

## Natural Language Processing: MT & Word Alignment Models



Christopher Manning

Borrows some slides from Kevin Knight, Dan Klein,  
and Bill MacCartney



### Lecture Plan

1. A bit more course overview [5 mins]
2. Briefly learn the history of Machine Translation [10 mins]
3. Learn about translation with one example [10 mins]
4. Speech recognition & Applying it to MT: The noisy channel model [10 mins]  
[Stretch! **Emergency time reserves:** 5 mins]
5. Parallel-text word alignments: the IBM models [30 mins]



### Lecture Plan

1. A bit more course overview [5 mins]
2. Briefly learn the history of Machine Translation [10 mins]
3. Learn about translation with one example [10 mins]
4. Speech recognition & Applying it to MT: The noisy channel model [10 mins]  
[Stretch! **Emergency time reserves:** 5 mins]
5. Parallel-text word alignments: the IBM models [30 mins]



### Learning Goals

1. To be able to **write** a natural language processing system which isn't just a keyword spotter or a bag-of-words text classifier
2. To have a good sense of the complexity of natural language understanding (by either computers or humans)
3. To be aware of different levels of linguistic representation: words, syntax, semantics and tools used to model them
4. To be familiar with major approaches to natural language processing: rule-based, statistical NLP, other machine learning approaches including deep learning
5. To think that human languages and NLP are cool ☺



### The class

#### Topics

- The course is not all MT! But I find it a fun place to start...
- Syntactic parsing, coreference resolution, named entity recognition, computational semantics, applications
- Classifiers, probabilistic models, deep learning, sequence models, generative and discriminative models applied to NLP
- Organization
  - Make sure you're on OpenEdX, Piazza, mailing list, website
    - Read about grading, collaboration, honor code, etc.
  - Programming assignment 1 (MT) is out today (!)
    - Need to use Java (except Final Project); encouraged to work in pairs

5

## Do some reading!

- Jurafsky and Martin, Chapter 25: MT
  - Great for big picture of MT, okay for Model 1, no Model 2
- Adam Lopez, *Statistical Machine Translation*
  - Fairly modern, comprehensive Stat MT survey
- Brown et al. 1993 – *The Mathematics of Statistical Machine Translation*<sup>2</sup>
- Kevin Knight, *A Statistical MT Tutorial Workbook*
  - Professor's old favorite, but maybe time to move on? No model 2
- Michael Collins, *Statistical MT: IBM Models 1 & 2*
  - Closest to assignment/lecture notation. Maybe start here?
- Philip Koehn, *Statistical MT*, ch. 4: Word-based Models
  - Makes it too easy?!? Gee, there's pseudo-code there...



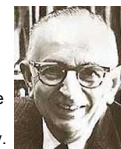
## Lecture Plan

1. A bit more course overview [5 mins]
2. **Briefly learn the history of Machine Translation [10 mins]**
3. Learn about translation with one example [10 mins]
4. Speech recognition & Applying it to MT: The noisy channel model [10 mins]  
[Stretch! **Emergency time reserves:** 5 mins]
5. Parallel-text word alignments: the IBM models [30 mins]



## MT: Just a Code?

"Also knowing nothing official about, but having guessed and inferred considerable about, the powerful new mechanized methods in cryptography—methods which I believe succeed even when one does not know what language has been coded—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)



"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, March 1947



"... as to the problem of mechanical translation, I frankly am afraid that the [semantic] boundaries of words in different languages are too vague ... to make any quasi-mechanical translation scheme very hopeful."

