

## Statistical NLP Spring 2008



### Lecture 11: Word Alignment

Dan Klein – UC Berkeley

## Machine Translation: Examples

### 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 Coca Strains and of Giochi 1996, heart of one popolosa metropolitan area, was remained paralyzed.

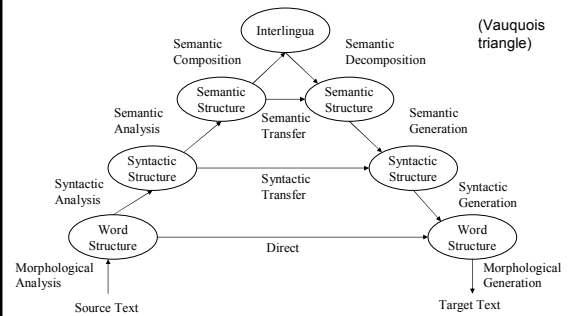
## Machine Translation

Madame la présidente, votre présidence de cette institution a été marquante.  
Mrs Fontaine, your presidency of this institution has been outstanding.  
Madam President, president of this house has been discoveries.  
Madam President, your presidency of this institution has been impressive.

Je vais maintenant m'exprimer brièvement en irlandais.  
I shall now speak briefly in Irish .  
I will now speak briefly in Ireland .  
I will now speak briefly in Irish .

Nous trouvons en vous un président tel que nous le souhaitons.  
We think that you are the type of president that we want.  
We are in you a president as the wanted.  
We are in you a president as we the wanted.

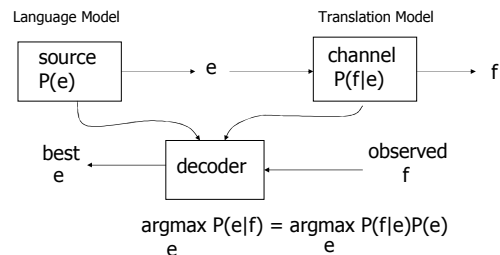
## Levels of Transfer



## 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)

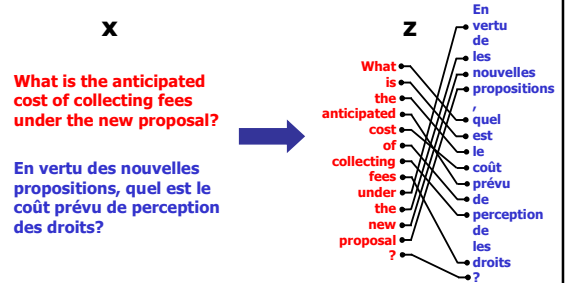
## MT System Components



## 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
- Next few classes
  - More complex word-level and phrase-level TMs
  - Tree-to-tree and tree-to-string TMs
  - More sophisticated decoders

## Word Alignment

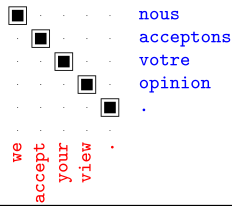


## Unsupervised Word Alignment

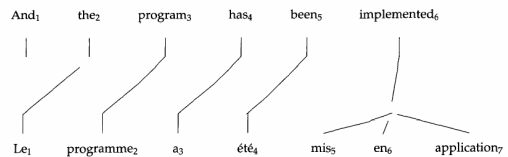
- Input: a *bitext*: pairs of translated sentences

nous acceptons votre opinion .  
we accept your view .

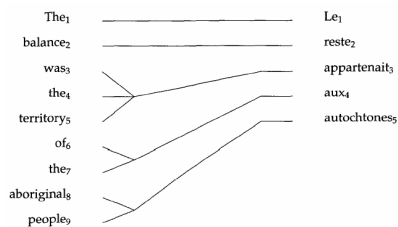
- Output: *alignments*: pairs of translated words
  - When words have unique sources, can represent as a (forward) alignment function  $a$  from French to English positions



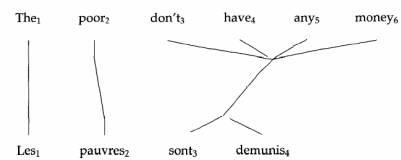
## 1-to-Many Alignments



## Many-to-1 Alignments



## Many-to-Many Alignments



## A Word-Level TM?

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

$e = e_1 \dots e_I$     And<sub>1</sub>    the<sub>2</sub>    program<sub>3</sub>    has<sub>4</sub>    been<sub>5</sub>    implemented<sub>6</sub>  
 $f = f_1 \dots f_J$     Le<sub>1</sub>    programme<sub>2</sub>    a<sub>3</sub>    été<sub>4</sub>    mis<sub>5</sub>    en<sub>6</sub>    applicat<sub>7</sub>

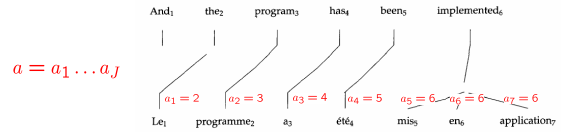
$$P(f|e) = \prod_j P(f_j | e_1 \dots e_I)$$

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.



