Machine Translation

Lecture #17

Computational Linguistics
CMPSCI 591N, Spring 2006

Andrew McCallum

(including slides from Michael Collins, and Dan Klein)

The challenges of Machine Translation

Lexical Ambiguity

Example 1:

book the flight $\Rightarrow$ reservar

read the book $\Rightarrow$ libro

Example 2:

the box was in the pen

the pen was on the table

Example 3:

kill a man $\Rightarrow$ matar

kill a process $\Rightarrow$ acabar

Differing Word Orders

- English word order is *subject – verb – object*
- Japanese word order is *subject – object – verb*

English: IBM bought Lotus
Japanese: *IBM Lotus bought*

English: Sources said that IBM bought Lotus yesterday
Japanese: *Sources yesterday IBM Lotus bought that said*

Syntactic Structure is not Preserved Across Translations

The bottle floated into the cave

$\Downarrow$

La botella entro a la cuerva flotando
(the bottle entered the cave floating)

Syntactic Ambiguity Causes Problems

John hit the dog with the stick

$\Downarrow$

John golpeo el perro con el palo/que tenia el palo

Pronoun Resolution

The computer outputs the data; it is fast.

$\Downarrow$

La computadora imprime los datos; es rapida

The computer outputs the data; it is stored in ascii.

$\Downarrow$

La computadora imprime los datos; estan almacendos en ascii

Differing Treatments of Tense

From Dorr et. al 1998:

Mary went to Mexico. During her stay she learned Spanish.

Went $\Rightarrow$ iba (simple past/preterit)

Mary went to Mexico. When she returned she started to speak Spanish.

Went $\Rightarrow$ fue (ongoing past/imperfect)

The Best Translation May not be 1-1

(From Manning and Schuetze):

According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above average growth rates.

⇒

Quant aux eaux minerales et aux limonades, elles recontrent toujours plus d'adeptes. En effet notre sondage fait ressortir des ventes nettement superieures a celles de 1987, pour les boissons a base de cola notamment.

With regard to the mineral waters and the lemonades (soft drinks) they encounter still more users. Indeed our survey makes stand out the sales clearly superior to those in 1987 for cola-based drinks especially.

Machine Translation: Example

Atlanta, preso il killer del palazzo di Giustizia

ATLANTA - La grande paura che per 26 ore ha attanagliato Atlanta è finita: Brian Nichols, l'uomo che aveva ucciso tre persone a palazzo di Giustizia e che ha poi ucciso un agente di dogana, s'è consegnato alla polizia, dopo avere cercato rifugio nell'alloggio di una donna in un complesso d'appartamenti alla periferia della città. Per tutto il giorno, il centro della città, sede della Coca Cola e dei Giochi 1996, cuore di una popolosa area metropolitana, era rimasto paralizzato.

Atlanta, taken the killer of the palace of Justice

ATLANTA - The great fear that for 26 hours has gripped Atlanta is ended: Brian Nichols, the man who had killed three persons to palace of Justice and that a customs agent has then killed, s' is delivered to the police, after to have tried shelter in the lodging of one woman in a complex of apartments to the periphery of the city. For all the day, the center of the city, center of the Coke Strains and of Giochi 1996, heart of one popolosa metropolitan area, was remained paralyzed.

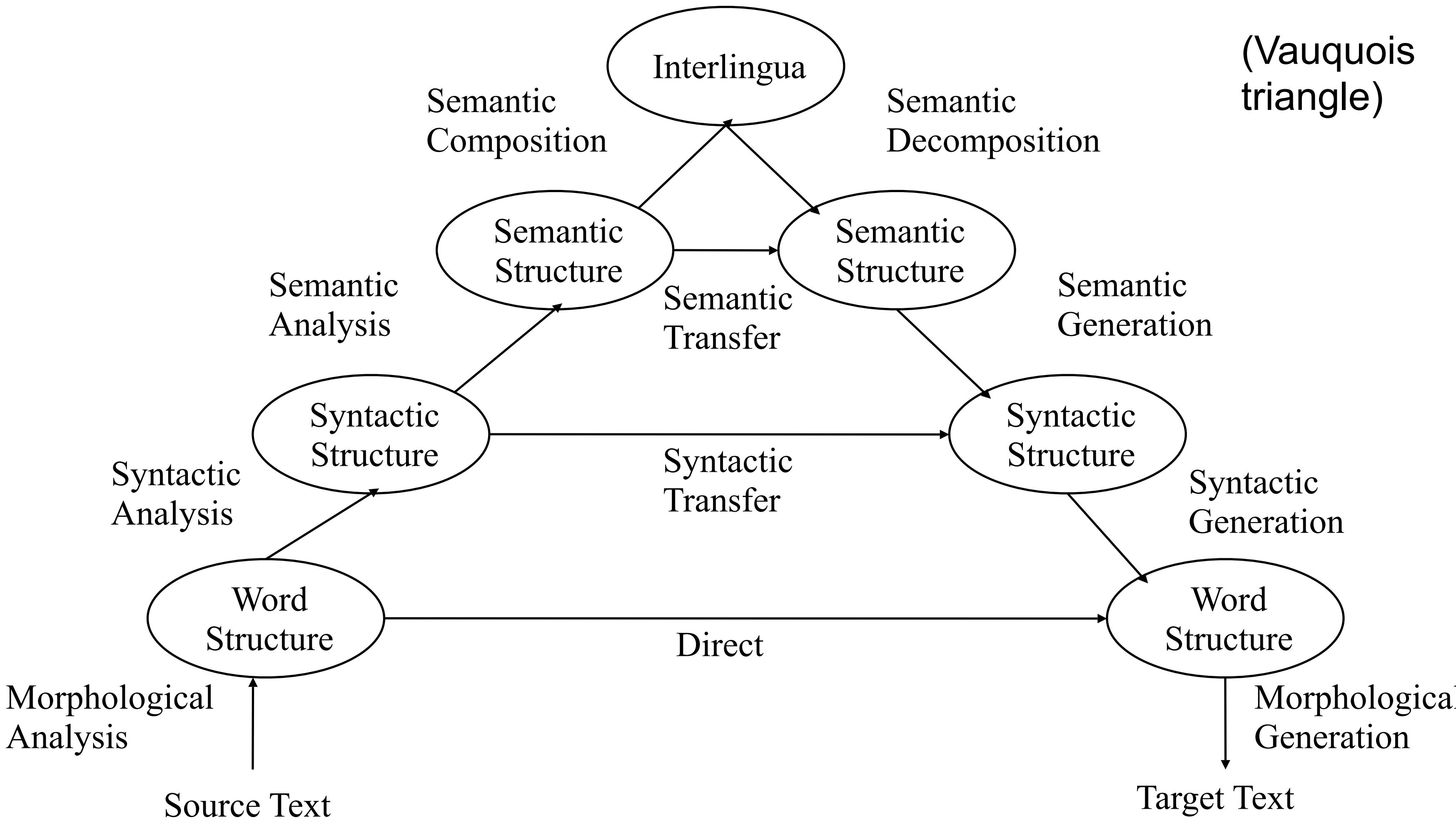
History

- 1950's: Intensive research activity in MT
- 1960's: Direct word-for-word replacement
- 1966 (ALPAC): NRC Report on MT
  - Conclusion: MT no longer worthy of serious scientific investigation.
- 1966-1975: `Recovery period'
- 1975-1985: Resurgence (Europe, Japan)
- 1985-present: Gradual Resurgence (US)

http://ourworld.compuserve.com/homepages/WJHutchins/MTS-93.htm

Levels of Transfer

(Vauquois triangle)