– Norbert Wiener, April 1947



## The early history: 1950s

- Early powerful computers
- Foundational probability
- First serious attempts
- MT has substantial history
- Little semantic understanding
- Problems



## Machine Translation History

[see <http://www.hutchinsweb.me.uk/history.htm>]

- 1950s: Intensive research activity in MT
- 1960s: Direct word-for-word replacement
- 1966 (ALPAC): NRC Report on MT
  - Conclusion: MT not worthy of serious scientific investigation; focus on understanding language
- 1966–1975: MT winter
- 1975–1985: Resurgence in Europe and Japan
  - Domain-specific rule-based systems
- 1985–1993: Gradual Resurgence in the US
- 1993–2012: Statistical MT surges! Field becomes popular
- 2013: Out of data, might need new theory/models?
- 2014–2015: People excited about new neural MT models



## What has happened between ALPAC and Now?

- Need for MT and other NLP applications confirmed
- Computers have become faster, more powerful
- Hugely increased availability of data: WWW
- People understand more about linguistics
- Change in expectations of quality
- Development of empirical, data-intensive, statistical, hybrid statistical/grammar-based, and neural approaches



## Lecture Plan

1. A bit more course overview [5 mins]
2. Briefly learn the history of Machine Translation [10 mins]
3. Learn about translation with one example [10 mins]
4. Speech recognition & Applying it to MT: The noisy channel model [10 mins]  
[Stretch! Emergency time reserves: 5 mins]
5. Parallel-text word alignments: the IBM models [30 mins]



أقرت الحكومة المصرية تعديل القانون الانتخابي الذي على أساسه ستجري الانتخابات البرلمانية المرتقبة قبل نهاية العام. وغلبت فيه الاقتراع بنظام القائمة على النظام الفردي. وذلك بعد اعتراضات قوية على النسخة السابقة من القانون.

The Egyptian government approved on Sunday the electoral law amendment on the basis of which the forthcoming parliamentary elections will be held before the end of the year,

On Sunday said the government approved the amended electoral law on the basis of the forthcoming parliamentary elections will be held before the end of the year, dominated the voting system based on individual system, after strong objections to the previous version of the law.

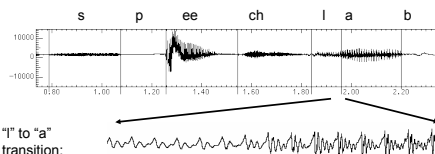


## Lecture Plan

1. A bit more course overview [5 mins]
2. Briefly learn the history of Machine Translation [10 mins]
3. Learn about translation with one example [10 mins]
4. Speech recognition & Applying it to MT: The noisy channel model [10 mins]  
[Stretch! Emergency time reserves: 5 mins]
5. Parallel-text word alignments: the IBM models [30 mins]

## Speech Recognition: Acoustic Waves

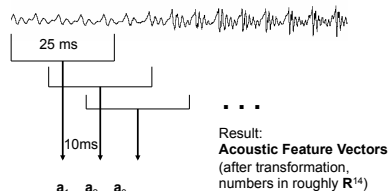
- Human speech generates a wave  
– like a loudspeaker moving its magnet
- A wave for the words "speech lab" looks like:



From Simon Arnfield's web tutorial on speech, Sheffield:  
<http://www.psyc.leeds.ac.uk/research/cogn/speech/tutorial/>

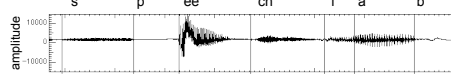
## Acoustic Sampling

- 10 ms frame (ms = millisecond = 1/1000 second)
- ~25 ms window around frame [wide band] to allow/ smooth signal processing – it let's you see formants

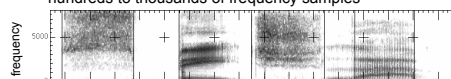


## Spectral Analysis

- Frequency gives pitch; amplitude gives volume
  - sampling at ~8 kHz phone, ~16 kHz mic (kHz=1000 cycles/sec)



- Fourier transform of wave displayed as a spectrogram
  - darkness indicates energy at each frequency
  - hundreds to thousands of frequency samples