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

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

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

## IBM Model 1

- Obvious first stab: greedy matchings
- Better approach: re-estimated generative models

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

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

$$P(a_j = i|e, f) = \frac{P(f_j | e_i)}{\sum_{i'} P(f_j | e_{i'})}$$

- Basic idea: pick a source for each word, update co-occurrence statistics, repeat

## Evaluating TMs

- How do we measure quality of a word-to-word model?
  - Method 1: use in an end-to-end translation system
    - Hard to measure translation quality
    - Option: human judges
    - Option: reference translations (NIST, BLEU)
    - Option: combinations (HTER)
    - Actually, no one uses word-to-word models alone as TMs
  - Method 2: measure quality of the alignments produced
    - Easy to measure
    - Hard to know what the gold alignments should be
    - Often does not correlate well with translation quality (like perplexity in LMs)

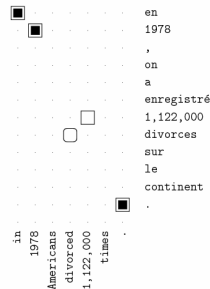
## Alignment Error Rate

- Alignment Error Rate

- ☐ = Sure
- = Possible
- = Predicted

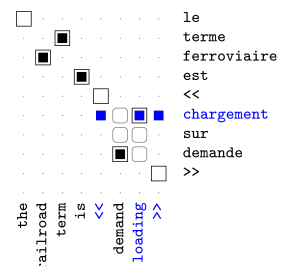
$$AER(A, S, P) = \left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right)$$

$$= \left(1 - \frac{3+3}{3+4}\right) = \frac{1}{7}$$



## Problems with Model 1

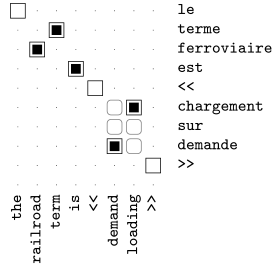
- There's a reason they designed models 2-5!
- Problems: alignments jump around, align everything to rare words
- Experimental setup:
  - Training data: 1.1M sentences of French-English text, Canadian Hansards
  - Evaluation metric: alignment error Rate (AER)
  - Evaluation data: 447 hand-aligned sentences



## Intersected Model 1

- Post-intersection: standard practice to train models in each direction then intersect their predictions [Och and Ney, 03]
- Second model is basically a filter on the first
  - Precision jumps, recall drops
  - End up not guessing hard alignments

Model	P/R	AER
Model 1 E→F	82/58	30.6
Model 1 F→E	85/58	28.7
Model 1 AND	96/46	34.8

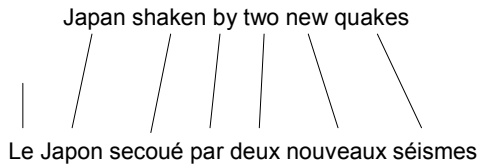


## Joint Training?

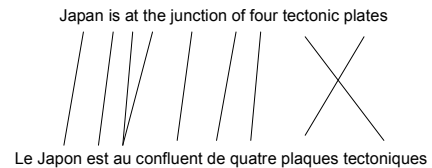
- Overall:
  - Similar high precision to post-intersection
  - But recall is much higher
  - More confident about positing non-null alignments

Model	P/R	AER
Model 1 E→F	82/58	30.6
Model 1 F→E	85/58	28.7
Model 1 AND	96/46	34.8
Model 1 INT	93/69	19.5

## Monotonic Translation



## Local Order Change



## IBM Model 2

- Alignments tend to the diagonal (broadly at least)

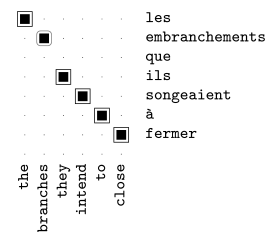
$$P(f, a|e) = \prod_j P(a_j = i|j, I, J) P(f_j|e_i)$$

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

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

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

## Example



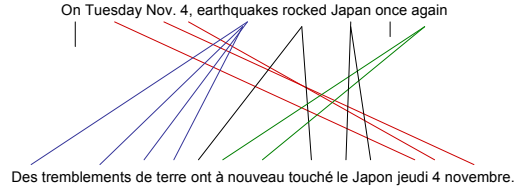
## EM for Models 1/2

- Model 1 Parameters:
  - Translation probabilities (1+2)  $P(f_j|e_i)$
  - Distortion parameters (2 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