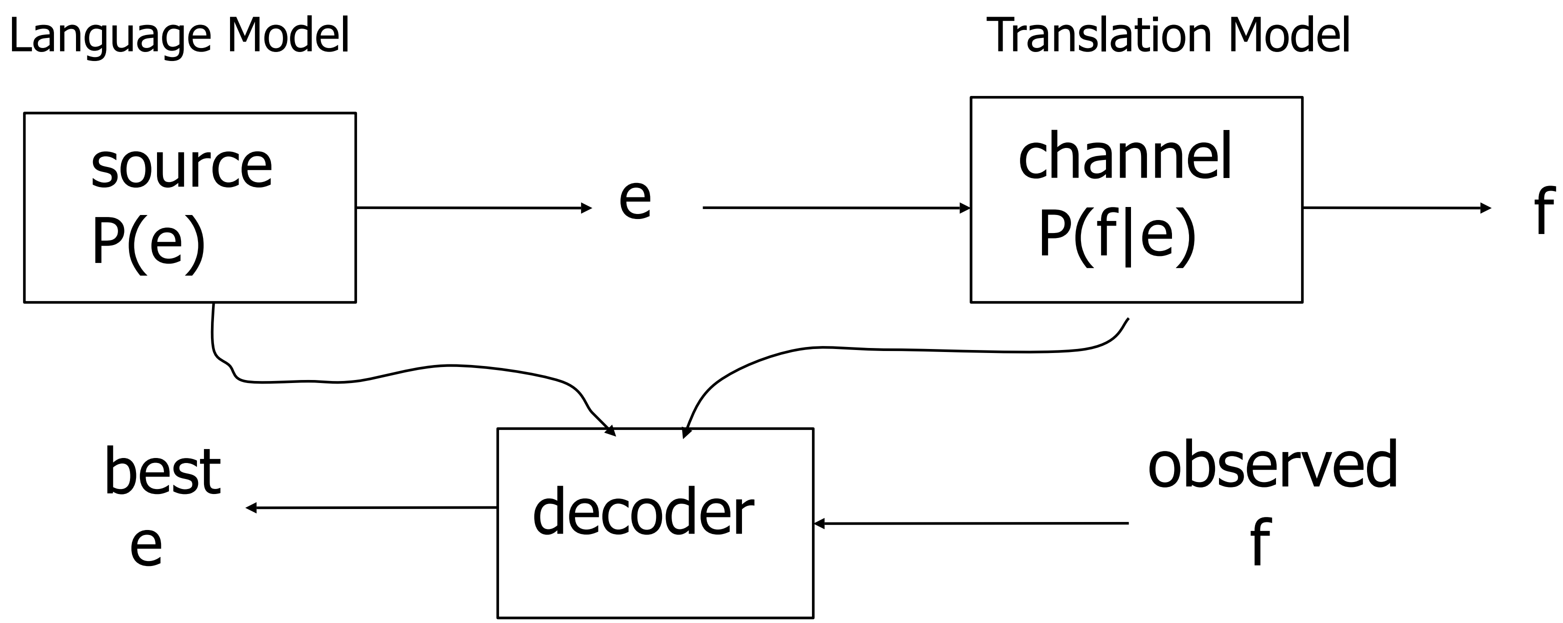
General Approaches

- Rule-based approaches
  - Expert system-like rewrite systems
  - Interlingua methods (analyze and generate)
  - Lexicons come from humans
  - Can be very fast, and can accumulate a lot of knowledge over time (e.g. Systran)
- Statistical approaches
  - Word-to-word translation
  - Phrase-based translation
  - Syntax-based translation (tree-to-tree, tree-to-string)
  - Trained on parallel corpora
  - Usually noisy-channel (at least in spirit)

The Coding View

- “One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: ‘This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.’ ”
  - Warren Weaver (1955:18, quoting a letter he wrote in 1947)

MT System Components



$$\underset{e}{\operatorname{argmax}}\, P(e|f) = \underset{e}{\operatorname{argmax}}\, P(f|e)P(e)$$

Why not simply P(e|f)?

More data for P(e).

Finds an English translation which is both fluent and semantically faithful to the French source

A Brief Introduction to Statistical MT

- Parallel corpora are available in several language pairs
- Basic idea: use a parallel corpus as a training set of translation examples
- Classic example: IBM work on French-English translation, using the Canadian Hansards. (1.7 million sentences of 30 words or less in length).

Example from Koehn and Knight tutorial

Translation from Spanish to English, candidate translations based on $P(Spanish \mid English)$ alone:

Que hambre tengo yo
$\rightarrow$
What hunger have $\quad P(S|E) = 0.000014$
Hungry I am so $\quad P(S|E) = 0.000001$
I am so hungry $\quad P(S|E) = 0.0000015$
Have i that hunger $\quad P(S|E) = 0.000020$
$\ldots$

With $P(Spanish \mid English) \times P(English)$:

Que hambre tengo yo

$\rightarrow$

What hunger have	$P(S\|E)P(E) = 0.000014 \times 0.000001$
Hungry I am so	$P(S\|E)P(E) = 0.000001 \times 0.0000014$
I am so hungry	$P(S\|E)P(E) = 0.0000015 \times 0.0001$
Have i that hunger	$P(S\|E)P(E) = 0.000020 \times 0.00000098$

$\ldots$

The Sentence Alignment Problem

- Might have 1003 sentences (in sequence) of English, 987 sentences (in sequence) of French: **but which English sentence(s) corresponds to which French sentence(s)?**

$$
\begin{array}{ll}
e_1 & f_1 \\
e_2 & f_2 \\
e_3 & f_3 \\
e_4 & f_4 \\
e_5 & f_5 \\
e_6 & f_6 \\
e_7 & f_7 \\
\ldots &
\end{array}
\quad \Rightarrow \quad
\begin{array}{ll}
e_1 & f_1 \\
e_2 & \\
\text{---------} & \\
e_3 & f_2 \\
\text{---------} & \\
e_4 & f_3 \\
\text{---------} & \\
e_5 & f_4 \\
 & f_5 \\
\text{---------} & \\
e_6 & f_6 \\
e_7 & f_7 \\
\text{---------} & \\
\ldots &
\end{array}
$$

- Might have 1-1 alignments, 1-2, 2-1, 2-2 etc.

The Sentence Alignment Problem

- Clearly needed before we can train a translation model
- Also useful for other multi-lingual problems
- Two broad classes of methods we'll cover:
  - Methods based on sentence lengths alone.
  - Methods based on lexical matches, or "cognates".