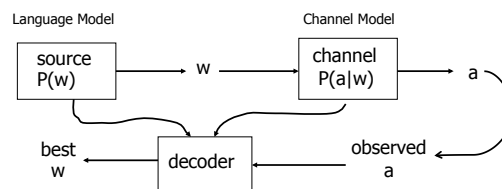


## The Speech Recognition Problem

- **Speech Recognition by the Noisy Channel Model**
  - Build a generative model of encoding: We started with English words, they are transmitted as an audio signal, and we now wish to decode what we hear.
  - Listener finds most likely sequence **w** of “words” given the sequence of acoustic observation vectors **a**
- Use this **generative model** to decode:
- That is, use Bayes Rule

Automatic Speech Recognition

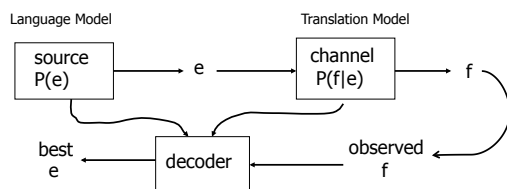
## ASR System Components



$$\hat{w} = \operatorname{argmax}_w P(w | a) = \operatorname{argmax}_w \frac{P(a | w)P(w)}{P(a)}$$

$$= \operatorname{argmax}_w P(a | w)P(w)$$

## MT System Components



$$\hat{e} = \operatorname{argmax}_e P(e | f) = \operatorname{argmax}_e \frac{P(f | e)P(e)}{P(f)}$$

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

## Other Noisy-Channel Processes

- Handwriting recognition
 
$$P(\text{text} | \text{strokes}) \propto P(\text{text})P(\text{strokes} | \text{text})$$
- OCR
 
$$P(\text{text} | \text{pixels}) \propto P(\text{text})P(\text{pixels} | \text{text})$$
- Spelling Correction
 
$$P(\text{text} | \text{typos}) \propto P(\text{text})P(\text{typos} | \text{text})$$

## Statistical MT

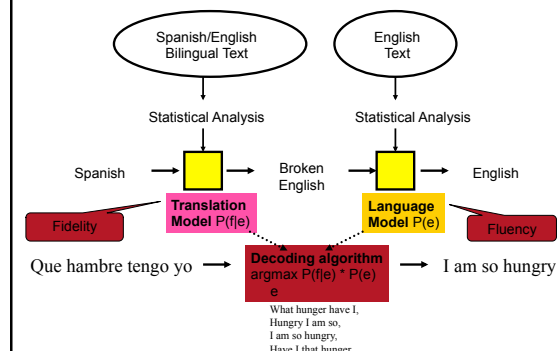
Pioneered at IBM in the early 1990s

Let's make a probabilistic model of translation  
 $P(e | f)$

Suppose  $f$  is *de rien*  
 $P(\text{you're welcome} | \text{de rien}) = 0.45$   
 $P(\text{nothing} | \text{de rien}) = 0.13$   
 $P(\text{pidling} | \text{de rien}) = 0.01$   
 $P(\text{underpants} | \text{de rien}) = 0.000000001$



## A Division of Labor



## Lecture Plan

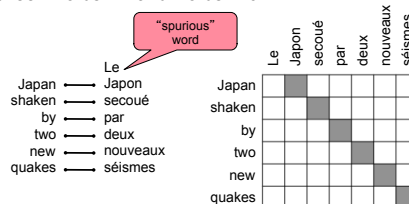
1. A bit more course overview [5 mins]
2. Briefly learn the history of Machine Translation [10 mins]
3. Learn about translation with one example [10 mins]
4. Speech recognition & Applying it to MT: The noisy channel model [10 mins]

[Stretch! Emergency time reserves: 5 mins]

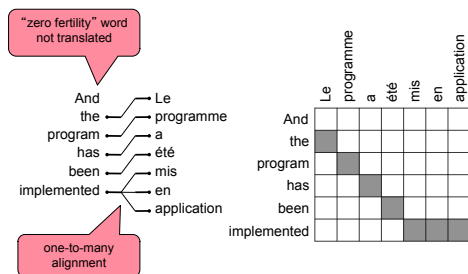
5. Parallel-text word alignments: the IBM models [30 mins]

