Approaches to Machine Translation: Rule-based, Statistical and Hybrid

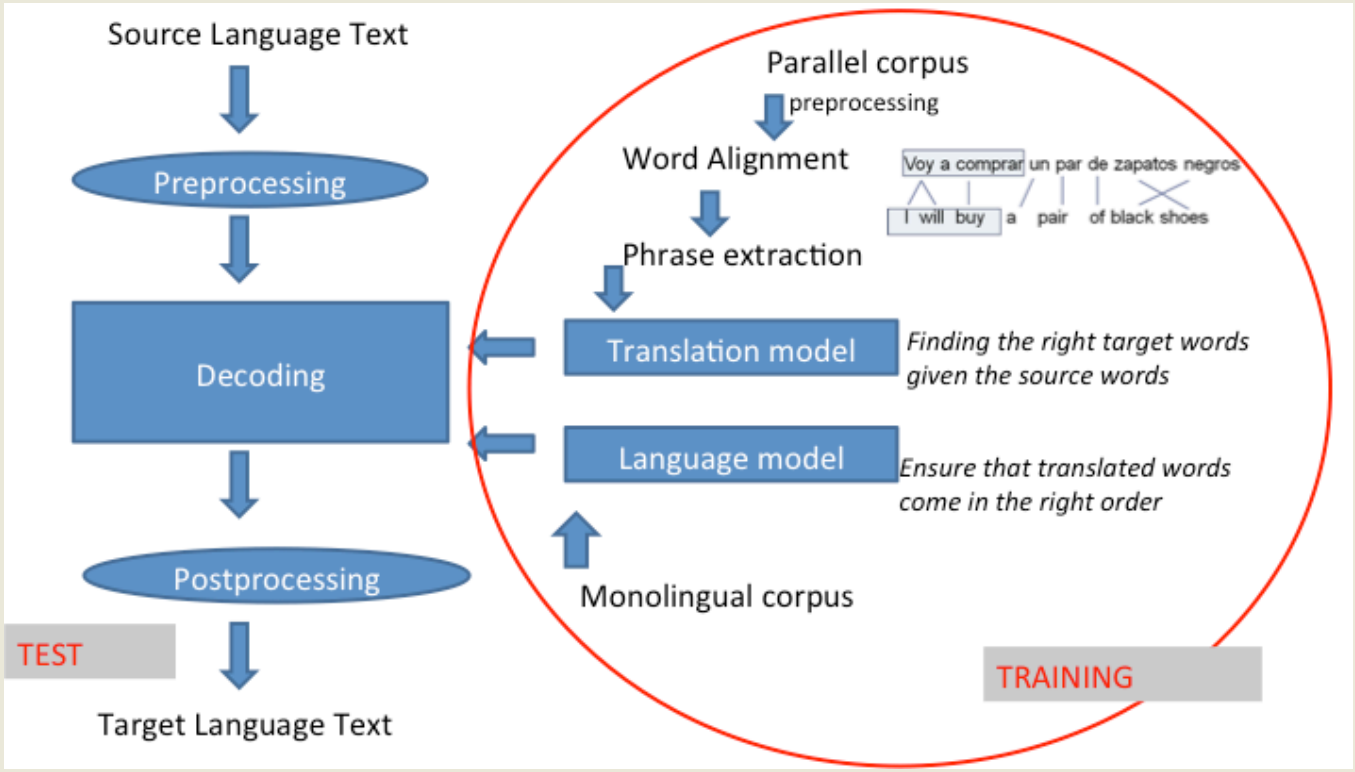
Alignment - IBM Model 1 and EM (II)

Lluís Formiga

July 11, 2014

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

A picture is worth a million equations



Concepts to be studied

- ~~Noisy Channel Model~~
- ~~Lexical translation~~
- ~~Word Alignment~~
- Expectation Maximization (EM) Algorithm
- IBM Models 1--5
  - IBM Model 1: lexical translation
  - IBM Model 2: alignment model
  - IBM Model 3: fertility
  - IBM Model 4: relative alignment model
  - IBM Model 5: deficiency
- HMM Models: dependent alignment model
- Problems of Word Alignment
- Quality of Word Alignment

IBM Model 1

Basic statistics:

- IBM Model 1 captures $p(f|e)$

Translations of *mesa*	$p(f\|e)$
table	0.3771
round	0.1476
panel	0.1344
round-table	0.0452
petitioners	0.0282
bureau	0.0229
officers	0.0190
Committee	0.0169
Round	0.0153
roundtable	0.0124

IBM Model 1

Definition

- Generative: break up translation process into smaller steps
  - IBM Model 1 only uses lexical translation
- Translation probability
  - for a foreign sentence $f = (f_1, \ldots, f_{l_f})$ of length l_f
  - to a target sentence $e = (e_1, \ldots, e_{l_e})$ of length l_e
  - with an alignment of each foreign word f_j to a translated word e_i according to the alignment function $a : i \rightarrow j$

$$P(e, a|f) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{i=1}^{l_e} t(e_i|f_{a(i)})$$

  - Parameter ϵ is a normalization constant

Example

the

e	t(e|f)
el	0.7
la	0.15
lo	0.075
un	0.05
este	0.025

train

e	t(e|f)
tren	0.8
locomotora	0.16
máquina	0.02
convoy	0.015
ferrocarril	0.005

goes

e	t(e|f)
va	0.8
anda	0.16
marcha	0.02
parte	0.015
funciona	0.005

fast

e	t(e|f)
rápido	0.4
rápida	0.4
veloz	0.1
presto	0.06
fugaz	0.04

$$
\begin{aligned}
\mathrm{p}(\mathrm{e}, \mathrm{a}|\mathrm{f}) &= \frac{\epsilon}{5^4} \times \mathrm{t}(\textbf{el}|\textbf{the}) \times \mathrm{t}(\textbf{tren}|\textbf{train}) \times \mathrm{t}(\textbf{va}|\textbf{goes}) \times \mathrm{t}(\textbf{rápido}|\textbf{fast}) \\
&= \frac{\epsilon}{5^4} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\
&= 0.00028672\epsilon
\end{aligned}
$$

Learning Lexical Translation Models

- We would like to estimate the lexical translation probabilities $t(e|f)$ from a parallel corpus
- ... but we do not have the alignments

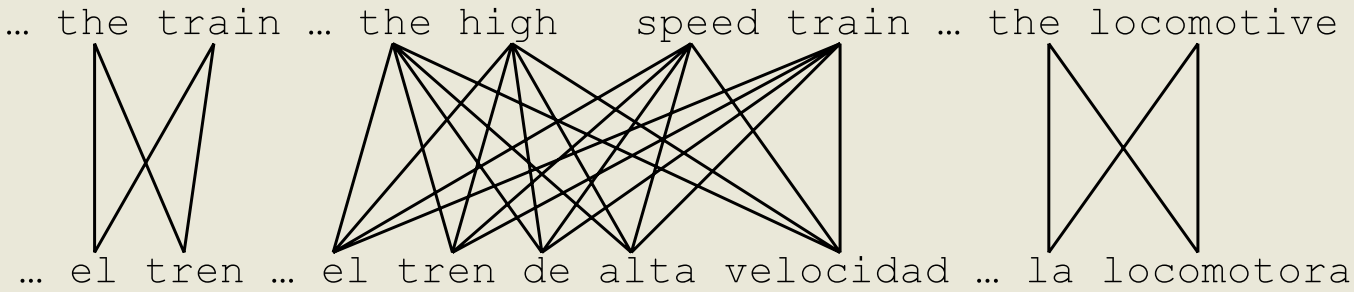
Chicken and egg problem

- if we had the alignments,
  $\rightarrow$ we could estimate the parameters of our generative model
- if we had the parameters,
  $\rightarrow$ we could estimate the alignments