Sentence Length Methods

(Gale and Church, 1993):

- Method assumes paragraph alignment is known, sentence alignment is not known.
- Define:
  - l_e = length of English sentence, in characters
  - l_f = length of French sentence, in characters
- Assumption: given length l_e, length l_f has a gaussian/normal distribution with mean $c \times l_e$, and variance $s^2 \times l_e$ for some constants c and s.
- Result: we have a cost

$$Cost(l_e, l_f)$$

for any pairs of lengths l_e and l_f.

Each Possible Alignment Has a Cost

e_1 f_1
e_2
----
e_3 f_2
----
e_4 f_3
----
e_5 f_4
f_5
----
e_6 f_6
e_7 f_7

$\ldots$

In this case, if length of e_i is l_i, and length of f_i is m_i, total cost is

$$
\begin{aligned}
Cost = {} & Cost(l_1 + l_2, m_1) + Cost_{21} + \\
& Cost(l_3, m_2) + Cost_{11} + \\
& Cost(l_4, m_3) + Cost_{11} + \\
& Cost(l_4, m_4 + m_5) + Cost_{12} + \\
& Cost(l_6 + l_7, m_6 + m_7) + Cost_{22}
\end{aligned}
$$

where $Cost_{ij}$ terms correspond to costs for 1-1, 1-2, 2-1 and 2-2 alignments.

- Dynamic programming can be used to search for the lowest cost alignment

Methods Based on Cognates

- Intuition: related words in different languages often have similar spellings e.g., government and gouvernement

- Cognate matches can "anchor" sentence-sentence correspondences

- A method from (Church 1993): track all 4-grams of characters which are identical in the two texts.

- A method from (Melamed 1993), measures similarity of words A and B:

$$LCSR(A, B) = \frac{length(LCS(A, B))}{max(length(A), length(B))}$$

  where LCS is the longest common subsequence (not necessarily contiguous) in A and B. e.g.,

$$LCSR(\text{government,gouvernement}) = \frac{10}{13}$$

Today

- The components of a simple MT system
  - You already know about the LM
  - Word-alignment based TMs
    - IBM models 1 and 2, HMM model
  - A simple decoder

- Not today
  - More complex word-level and phrase-level TMs
  - Tree-to-tree and tree-to-string TMs
  - More sophisticated decoders

A Word-Level TM?

- What might a model of P(f|e) look like?

$e = e_1 \ldots e_I$	And_1	the_2	program_3	has_4	been_5	implemented_6	
$f = f_1 \ldots f_J$	Le_1	programme_2	a_3	été_4	mis_5	en_6	application_7

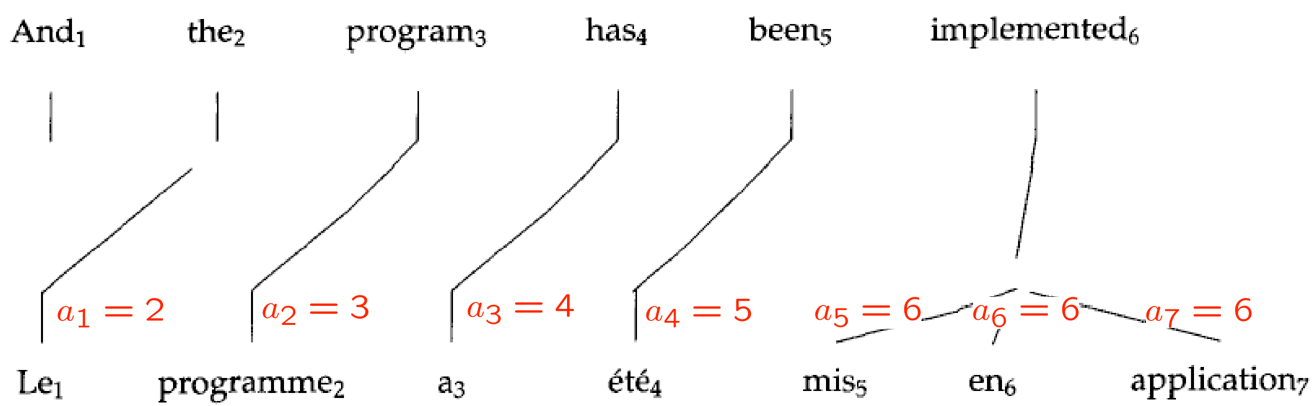
$$P(f|e) = \prod_j \underbrace{P(f_j|e_1 \ldots e_I)}_{\text{How to estimate this?}}$$

What can go wrong here?

IBM Model 1 (Brown 93)

- Alignments: a hidden vector called an *alignment* specifies which English source is responsible for each French target word.