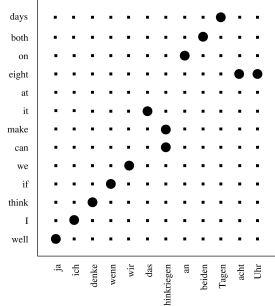
$$P(a_j = i|f, e) = \frac{P(a_j = i|j, I, J)P(f_j|e_i)}{\sum_{i'} P(a_j = i'|j, I, J)P(f_j|e_{i'})}$$

- (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

## Phrase Movement



## Phrase Movement



## The HMM Model

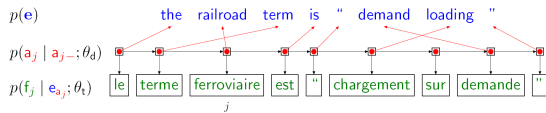
- Model 2 preferred global monotonicity
- We want local monotonicity:
  - Most jumps are small
- HMM model (Vogel 96)

$f$	$i(f e)$
nationale	0.469
national	0.418
nationaux	0.054
nationales	0.029

$$P(f, a|e) = \prod_j P(a_j|a_{j-1})P(f_j|e_i)$$

- Re-estimate using the forward-backward algorithm
- Handling nulls requires some care
- What are we still missing?

## The HMM Model



Distortion  $\theta_d$

$$p(\begin{matrix} \uparrow \\ \uparrow \\ \uparrow \end{matrix}) = 0.6$$

$$p(\begin{matrix} \uparrow \\ \rightarrow \\ \uparrow \end{matrix}) = 0.2$$

$$p(\begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix}) = 0.1$$

Translation  $\theta_t$

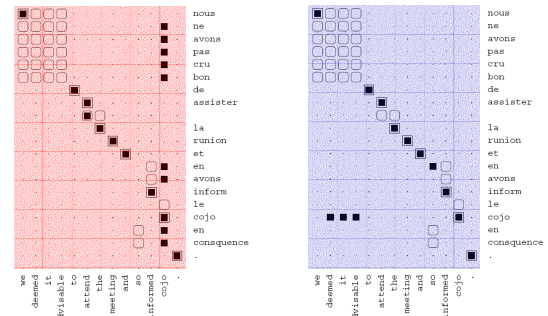
$$p(\text{the} \rightarrow \text{le}) = 0.53$$

$$p(\text{the} \rightarrow \text{la}) = 0.24$$

$$p(\text{railroad} \rightarrow \text{ferroviaire}) = 0.19$$

$$p(\text{NULL} \rightarrow \text{le}) = 0.12$$

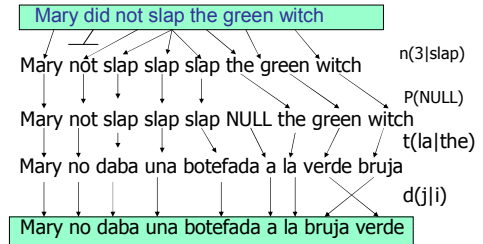
## HMM Examples



## AER for HMMs

Model	AER
Model 1 INT	19.5
HMM E→F	11.4
HMM F→E	10.8
HMM AND	7.1
HMM INT	4.7
GIZA M4 AND	6.9

## IBM Models 3/4/5



[from Al-Onaizan and Knight, 1998]

## Examples: Translation and Fertility

<i>the</i>				<i>not</i>			
f	t(f e)	φ	n(φ e)	f	t(f e)	φ	n(φ e)
le	0.497	1	0.746	ne	0.497	2	0.735
la	0.207	0	0.254	pas	0.442	0	0.154
les	0.155			non	0.029	1	0.107
l'	0.086			rien	0.011		
ce	0.018						
cette	0.011						

<i>farmers</i>			
f	t(f e)	φ	n(φ e)
agriculteurs	0.442	2	0.731
les	0.418	1	0.228
cultivateurs	0.046	0	0.039
producteurs	0.021		

## Example: Idioms

<i>nodding</i>			
f	t(f e)	φ	n(φ e)
signe	0.164	4	0.342
la	0.123	3	0.293
tête	0.097	2	0.167
oui	0.086	1	0.163
fait	0.073	0	0.023
que	0.073		
hoche	0.054		
hocher	0.048		
faire	0.030		
me	0.024		
approuve	0.019		
qui	0.019		
un	0.012		
faites	0.011		

## Example: Morphology

<i>should</i>			
f	t(f e)	φ	n(φ e)
devrait	0.330	1	0.649
devraient	0.123	0	0.336
devrions	0.109	2	0.014
faudrait	0.073		
faut	0.058		
doit	0.058		
aurait	0.041		
doivent	0.024		
devons	0.017		
devrais	0.013		