## Alignments

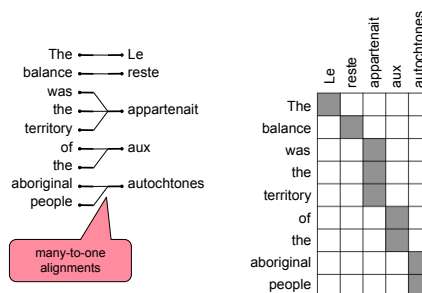
We can factor the translation model  $P(f | e)$  by identifying *alignments* (correspondences) between words in  $f$  and words in  $e$



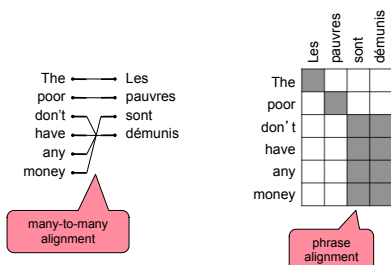
## Alignments: harder



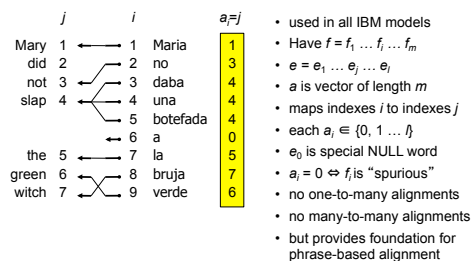
## Alignments: harder



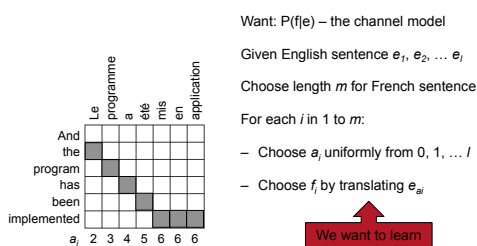
## Alignments: hardest



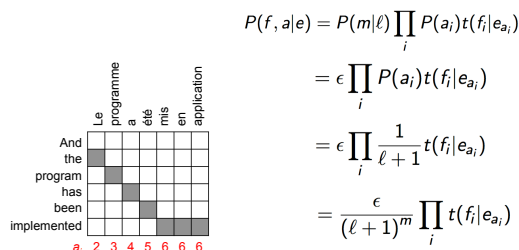
## Alignment as a vector



## IBM Model 1 generative story



## IBM Model 1 parameters



## Applying Model 1\*

$P(f, a | e)$  can be used as a *translation model* or an *alignment model*

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

As alignment model  $P(a|e, f) = \frac{P(f, a|e)}{P(f|e)}$

$$= \frac{P(f, a|e)}{\sum_{a'} P(f, a'|e)}$$

\* Actually, any  $P(f, a | e)$ , e.g., any IBM model

## Unsupervised Word Alignment

Input: a *bitext*: pairs of translated sentences  
Output: *alignments*: pairs of translated words



Starting point: All word alignments equally likely

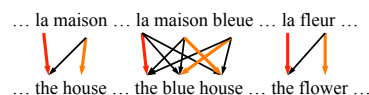
All  $P(\text{french-word} | \text{english-word})$  equally likely

## Unsupervised Word Alignment



"la" and "the" observed to co-occur frequently,  
so  $P(\text{la} | \text{the})$  is increased.

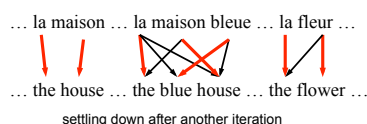
## Unsupervised Word Alignment



"maison" co-occurs with both "the" and "house", but  
 $P(\text{maison} | \text{house})$  can be raised without limit, to 1.0,  
while  $P(\text{maison} | \text{the})$  is limited (see 3<sup>rd</sup> example)

(pigeonhole principle)

## Unsupervised Word Alignment



settling down after another iteration

That was the idea of IBM Model 1 !