$$a = a_1 \dots a_J$$

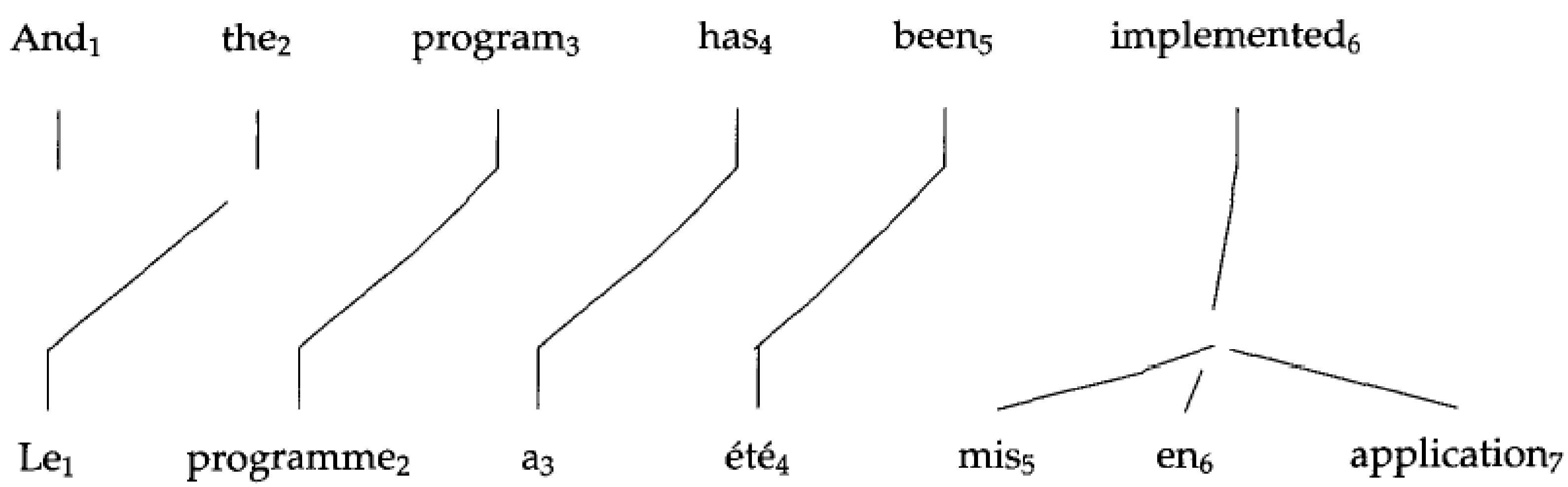


$$P(f, a|e) = \prod_j P(a_j = i) P(f_j|e_i)$$

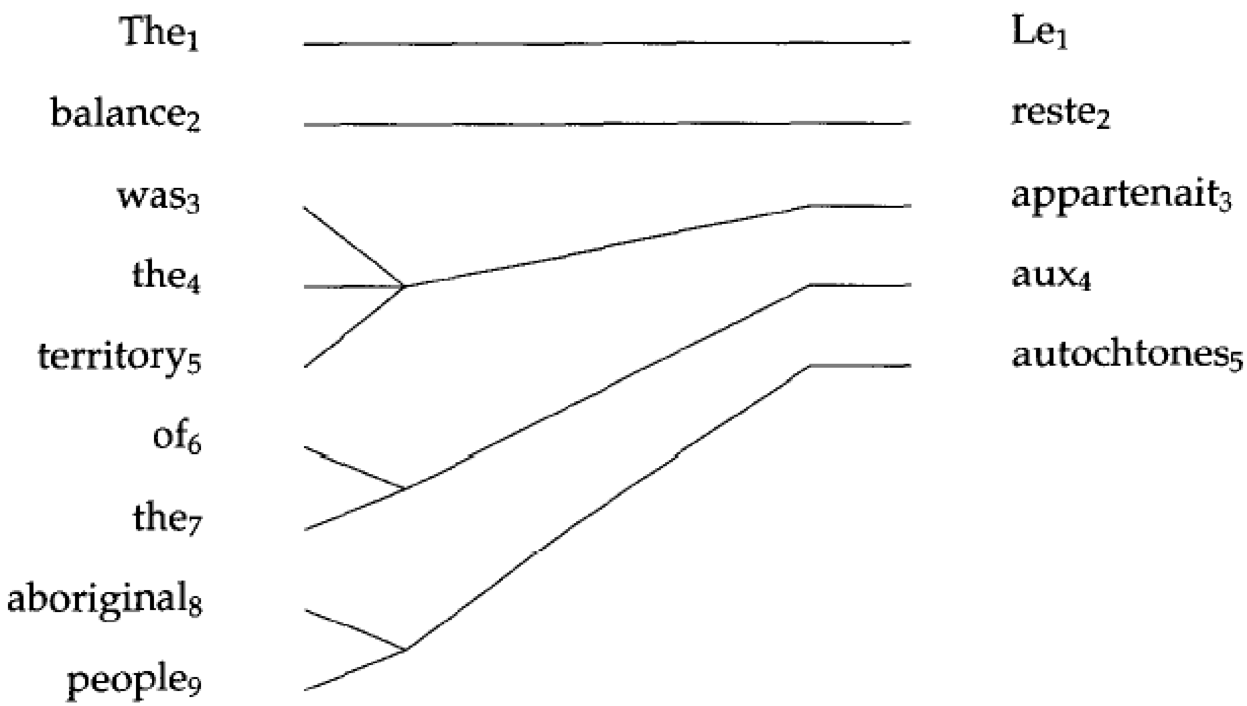
$$= \prod_j \frac{1}{I+1} P(f_j|e_i)$$

$$P(f|e) = \sum_a P(f, a|e)$$

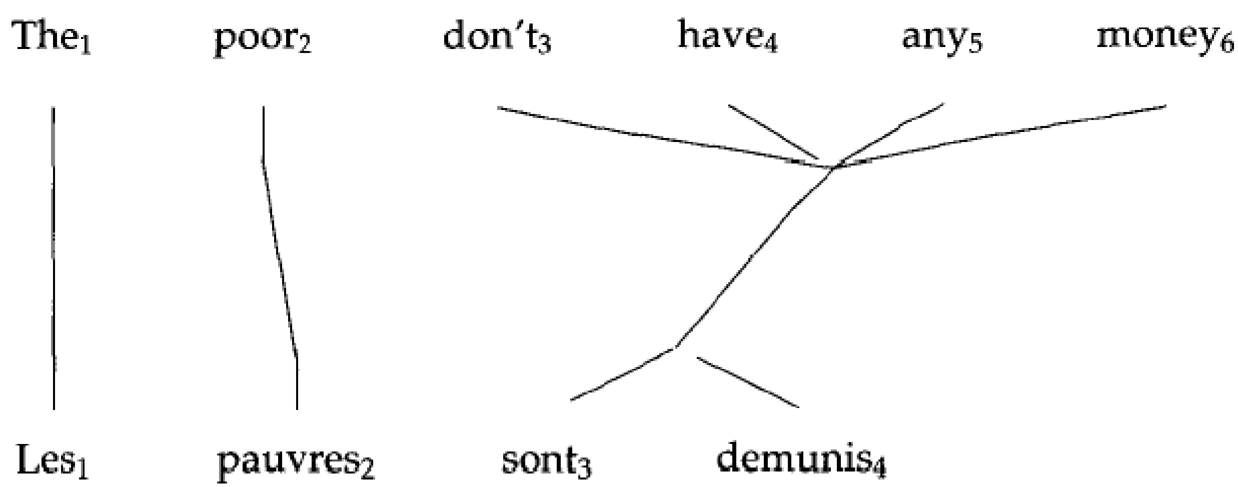
1-to-Many Alignments



Many-to-1 Alignments



Many-to-Many Alignments



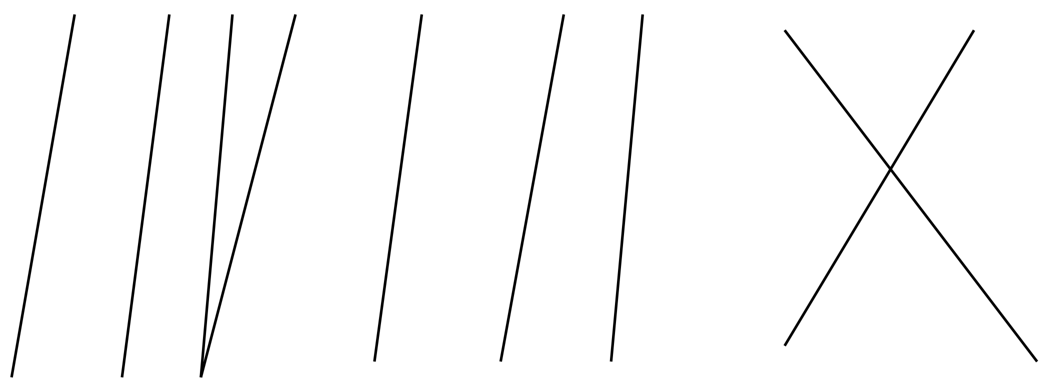
Monotonic Translation

Japan shaken by two new quakes

Le Japon secoué par deux nouveaux séismes

Local Order Change

Japan is at the junction of four tectonic plates



Le Japon est au confluent de quatre plaques tectoniques

IBM Model 2

- Alignments tend to the diagonal (broadly at least)

$$P(f, a|e) = \prod_j P(a_j = i|j, I, J) P(f_j|e_i)$$

$$P(i - j\frac{I}{J})$$

$$\frac{1}{Z} e^{-\alpha(i - j\frac{I}{J})}$$

- Other schemes for biasing alignments towards the diagonal:
  - Relative alignment
  - Asymmetric distances
  - Learning a multinomial over distances

IBM Model 2 - Alternative

- Model $P(a_j = i | j, I, J)$ as a simple dense table.

$$P(f, a|e) = \prod_j P(a_j = i|j, I, J)P(f_j|e_i)$$

- In other words, a simple multinomial over i for each j, I, J
  - e.g. D(i=2 | j=1, I=6, J=7)

How to learn these parameters from pairs of sentences?