EM Algorithm

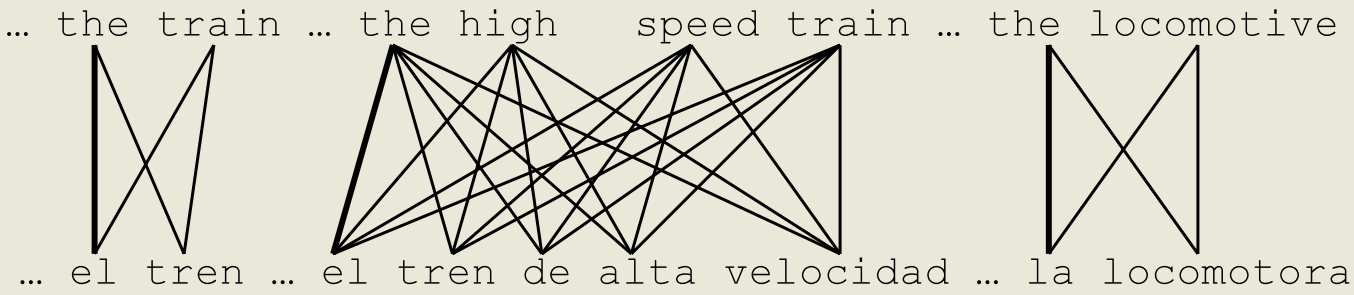
- Incomplete data
  - if we had complete data, would could estimate model
  - if we had model, we could fill in the gaps in the data
- Expectation Maximization (EM) in a nutshell
  1. initialize model parameters (e.g. uniform)
  2. assign probabilities to the missing data
  3. estimate model parameters from completed data
  4. iterate steps 2--3 until convergence

EM Algorithm



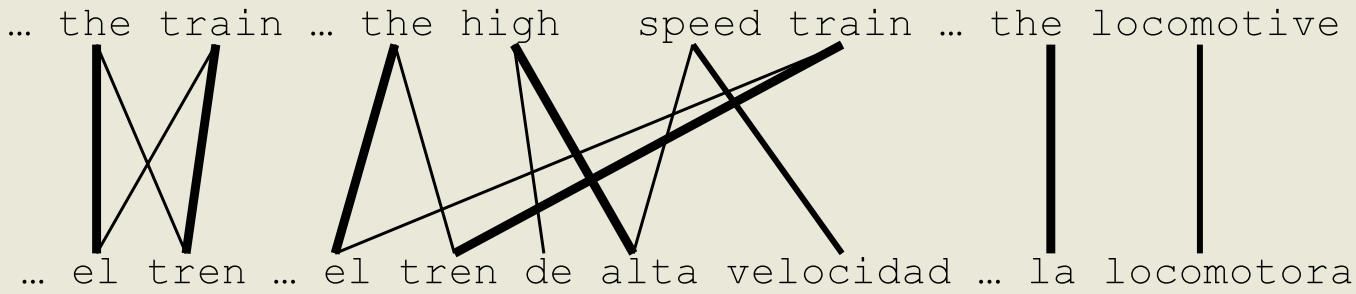
- Initial step: alignments are uniformly initialized

EM Algorithm



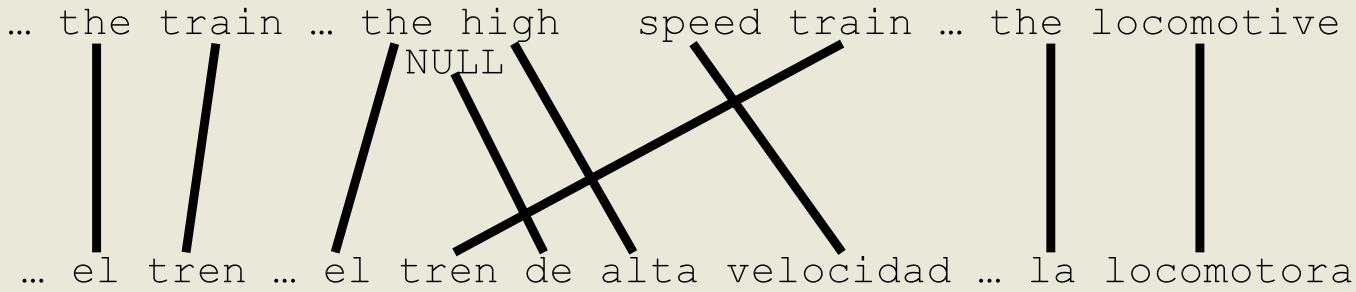
- Second iteration
- Model starts learning based on co-occurrence, e.g., la frequently paired with the