## Model 1: Word alignment learning with Expectation-Maximization (EM)

- Start with  $t(f^p|e^q)$  uniform, including  $P(f^p|\text{NULL})$
- For each sentence pair  $(e, f)$

- For each French position  $i$ 
  - Calculate posterior over English positions  $P(a_i | e, f)$

$$P(a_i = j | f, e) = \frac{t(f_i | e_j)}{\sum_{j'} t(f_i | e_{j'})}$$

- Increment count of word  $f_i$  translating each word  $e_{a_i}$ 
  - $C(f_i | e_j) += P(a_i = j | f, e)$

Renormalize counts to give probs  $t(f^p|e^q) = \frac{C(f^p|e^q)}{\sum_{f^x} C(f^x|e^q)}$

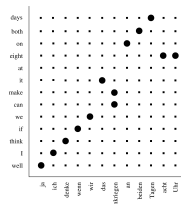
- Iterate until convergence

## IBM Models 1,2,3,4,5

- Models for  $P(f|e)$  and  $P(a|f,e)$  via  $P(f,a|e)$
- There is a set of English words and the extra English word NULL
- Each English word generates and places 0 or more French words
- Any remaining French words are deemed to have been produced by NULL
  - "Spurious" words

## IBM Models 1,2,3,4,5

- In Model 2, the placement of a word in the French depends on where it was in the English



• Unlike Model 1, Model 2 captures the intuition that translations should usually "lie along the diagonal"

• A main focus of PA #1

## IBM Models 1,2,3,4,5

- In Model 3, we model how many French words an English word can produce, using a concept called *fertility*

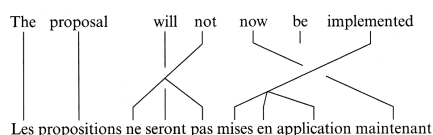
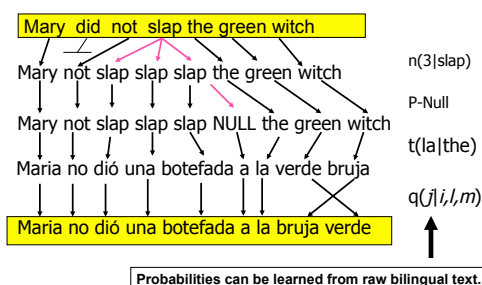


Figure 32.3  
Alignment example.

## Model 3 generative story



Probabilities can be learned from raw bilingual text.

## IBM Model 3 (from Knight 1999)

- For each word  $e_j$  in English sentence, choose a **fertility**  $\phi_j$ . The choice of  $\phi_j$  depends only on  $e_j$ , not other words or  $\phi$ 's:  $n(\phi_j | e_j)$
- For each word  $e_j$ , generate  $\phi_j$  French words. Choice of French word depends only on English word  $e_j$ , not on English context or any other French words.
- Permute all the French words. Each French word gets assigned absolute target position slot (1,2,3, etc.). Choice of French word position dependent only on absolute position of English word generating it and sentence lengths

## Model 3: $P(f|e)$ parameters

- What are the parameters for this model?
- Word translation:**  $t(\text{casa} | \text{house})$
- Spurious words:**  $t(f_i | \text{NULL})$
- Fertilities:**  $n(1|\text{house})$ : prob that "house" will produce 1 Spanish word whenever it appears.
- Distortions:**  $q(5|2,4,6)$ : prob that word in position 2 of French translation was generated by word in position 5 of English sentence, given that 4 is length of English sentence, 6 is French length