EM for Models 1/2

- Model 1 Parameters:
  Translation probabilities (word pairs) $P(f_j|e_i)$
  Distortion parameters (1 only) $P(a_j = i|j, I, J)$
- Start with $P(f_j|e_i)$ uniform, including $P(f_j|null)$
- For each sentence:
  - For each French position j
    - Calculate posterior over English positions

$$P(a_j = i|f, e) = \frac{P(a_j = i|j, I, J)P(f_j|e_i)}{\sum_{i'} P(a_j = i'|j, I, J)P(f_j|e'_i)}$$

    - (or just use best single alignment)
    - Increment count of word f_j with word e_i by these amounts
    - Also re-estimate distortion probabilities for model 2
- Iterate until convergence

Notation switch:

$l = I$ length of English document
$m = J$ length of French document

IBM Model 2

- Only difference: we now introduce **alignment** or **distortion** parameters

$$\mathbf{D}(i \mid j, l, m) \quad = \quad \text{Probability that } j\text{'th French word is connected to } i\text{'th English word, given sentence lengths of } \mathbf{e} \text{ and } \mathbf{f} \text{ are } l \text{ and } m \text{ respectively}$$

- Define

$$P(\mathbf{a} = \{a_1, \ldots a_m\} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)$$

- Gives

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)\mathbf{T}(f_j \mid e_{a_j})$$

- Note: Model 1 is a special case of Model 2, where $\mathbf{D}(i \mid j, l, m) = \frac{1}{l+1}$ for all i, j.

An Example

$$
\begin{aligned}
l &= 6 \\
m &= 7 \\
\mathbf{e} &= \text{And the program has been implemented} \\
\mathbf{f} &= \text{Le programme a ete mis en application} \\
\mathbf{a} &= \{2, 3, 4, 5, 6, 6, 6\}
\end{aligned}
$$

$$
\begin{aligned}
P(\mathbf{a} \mid \mathbf{e}, l = 6, m = 7) = \; & \mathbf{D}(i = 2 \mid j = 1, l = 6, m = 7) \times \\
& \mathbf{D}(i = 3 \mid j = 2, l = 6, m = 7) \times \\
& \mathbf{D}(i = 4 \mid j = 3, l = 6, m = 7) \times \\
& \mathbf{D}(i = 5 \mid j = 4, l = 6, m = 7) \times \\
& \mathbf{D}(i = 6 \mid j = 5, l = 6, m = 7) \times \\
& \mathbf{D}(i = 6 \mid j = 6, l = 6, m = 7) \times \\
& \mathbf{D}(i = 6 \mid j = 7, l = 6, m = 7)
\end{aligned}
$$

$$
\begin{aligned}
P(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) \quad = \quad & \mathbf{T}(Le \mid the) \times \\
& \mathbf{T}(programme \mid program) \times \\
& \mathbf{T}(a \mid has) \times \\
& \mathbf{T}(ete \mid been) \times \\
& \mathbf{T}(mis \mid implemented) \times \\
& \mathbf{T}(en \mid implemented) \times \\
& \mathbf{T}(application \mid implemented)
\end{aligned}
$$

IBM Model 2: The Generative Process

To generate a French string $\mathbf{f}$ from an English string $\mathbf{e}$:

- **Step 1:** Pick the length of $\mathbf{f}$ (all lengths equally probable, probability C)
- **Step 2:** Pick an alignment $\mathbf{a} = \{a_1, a_2 \ldots a_m\}$ with probability

$$\prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)$$

- **Step 3:** Pick the French words with probability

$$P(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) = \prod_{j=1}^{m} \mathbf{T}(f_j \mid e_{a_j})$$

The final result:

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = P(\mathbf{a} \mid \mathbf{e}) P(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) = C \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m) \mathbf{T}(f_j \mid e_{a_j})$$

EM Training of Alignment and Translation Parameters

A Hidden Variable Problem

- **We have:**

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = C \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m) \mathbf{T}(f_j \mid e_{a_j})$$

- **And:**

$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a} \in \mathcal{A}} C \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m) \mathbf{T}(f_j \mid e_{a_j})$$

where $\mathcal{A}$ is the set of all possible alignments.

A Hidden Variable Problem

- Training data is a set of $(\mathbf{f}_k, \mathbf{e}_k)$ pairs, likelihood is

$$\sum_k \log P(\mathbf{f}_k \mid \mathbf{e}_k) = \sum_k \log \sum_{\mathbf{a} \in \mathcal{A}} P(\mathbf{a} \mid \mathbf{e}_k) P(\mathbf{f}_k \mid \mathbf{a}, \mathbf{e}_k)$$

  where $\mathcal{A}$ is the set of all possible alignments.

- We need to maximize this function w.r.t. the translation parameters, and the alignment probabilities

- EM can be used for this problem: initialize parameters randomly, and at each iteration choose

$$\Theta_t = \text{argmax}_\Theta \sum_i \sum_{\mathbf{a} \in \mathcal{A}} P(\mathbf{a} \mid \mathbf{e}_k, \mathbf{f}_k, \Theta^{t-1}) \log P(\mathbf{f}_k, \mathbf{a} \mid \mathbf{e}_k, \Theta)$$

  where Θ^t are the parameter values at the t'th iteration.

Models 1 and 2 Have a Simple Structure

- We have $\mathbf{f} = \{f_1 \ldots f_m\}$, $\mathbf{a} = \{a_1 \ldots a_m\}$, and

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} P(a_j, f_j \mid \mathbf{e}, l, m)$$

where

$$P(a_j, f_j \mid \mathbf{e}, l, m) = \mathbf{D}(a_j \mid j, l, m)\mathbf{T}(f_j \mid e_{a_j})$$

- **We can think of the m (f_j, a_j) pairs as being generated independently**

A Crucial Step in the EM Algorithm

- Say we have the following $(\mathbf{e}, \mathbf{f})$ pair:

$$\mathbf{e} = \text{And the program has been implemented}$$

$$\mathbf{f} = \text{Le programme a ete mis en application}$$

- Given that $\mathbf{f}$ was generated according to Model 2, what is the probability that $a_1 = 2$? **Formally:**

$$Prob(a_1 = 2 \mid \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}: a_1 = 2} P(\mathbf{a} \mid \mathbf{f}, \mathbf{e}, l, m)$$