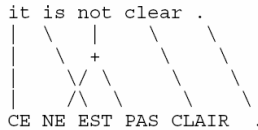
## Some Results

- [Och and Ney 03]

Model	Training scheme	0.5K	8K	128K	1.47M
Dice		50.9	43.4	39.6	38.9
Dice+C		46.3	37.6	35.0	34.0
Model 1	1 <sup>5</sup>	40.6	33.6	28.6	25.9
Model 2	1 <sup>5</sup> 2 <sup>5</sup>	46.7	29.3	22.0	19.5
HMM	1 <sup>5</sup> H <sup>5</sup>	26.3	23.3	15.0	10.8
Model 3	1 <sup>5</sup> 2 <sup>5</sup> 3 <sup>3</sup>	43.6	27.5	20.5	18.0
	1 <sup>5</sup> H <sup>5</sup> 3 <sup>3</sup>	27.5	22.5	16.6	13.2
Model 4	1 <sup>5</sup> 2 <sup>5</sup> 3 <sup>3</sup> 4 <sup>3</sup>	41.7	25.1	17.3	14.1
	1 <sup>5</sup> H <sup>5</sup> 3 <sup>3</sup> 4 <sup>3</sup>	26.1	20.2	13.1	9.4
	1 <sup>5</sup> H <sup>5</sup> 4 <sup>3</sup>	26.3	21.8	13.3	9.3
Model 5	1 <sup>5</sup> H <sup>5</sup> 4 <sup>3</sup> 5 <sup>3</sup>	26.5	21.5	13.7	9.6
	1 <sup>5</sup> H <sup>5</sup> 3 <sup>3</sup> 4 <sup>3</sup> 5 <sup>3</sup>	26.5	20.4	13.4	9.4
Model 6	1 <sup>5</sup> H <sup>5</sup> 4 <sup>3</sup> 6 <sup>3</sup>	26.0	21.6	12.8	8.8
	1 <sup>5</sup> H <sup>5</sup> 3 <sup>3</sup> 4 <sup>3</sup> 6 <sup>3</sup>	25.9	20.3	12.5	8.7

## Decoding

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



## Bag "Generation" (Decoding)

*Exact reconstruction* (24 of 38)

Please give me your response as soon as possible.  
 ⇒ Please give me your response as soon as possible.

*Reconstruction preserving meaning* (8 of 38)

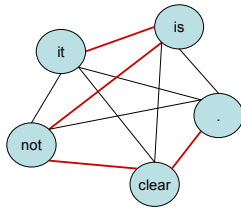
Now let me mention some of the disadvantages.  
 ⇒ Let me mention some of the disadvantages now.

*Garbage reconstruction* (6 of 38)

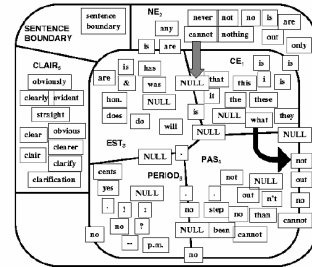
In our organization research has two missions.  
 ⇒ In our missions research organization has two.

## Bag Generation as 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!



## IBM Decoding as a TSP



## Decoding, Anyway

- Simplest possible decoder:
  - Enumerate sentences, score each with TM and LM
- Greedy decoding:
  - Assign each French word its 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)

## Greedy Decoding

NULL well heard , it talks a great victory .  
 bien entendu , il parle de une belle victoire .  
 translateTwoWords(2,understood,0,about)

NULL well understood , it talks about a great victory .  
 bien entendu , il parle de une belle victoire .  
 translateOneWord(4,he)

NULL well understood , he talks about a great victory .  
 bien entendu , il parle de une belle victoire .  
 translateTwoWords(1,quite,2,naturally)

NULL quite naturally , he talks about a great victory .  
 bien entendu , il parle de une belle victoire .

## Stack Decoding

---

- **Stack decoding:**
  - Beam search
  - Usually A\* estimates for completion cost
  - One stack per candidate sentence length
- **Other methods:**
  - Dynamic programming decoders possible if we make assumptions about the set of allowable permutations

sent length	decoder type	time (sec/sent)	search errors	translation errors (semantic and/or syntactic)	NE	PME	DSE	FSE	HSE	CE
6	IP	47.50	0	57	44	57	0	0	0	0
6	stack	0.79	5	58	43	53	1	0	0	4
6	greedy	0.07	18	60	38	45	5	2	1	10
8	IP	499.00	0	76	27	74	0	0	0	0
8	stack	5.67	20	75	24	57	1	2	2	15
8	greedy	2.66	43	75	20	38	4	5	1	33