EM Algorithm



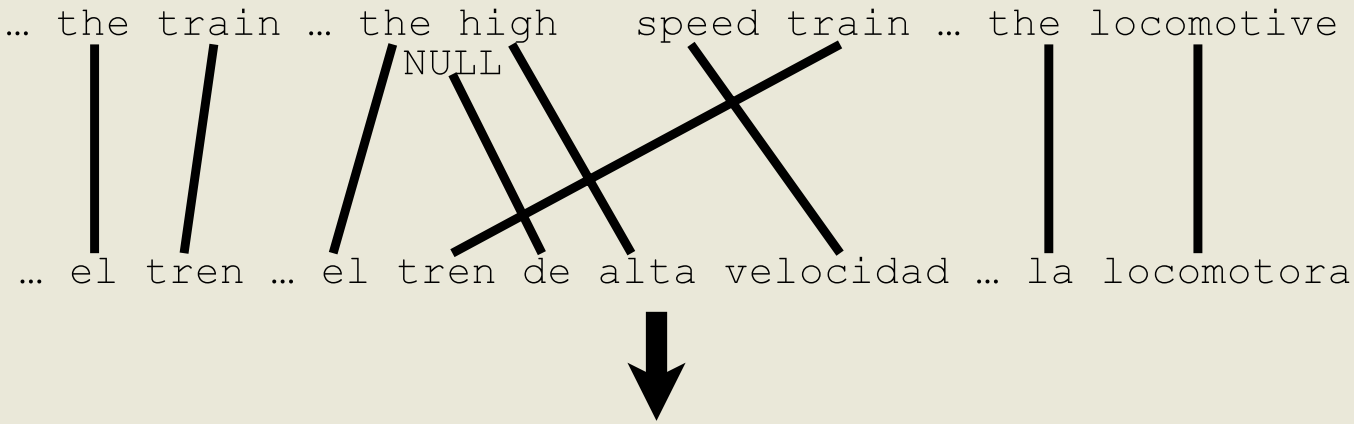
- Third iteration
- More evidence, e.g., between train and tren also co-occur frequently (pigeon hole principle)

EM Algorithm



- Convergence
- EM reveals inherent hidden structure

EM Algorithm



p(the|la)=0.546
p(the|el)=0.345
p(train|tren)=0.824
p(alta|high)=0.624
p(velocidad|speed)=0.734
p(locomotora|locomotive)=0.454
...

- Parameter estimation from the aligned corpus

IBM Model 1 and EM

EM: 2 step algorithm

Expectation-Step: Apply model to the data

- use the model to assign probabilities to observed values
- it assumes a hidden part of the model (alignments)

Maximization-Step: Estimate model from data

- Assigned values from E-step are a given fact
- Collect counts (MLE: weighted by probabilities)
- Build a new model from MLE (i.e. counts)

Repeat steps while not converged

IBM Model 1 and EM

We need to be able to compute:

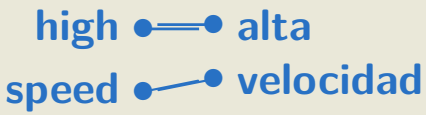
- Expectation-Step: probability of alignments
- Maximization-Step: count collection

IBM Model 1 and EM

Probabilities

$$\mathrm{p}(\textbf{alta}|\textbf{high}) = 0.7 \qquad \mathrm{p}(\textbf{velocidad}|\textbf{high}) = 0.05$$
$$\mathrm{p}(\textbf{alta}|\textbf{speed}) = 0.1 \qquad \mathrm{p}(\textbf{velocidad}|\textbf{speed}) = 0.8$$

Alignments



$\mathrm{p}(\mathrm{e}, \mathrm{a}|\mathrm{f}) = 0.56$
$\mathrm{p}(\mathrm{a}|\mathrm{e}, \mathrm{f}) = 0.824$

$\mathrm{p}(\mathrm{e}, \mathrm{a}|\mathrm{f}) = 0.035$
$\mathrm{p}(\mathrm{a}|\mathrm{e}, \mathrm{f}) = 0.052$

$\mathrm{p}(\mathrm{e}, \mathrm{a}|\mathrm{f}) = 0.08$
$\mathrm{p}(\mathrm{a}|\mathrm{e}, \mathrm{f}) = 0.118$