The Answer

$$\begin{aligned} Prob(a_1 = 2 \mid \mathbf{f}, \mathbf{e}) &= \sum_{\mathbf{a}: a_1 = 2} P(\mathbf{a} \mid \mathbf{f}, \mathbf{e}, l, m) \\ &= \frac{\mathbf{D}(a_1 = 2 \mid j = 1, l = 6, m = 7)\mathbf{T}(le \mid the)}{\sum_{i=0}^{l} \mathbf{D}(a_1 = i \mid j = 1, l = 6, m = 7)\mathbf{T}(le \mid e_i)} \end{aligned}$$

Follows directly because the (a_j, f_j) **pairs are independent**:

$$\begin{aligned} P(a_1 = 2 \mid \mathbf{f}, \mathbf{e}, l, m) &= \frac{P(a_1 = 2, f_1 = Le \mid f_2 \dots f_m, \mathbf{e}, l, m)}{P(f_1 = Le \mid f_2 \dots f_m, \mathbf{e}, l, m)} \quad (1) \\ &= \frac{P(a_1 = 2, f_1 = Le \mid \mathbf{e}, l, m)}{P(f_1 = Le \mid \mathbf{e}, l, m)} \quad (2) \\ &= \frac{P(a_1 = 2, f_1 = Le \mid \mathbf{e}, l, m)}{\sum_i P(a_1 = i, f_1 = Le \mid \mathbf{e}, l, m)} \end{aligned}$$

where (2) follows from (1) because $P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} P(a_j, f_j \mid \mathbf{e}, l, m)$

A General Result

$$
\begin{aligned}
Prob(a_j = i \mid \mathbf{f}, \mathbf{e}) &= \sum_{\mathbf{a}: a_j = i} P(\mathbf{a} \mid \mathbf{f}, \mathbf{e}, l, m) \\
&= \frac{\mathbf{D}(a_j = i \mid j, l, m)\mathbf{T}(f_j \mid e_i)}{\sum_{i'=0}^{l} \mathbf{D}(a_j = i' \mid j, l, m)\mathbf{T}(f_j \mid e_{i'})}
\end{aligned}
$$

Alignment Probabilities have a Simple Solution!

- e.g., Say we have $l = 6$, $m = 7$,

$$\mathbf{e} = \text{And the program has been implemented}$$
$$\mathbf{f} = \text{Le programme a ete mis en application}$$

- Probability of “mis” being connected to “the”:

$$P(a_5 = 2 \mid \mathbf{f}, \mathbf{e}) = \frac{\mathbf{D}(a_5 = 2 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid the)}{Z}$$

where

$$\begin{aligned} Z = \quad & \mathbf{D}(a_5 = 0 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid NULL) \\ + \quad & \mathbf{D}(a_5 = 1 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid And) \\ + \quad & \mathbf{D}(a_5 = 2 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid the) \\ + \quad & \mathbf{D}(a_5 = 3 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid program) \\ + \quad & \ldots \end{aligned}$$

The EM Algorithm for Model 2

- Define

$\mathbf{e}[k]$	for $k = 1 \ldots n$ is the k'th English sentence
$\mathbf{f}[k]$	for $k = 1 \ldots n$ is the k'th French sentence
$l[k]$	is the length of $\mathbf{e}[k]$
$m[k]$	is the length of $\mathbf{f}[k]$
$\mathbf{e}[k, i]$	is the i'th word in $\mathbf{e}[k]$
$\mathbf{f}[k, j]$	is the j'th word in $\mathbf{f}[k]$

- Current parameters Θ^{t-1} are

$$\mathbf{T}(f \mid e) \qquad \text{for all } f \in \mathcal{F}, e \in \mathcal{E}$$

$$\mathbf{D}(i \mid j, l, m)$$

- We'll see how the EM algorithm re-estimates the $\mathbf{T}$ and $\mathbf{D}$ parameters

Step 1: Calculate the Alignment Probabilities

- Calculate an array of alignment probabilities (for $(k = 1 \dots n)$, $(j = 1 \dots m[k])$, $(i = 0 \dots l[k])$):

$$\begin{aligned} a[i,j,k] &= P(a_j = i \mid \mathbf{e}[k], \mathbf{f}[k], \Theta^{t-1}) \\ &= \frac{\mathbf{D}(a_j = i \mid j, l, m)\mathbf{T}(f_j \mid e_i)}{\sum_{i'=0}^{l} \mathbf{D}(a_j = i' \mid j, l, m)\mathbf{T}(f_j \mid e_{i'})} \end{aligned}$$

where $e_i = \mathbf{e}[k,i]$, $f_j = \mathbf{f}[k,j]$, and $l = l[k], m = m[k]$

i.e., the probability of $\mathbf{f}[k,j]$ being aligned to $\mathbf{e}[k,i]$.

Step 2: Calculating the Expected Counts

- Calculate the translation counts

$$tcount(e, f) = \sum_{\substack{i,j,k: \\ \mathbf{e}[k,i]=e, \\ \mathbf{f}[k,j]=f}} a[i, j, k]$$

- $tcount(e, f)$ is expected number of times that e is aligned with f in the corpus

Step 2: Calculating the Expected Counts

- Calculate the source counts

$$scount(e) = \sum_{\substack{i,k: \\ \mathbf{e}[k,i]=e}} \sum_{j=1}^{m[k]} a[i,j,k]$$

- $scount(e)$ is expected number of times that e is aligned with any French word in the corpus

Step 2: Calculating the Expected Counts

- Calculate the alignment counts

$$acount(i,j,l,m) = \sum_{\substack{k: \\ l[k]=l, m[k]=m}} a[i,j,k]$$

$$acount(j,l,m) = |\{k : l[k] = l, m[k] = m\}|$$

- Here, $acount(i,j,l,m)$ is expected number of times that e_i is aligned to f_j in English/French sentences of lengths l and m respectively

- $acount(j,l,m)$ is number of times that we have sentences $\mathbf{e}$ and $\mathbf{f}$ of lengths l and m respectively

Step 3: Re-estimating the Parameters

- New translation probabilities are then defined as

$$P(f \mid e) = \frac{tcount(e, f)}{scount(e)}$$

- New alignment probabilities are defined as

$$P(a_j = i \mid j, l, m) = \frac{acount(i, j, l, m)}{acount(j, l, m)}$$

This defines the mapping from Θ^{t-1} to Θ^t

A Summary of the EM Procedure

- Start with parameters Θ^{t-1} as

$$\begin{array}{rl} \mathbf{T}(f \mid e) & \text{for all } f \in \mathcal{F}, e \in \mathcal{E} \\ \mathbf{D}(i \mid j, l, m) & \end{array}$$

- Calculate **alignment probabilities** under current parameters

$$a[i,j,k] = \frac{\mathbf{D}(a_j = i \mid j, l, m)\mathbf{T}(f_j \mid e_i)}{\sum_{i'=0}^{l} \mathbf{D}(a_j = i' \mid j, l, m)\mathbf{T}(f_j \mid e_{i'})}$$

- Calculate **expected counts** $tcount(e, f)$, $scount(e)$, $acount(i, j, l, m)$, and $acount(j, l, m)$ from the alignment probabilities