## Spurious words

- We could have  $n(3|\text{NULL})$  (probability of there being exactly 3 spurious words in a French translation)
  - But seems wrong...
- Instead, of  $n(0|\text{NULL})$ ,  $n(1|\text{NULL})$  ...  $n(25|\text{NULL})$ , have a single parameter  $p_1$
- After assign fertilities to non-NULL English words we want to generate (say)  $z$  French words.
- As we generate each of  $z$  words, we optionally toss in spurious French word with probability  $p_1$
- Probability of not adding spurious word:  $p_0 = 1 - p_1$

## Distortion probabilities for spurious words

- Shouldn't just have  $q(0|5,4,6)$ , i.e., chance that source position for word 5 is position 0 (NULL).
- Why? These are spurious words! Could occur anywhere!! Too hard to predict
- Instead,
  - Use normal-word distortion parameters to choose positions for normally-generated French words
  - Put NULL-generated words into empty slots left over
  - If three NULL-generated words, and three empty slots, then there are  $3!$ , or six, ways for slotting them all in
  - We'll assign a probability of  $1/6$  for each way!



### Model 3 parameters

- $n, t, p, q$
- Again, if we had complete data of English strings and step-by-step rewritings into Spanish, we could:
  - Compute  $n(0|did)$  by locating every instance of “did”, and seeing how many words it translates to
  - $t(maison|house)$  how many of all French words generated by “house” were “maison”
  - $q(5|2,4,6)$  out of all times some second word is in a translation, how many times did it come from the fifth word (in sentences of length 4 and 6 respectively)?

### Since we don't have word-aligned data...

- We bootstrap alignments from incomplete data
- From a sentence-aligned bilingual corpus
  - 1) Assume some startup values for  $n, q, t, p$ .
  - 2) Use values for  $n, q, t, p$  in model 3 to work out chances of different possible alignments. Use these alignments to update values of  $n, q, t, p$ .
  - 3) Go to 2
- This is a more complicated case of the EM algorithm

Difficulty: Alignments are no longer independent of each other. Have to use approximate inference

### Examples: translation & fertility

<i>the</i>				<i>not</i>			
$f$	$t(f e)$	$\phi$	$n(\phi e)$	$f$	$t(f e)$	$\phi$	$n(\phi e)$
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>			
$f$	$t(f e)$	$\phi$	$n(\phi e)$
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>			
$f$	$t(f e)$	$\phi$	$n(\phi e)$
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		

he is nodding  
 il hoche la tête

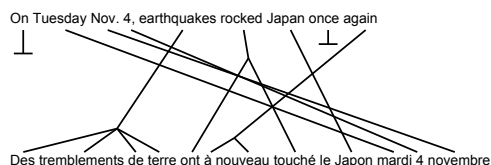
### Example: morphology

<i>should</i>			
$f$	$t(f e)$	$\phi$	$n(\phi e)$
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		

### IBM Models 1,2,3,4,5

- In model 4 the placement of later French words produced by an English word depends on what happened to earlier French words generated by that same English word

## Alignments: linguistics



## IBM Models 1,2,3,4,5

- In model 5 they patch model 4. They make it do non-deficient alignment. That is, you can't put probability mass on impossible things.

## IBM StatMT Translation Models

- IBM1 – lexical probabilities only
- IBM2 – lexicon plus absolute position
- HMM – lexicon plus relative position
- IBM3 – plus fertilities
- IBM4 – inverted relative position alignment
- IBM5 – non-deficient version of model 4
- All the models above handle 0:1, 1:0, 1:1, 1:n alignments *only*

[Brown et al. 93, Vogel et al. 96]

## Why all the models?

- We don't start with aligned text, so we have to get initial alignments from somewhere.
- The alignment space has many local maxima
- Model 1 is words only, a simple model that is relatively easy and fast to train.
- The output of M1 can be a good place to start M2
  - “Starting small”. Also, it's *convex*!
- The sequence of models allows a better model to be found, faster
  - The intuition is like *deterministic annealing*