making simplification assumptions in this example

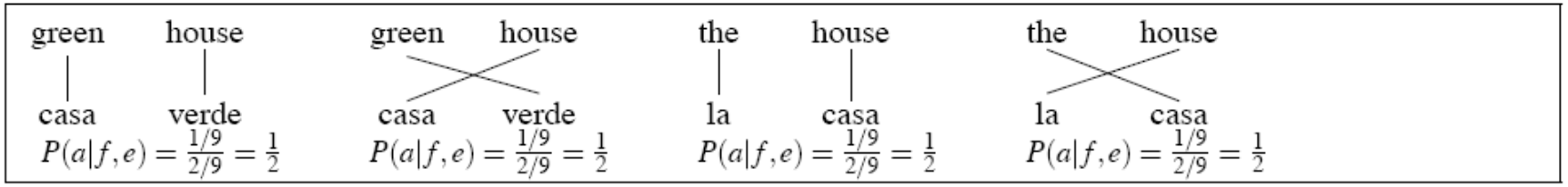
- No NULL word
- We only consider alignments were each French and English word is aligned to something
- We ignore q!

EM example: E-step
(b) normalize to get p(a|f,e)



EM example: E-step

(c) compute expected counts

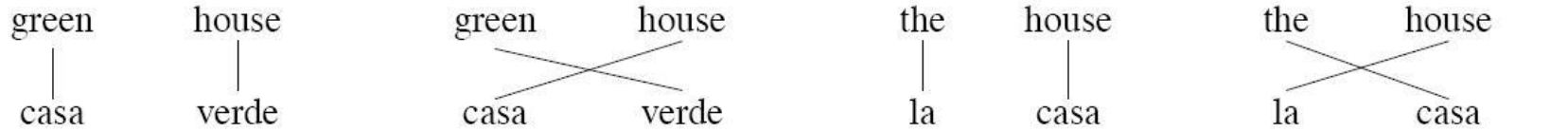


$\text{tcount}(\text{casa}\|\text{green}) = \frac{1}{2}$	$\text{tcount}(\text{verde}\|\text{green}) = \frac{1}{2}$	$\text{tcount}(\text{la}\|\text{green}) = 0$	$\text{total}(\text{green}) = 1$
$\text{tcount}(\text{casa}\|\text{house}) = \frac{1}{2} + \frac{1}{2}$	$\text{tcount}(\text{verde}\|\text{house}) = \frac{1}{2}$	$\text{tcount}(\text{la}\|\text{house}) = \frac{1}{2}$	$\text{total}(\text{house}) = 2$
$\text{tcount}(\text{casa}\|\text{the}) = \frac{1}{2}$	$\text{tcount}(\text{verde}\|\text{the}) = 0$	$\text{tcount}(\text{la}\|\text{the}) = \frac{1}{2}$	$\text{total}(\text{the}) = 1$

EM example: M-step
(d) normalize expected counts

$t(\text{casa}\|\text{green}) = \frac{1/2}{1} = \frac{1}{2}$	$t(\text{verde}\|\text{green}) = \frac{1/2}{1} = \frac{1}{2}$	$t(\text{la}\|\text{green}) = \frac{0}{1} = 0$
$t(\text{casa}\|\text{house}) = \frac{1}{2} = \frac{1}{2}$	$t(\text{verde}\|\text{house}) = \frac{1/2}{2} = \frac{1}{4}$	$t(\text{la}\|\text{house}) = \frac{1/2}{2} = \frac{1}{4}$
$t(\text{casa}\|\text{the}) = \frac{1/2}{1} = \frac{1}{2}$	$t(\text{verde}\|\text{the}) = \frac{0}{1} = 0$	$t(\text{la}\|\text{the}) = \frac{1/2}{1} = \frac{1}{2}$

EM example: next iteration



$P(a, f|e) = t(\text{casa,green}) \times t(\text{verde,house}) = \frac{1}{2} \times \frac{1}{4} = \mathbf{\frac{1}{8}}$

$P(a, f|e) = t(\text{verde,green}) \times t(\text{casa,house}) = \frac{1}{2} \times \frac{1}{2} = \mathbf{\frac{1}{4}}$

$P(a, f|e) = t(\text{la,the}) \times t(\text{casa,house}) = \frac{1}{2} \times \frac{1}{2} = \mathbf{\frac{1}{4}}$

$P(a, f|e) = t(\text{casa,the}) \times t(\text{la,house}) = \frac{1}{2} \times \frac{1}{4} = \mathbf{\frac{1}{8}}$

Parameter Estimation with EM

- EM guarantees that data likelihood does not decrease across iterations

$$\log \mathcal{L}(t, q \mid E, F) = \log \prod_{n=1}^{N} p(f^{(n)} \mid e^{(n)})$$

$$= \sum_{n=1}^{N} \log \sum_{a \in A} p(f^{(n)}, a \mid e^{(n)})$$

- EM can get stuck in a local optimum
  - Initialization matters

EM for IBM 1 in practice

- The previous example illustrates the EM algorithm

- But it is a little naïve
  - we had to enumerate all possible alignments
  - In practice, we don't need to sum overall all possible alignments explicitly for IBM1

  http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/ibm12.pdf

Word Alignment with IBM Models 1, 2

- Probabilistic models with **strong independence assumptions**
  - Results in linguistically naïve models
    - asymmetric, 1-to-many alignments
  - But allows efficient parameter estimation and inference

- Alignments are hidden variables
  - unlike words which are observed
  - require **unsupervised learning** (EM algorithm)