- Re-estimate $\mathbf{T}(f \mid e)$ and $\mathbf{D}(i \mid j, l, m)$ from the expected counts

Some examples of training

An Example of Training Models 1 and 2

Example will use following translations:

e[1]	=	the	dog
f[1]	=	le	chien
e[2]	=	the	cat
f[2]	=	le	chat
e[3]	=	the	bus
f[3]	=	l'	autobus

NB: I won't use a NULL word e_0

Initial (random) parameters:

e	f	$\mathbf{T}(f \mid e)$
the	le	0.23
the	chien	0.2
the	chat	0.11
the	l'	0.25
the	autobus	0.21
dog	le	0.2
dog	chien	0.16
dog	chat	0.33
dog	l'	0.12
dog	autobus	0.18
cat	le	0.26
cat	chien	0.28
cat	chat	0.19
cat	l'	0.24
cat	autobus	0.03
bus	le	0.22
bus	chien	0.05
bus	chat	0.26
bus	l'	0.19
bus	autobus	0.27

Alignment probabilities:

i	j	k	a(i,j,k)
1	1	0	0.526423237959726
2	1	0	0.473576762040274
1	2	0	0.552517995605817
2	2	0	0.447482004394183
1	1	1	0.466532602066533
2	1	1	0.533467397933467
1	2	1	0.356364544422507
2	2	1	0.643635455577493
1	1	2	0.571950438336247
2	1	2	0.428049561663753
1	2	2	0.439081311724508
2	2	2	0.560918688275492

Expected counts:

e	f	$tcount(e, f)$
the	le	0.99295584002626
the	chien	0.552517995605817
the	chat	0.356364544422507
the	l’	0.571950438336247
the	autobus	0.439081311724508
dog	le	0.473576762040274
dog	chien	0.447482004394183
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.533467397933467
cat	chien	0
cat	chat	0.643635455577493
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.428049561663753
bus	autobus	0.560918688275492

Old and new parameters:

e	f	old	new
the	le	0.23	0.34
the	chien	0.2	0.19
the	chat	0.11	0.12
the	l’	0.25	0.2
the	autobus	0.21	0.15
dog	le	0.2	0.51
dog	chien	0.16	0.49
dog	chat	0.33	0
dog	l’	0.12	0
dog	autobus	0.18	0
cat	le	0.26	0.45
cat	chien	0.28	0
cat	chat	0.19	0.55
cat	l’	0.24	0
cat	autobus	0.03	0
bus	le	0.22	0
bus	chien	0.05	0
bus	chat	0.26	0
bus	l’	0.19	0.43
bus	autobus	0.27	0.57

e	f						
the	le	0.23	0.34	0.46	0.56	0.64	0.71
the	chien	0.2	0.19	0.15	0.12	0.09	0.06
the	chat	0.11	0.12	0.1	0.08	0.06	0.04
the	l’	0.25	0.2	0.17	0.15	0.13	0.11
the	autobus	0.21	0.15	0.12	0.1	0.08	0.07
dog	le	0.2	0.51	0.46	0.39	0.33	0.28
dog	chien	0.16	0.49	0.54	0.61	0.67	0.72
dog	chat	0.33	0	0	0	0	0
dog	l’	0.12	0	0	0	0	0
dog	autobus	0.18	0	0	0	0	0
cat	le	0.26	0.45	0.41	0.36	0.3	0.26
cat	chien	0.28	0	0	0	0	0
cat	chat	0.19	0.55	0.59	0.64	0.7	0.74
cat	l’	0.24	0	0	0	0	0
cat	autobus	0.03	0	0	0	0	0
bus	le	0.22	0	0	0	0	0
bus	chien	0.05	0	0	0	0	0
bus	chat	0.26	0	0	0	0	0
bus	l’	0.19	0.43	0.47	0.47	0.47	0.48
bus	autobus	0.27	0.57	0.53	0.53	0.53	0.52

After 20 iterations:

e	f	
the	le	0.94
the	chien	0
the	chat	0
the	l’	0.03
the	autobus	0.02
dog	le	0.06
dog	chien	0.94
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.06
cat	chien	0
cat	chat	0.94
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.49
bus	autobus	0.51

Model 2 has several local maxima – good one:

e	f	$\mathbf{T}(f \mid e)$
the	le	0.67
the	chien	0
the	chat	0
the	l’	0.33
the	autobus	0
dog	le	0
dog	chien	1
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0
cat	chien	0
cat	chat	1
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0
bus	autobus	1

Model 2 has several local maxima – bad one:

e	f	$\mathbf{T}(f \mid e)$
the	le	0
the	chien	0.4
the	chat	0.3
the	l’	0
the	autobus	0.3
dog	le	0.5
dog	chien	0.5
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.5
cat	chien	0
cat	chat	0.5
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.5
bus	autobus	0.5

another bad one:

e	f	$\mathbf{T}(f \mid e)$
the	le	0
the	chien	0.33
the	chat	0.33
the	l’	0
the	autobus	0.33
dog	le	1
dog	chien	0
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	1
cat	chien	0
cat	chat	0
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	1
bus	autobus	0

- Alignment parameters for good solution:

$$\begin{aligned} \mathbf{T}(i=1 \mid j=1, l=2, m=2) &= 1 \\ \mathbf{T}(i=2 \mid j=1, l=2, m=2) &= 0 \\ \mathbf{T}(i=1 \mid j=2, l=2, m=2) &= 0 \\ \mathbf{T}(i=2 \mid j=2, l=2, m=2) &= 1 \end{aligned}$$

log probability $= -1.91$

- Alignment parameters for first bad solution:

$$\begin{aligned} \mathbf{T}(i=1 \mid j=1, l=2, m=2) &= 0 \\ \mathbf{T}(i=2 \mid j=1, l=2, m=2) &= 1 \\ \mathbf{T}(i=1 \mid j=2, l=2, m=2) &= 0 \\ \mathbf{T}(i=2 \mid j=2, l=2, m=2) &= 1 \end{aligned}$$

log probability $= -4.16$

- Alignment parameters for second bad solution:

$$
\begin{aligned}
\mathbf{T}(i=1 \mid j=1, l=2, m=2) &= 0 \\
\mathbf{T}(i=2 \mid j=1, l=2, m=2) &= 1 \\
\mathbf{T}(i=1 \mid j=2, l=2, m=2) &= 1 \\
\mathbf{T}(i=2 \mid j=2, l=2, m=2) &= 0
\end{aligned}
$$

log probability $= -3.30$

Improving the Convergence Properties of Model 2

- **Out of 100 random starts, only 60 converged to the best local maxima**

- Model 1 converges to the same, global maximum every time (see the Brown et. al paper)

- Method in IBM paper: run Model 1 to estimate **T** parameters, then use these as the initial parameters for Model 2

- In 100 tests using this method, Model 2 converged to the correct point every time.

Evaluation of Machine Translation

Evaluation of Machine Translation Systems

- Method 1: human evaluations
  accurate, **but** expensive, slow

- "Cheap" and fast evaluation is essential

- We'll discuss one prominent method:
  Bleu (Papineni, Roukos, Ward and Zhu, 2002)

Evaluation of Machine Translation Systems

Bleu (Papineni, Roukos, Ward and Zhu, 2002):

Candidate 1: It is a guide to action which ensures that the military always obeys the commands of the party.