$\mathrm{p}(\mathrm{e}, \mathrm{a}|\mathrm{f}) = 0.005$
$\mathrm{p}(\mathrm{a}|\mathrm{e}, \mathrm{f}) = 0.007$

$\mathrm{p}(\mathrm{e}|\mathrm{f})$: $0.56 + 0.035 + 0.08 + 0.005 = 0.68$

Counts

$$\mathrm{c}(\textbf{alta}|\textbf{high}) = 0.824 + 0.052 \qquad \mathrm{c}(\textbf{velocidad}|\textbf{high}) = 0.052 + 0.007$$
$$\mathrm{c}(\textbf{alta}|\textbf{speed}) = 0.118 + 0.007 \qquad \mathrm{c}(\textbf{velocidad}|\textbf{speed}) = 0.824 + 0.118$$

IBM Model 1 and EM

Pseudocode (Soft Way)

Require: set of sentence pairs (e_s, f_s)
Ensure: translation prob. t(e|f) for all foreign words f and end word

{**initialize** t(f|e) **uniformly or from other training** }
repeat
  $\forall_{e_i \in e} \forall_{f_j \in f}$: $\text{count}(e_i|f_j) = 0$
  $\forall_{f_j \in f}$: $\text{total}(f_j) = 0$
  {**compute normalization** }
  for all sentence pairs (e_s, f_s) **do**
    for all words $f_j \in f_s$ **do**
      $\text{total}_s(f_j) = 0$
      for all words $e_i \in e_s$ **do**
        $\text{total}_s(f_j) \mathrel{+}= t(f_j|e_i)$
      end for
    end for
    {**collect counts** }
    for all words $f_j \in f_s$ **do**
      for all words $e_i \in e_s$ **do**
        $\text{count}(f_j, e_i) \mathrel{+}= \frac{t(f_j|e_i)}{\text{total}_s(f_j)}$
        $\text{total}(e_i) \mathrel{+}= \frac{t(f_j|e_i)}{\text{total}_s(f_j)}$
      end for
    end for
    {**estimate probabilities** }
    for all words $f_j \in f$ **do**
      for all words $e_i \in e$ **do**
        $t(f_j|e_i) = \frac{\text{count}(f_j, e_i)}{\text{total}(e_i)}$
      end for
    end for
until convergence

IBM Model 1 and EM Convergence

e	f	initial	1st it.	2nd it.	3rd it.	...	final
el	the	0.25	0.5	0.6364	0.7479	...	1
tren	the	0.25	0.25	0.1818	0.1208	...	0
coche	the	0.25	0.25	0.1818	0.1313	...	0
el	train	0.25	0.25	0.1818	0.1208	...	0
tren	train	0.25	0.5	0.6364	0.7479	...	1
un	train	0.25	0.25	0.1818	0.1313	...	0
tren	a	0.25	0.5	0.4286	0.3466	...	0
un	a	0.25	0.5	0.5714	0.6534	...	1
el	car	0.25	0.5	0.4286	0.3466	...	0
coche	car	0.25	0.5	0.5714	0.6534	...	1

IBM Model 1 and EM — EM algorithm

Why does it work?

- We are accumulating evidence (soft counts) for totally bogus alignments: all pairs of words that co- occur
  e.g., $t(\text{tren}|\text{train})$
- Words that co-occur frequently continually steal probability mass from pairs that co-occur less often

Properties

- The EM algorithm guarantees that data likelihood does not decrease across iterations

$$\log \mathcal{L}(t|E,F) = \log \prod_{n=1}^{N} \sum p(f^{(n)}|e^{(n)}) = \sum_{n=1}^{N} \log \sum_{a \in A} p(f^n, a|e^n)$$

- EM can get stuck in local optima: subprime peaks in the global likelihood function

Next session

- ~~Noisy Channel Model~~
- ~~Lexical translation~~
- ~~Word Alignment~~
- ~~Expectation Maximization (EM) Algorithm~~
- IBM Models 1--5
  - ~~IBM Model 1: lexical translation~~
  - IBM Model 2: alignment model
  - IBM Model 3: fertility
  - IBM Model 4: relative alignment model
  - IBM Model 5: deficiency
- HMM Models: dependent alignment model
- Problems of Word Alignment
- Quality of Word Alignment