Candidate 2: It is to insure the troops forever hearing the activity guidebook that party direct.

Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.

Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.

Reference 3: It is the practical guide for the army always to heed the directions of the party.

Unigram Precision

- **Unigram Precision** of a candidate translation:

$$\frac{C}{N}$$

where N is number of words in the candidate, C is the number of words in the candidate which are in at least one reference translation.

e.g.,

Candidate 1: It is a guide to action which ensures that the military always obeys the commands of the party.

$$Precision = \frac{17}{18}$$

(only *obeys* is missing from all reference translations)

Modified Unigram Precision

- Problem with unigram precision:

  Candidate: the the the the the the the

  Reference 1: the cat sat on the mat

  Reference 2: there is a cat on the mat

  precision = 7/7 = 1???

- **Modified unigram precision:** "Clipping"
  - Each word has a "cap". e.g., *cap(the) = 2*
  - A candidate word w can only be correct a maximum of $cap(w)$ times. e.g., in candidate above, $cap(the) = 2$, and *the* is correct twice in the candidate $\Rightarrow$

$$Precision = \frac{2}{7}$$

Modified N-gram Precision

- Can generalize modified unigram precision to other n-grams.
- For example, for candidates 1 and 2 above:

$$Precision_1(bigram) = \frac{10}{17}$$

$$Precision_2(bigram) = \frac{1}{13}$$

Precision Alone Isn't Enough

Candidate 1: of the

Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.

Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.

Reference 3: It is the practical guide for the army always to heed the directions of the party.

$$Precision(unigram) = 1$$

$$Precision(bigram) = 1$$

But Recall isn't Useful in this Case

- Standard measure used in addition to precision is **recall**:

$$Recall = \frac{C}{N}$$

where C is number of n-grams in candidate that are correct, N is number of words in the references.

Candidate 1: I always invariably perpetually do.

Candidate 2: I always do

Reference 1: I always do

Reference 1: I invariably do

Reference 1: I perpetually do

Sentence Brevity Penalty

- Step 1: for each candidate, compute closest matching reference (in terms of length)
  e.g., our candidate is length 12, references are length $12, 15, 17$. Best match is of length 12.

- Step 2: Say l_i is the length of the i'th candidate, r_i is length of best match for the i'th candidate, then compute

$$brevity = \frac{\sum_i r_i}{\sum_i l_i}$$

  (I think! from the Papineni paper, although $brevity = \frac{\sum_i r_i}{\sum_i min(l_i, r_i)}$ might make more sense?)

- Step 3: compute brevity penalty

$$BP = \begin{cases} 1 & \text{If } brevity < 1 \\ e^{1-brevity} & \text{If } brevity \geq 1 \end{cases}$$

  e.g., if $r_i = 1.1 \times l_i$ for all i (candidates are always 10% too short) then $BP = e^{-0.1} = 0.905$

The Final Score

- Corpus precision for any n-gram is

$$p_n = \frac{\sum_{C \in \{Candidate\}} \sum_{ngram \in C} \quad Count_{clip}(ngram)}{\sum_{C \in \{Candidate\}} \sum_{ngram \in C} \quad Count(ngram)}$$

i.e. number of correct ngrams in the candidates (after "clipping") divided by total number of ngrams in the candidates

- Final score is then

$$Bleu = BP \times (p_1 p_2 p_3 p_4)^{1/4}$$

i.e., BP multiplied by the geometric mean of the unigram, bigram, trigram, and four-gram precisions

Evaluating TMs

- How do we measure TM quality?
  - Method 1: use in an end-to-end translation system
    - Hard to measure translation quality
    - Option: human judges
    - Option: reference translations (NIST, BLEU scores)
  - Method 2: measure quality of the alignments produced
    - Easy to measure
    - Hard to know what the gold alignments should be
    - May not correlate with translation quality (like perplexity in LMs)

Decoding

- In these word-to-word models
  - Finding best alignments is easy
  - Finding translations is hard (why?)

```
it is not clear .
|  \   |    \    \
|   \  +     \    \
|    \/ \     \    \
|    /\  \     \    \
CE NE EST PAS CLAIR  .
```

Bag “Generation” (Decoding)

Exact reconstruction

Please give me your response as soon as possible.
$\Rightarrow$ Please give me your response as soon as possible.

Reconstruction preserving meaning

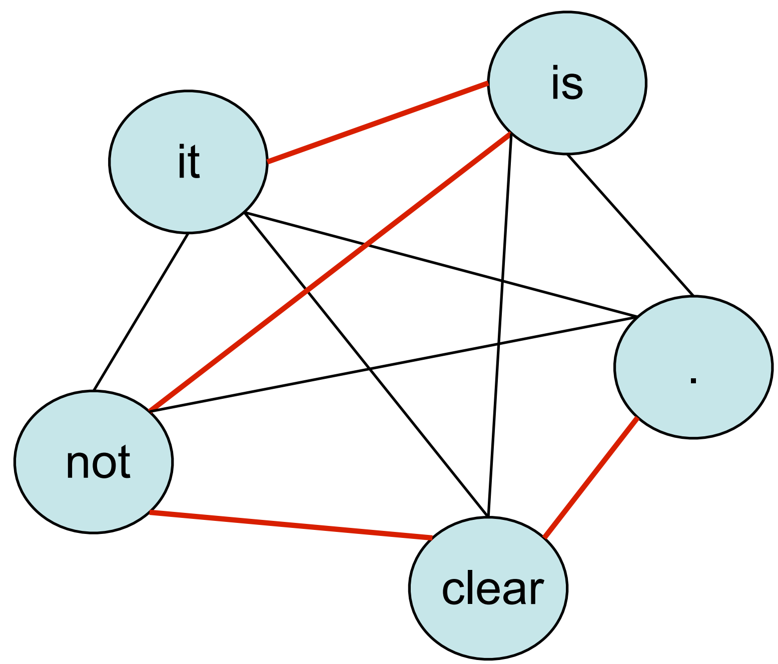
Now let me mention some of the disadvantages.
$\Rightarrow$ Let me mention some of the disadvantages now.

Garbage reconstruction

In our organization research has two missions.
$\Rightarrow$ In our missions research organization has two.

Bag Generation is a TSP

- Imagine bag generation with a bigram LM
  - Words are nodes
  - Edge weights are P(w|w')
  - Valid sentences are Hamiltonian paths
- Not the best news for word-based MT!



Decoding, Anyway

- Simplest possible decoder:
  - Enumerate sentences, score each with TM and LM
- Greedy decoding:
  - Assign each French word it's most likely English translation
  - Operators:
    - Change a translation
    - Insert a word into the English (zero-fertile French)
    - Remove a word from the English (null-generated French)
    - Swap two adjacent English words
  - Do hill-climbing (or annealing)
- You should be able to build a model 1/2 translator now
- More on word alignment, decoding next class

Greedy Decoding