CMPN463: Natural Language Processing

Lecture 09: Machine Translation I

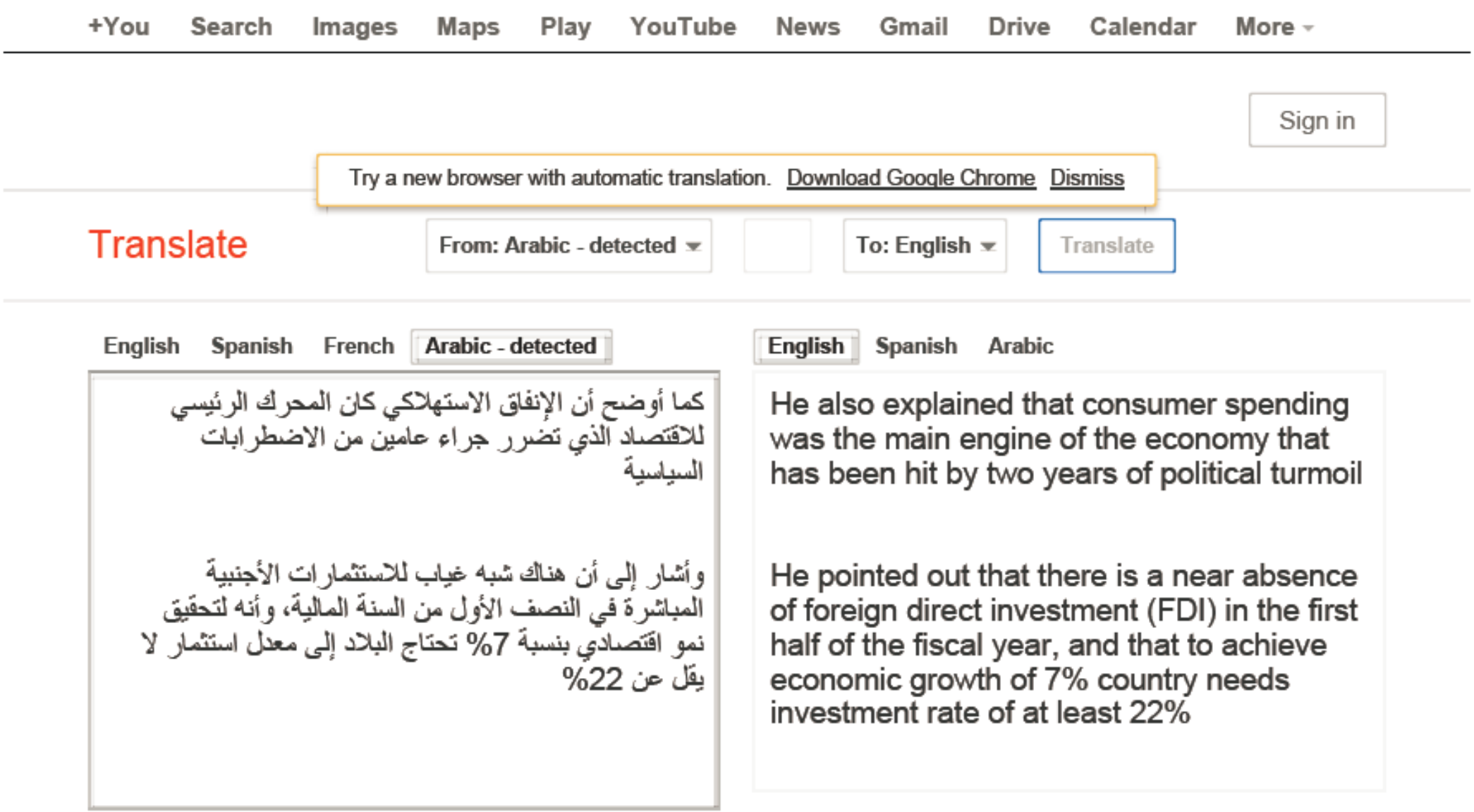
Mohamed Alaa El-Dien Aly
Computer Engineering Department
Cairo University
Fall 2013

Agenda

- Introduction
- Challenges of Machine Translation (MT)
- Classical Approaches
  - Direct MT
  - Transfer Based MT
  - Interlingua-Based MT
- Introduction to Statistical Machine Translation (SMT)
- IBM Models
- Training of IBM Models

Acknowledgment: Most slides adapted from Michael Collins NLP class on Coursera.

Introduction



Challenges: Lexical Ambiguity

Example1:

book the flight reservar
read the book libro

Example2:

the box was in the pen
the pen was on the table

Example3:

kill a man matar
kill a process acabar

Challenges: 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*

Challenges: Syntactic Structure Not Preserved Across Translation

(Example from Dorr et al. 1999)

The bottle floated into the cave

⇓

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

Challenges: Syntactic Ambiguity

(Example from Dorr et al. 1999)

John hit the dog with the stick

⇓

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

(hit with the stick OR the dog with the stick)

Challenges: Pronoun Resolution

(Example from Dorr et al. 1999)

The computer outputs the data; it is fast.

⇓

La computadora imprime los datos; es rapida

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

⇓

La computadora imprime los datos; estan almacendos en ascii

Direct Machine Translation

- Translation is word-by-word
- Very little analysis of the source text (e.g.,no syntactic or semantic analysis)
- Relies on a large bilingual dictionary. For each word in the source language, the dictionary specifies a set of rules for translating that word
- After the words are translated, simple reordering rules are applied (e.g. move adjectives after nouns when translating from English to French)

Example of a set of Direct Translation Rules

(From Jurafsky and Martin, edition 2, chapter 25. Originally from a system from Panov 1960)

Rules for translating much or many into Russian:

if preceding word is *how* **return** *skol'ko*

elseif preceding word is *as* **return** *stol'kozhe*

elseif word is *much*

  if preceding word is *very* **return** nil

  elseif following word is a noun **return** *mnogo*

else(wordismany)

  if preceding word is a preposition and following word is noun **return** *mnogii*

  else return *mnogo*

Problems with Direct Machine Translation

- Lack of any analysis of the source language causes several problems, for example:
  - Difficult or impossible to capture long-range reorderings

    English: Sources said that IBM bought Lotus yesterday

    Japanese: Sources yesterday IBM Lotus bought that said
- Words are translated without disambiguation of their syntactic role e.g., *that* can be a complementizer or determiner, and will often be translated differently for these two cases:

  They said *that* ...

  They like *that* ice-cream

Transfer Based Approaches

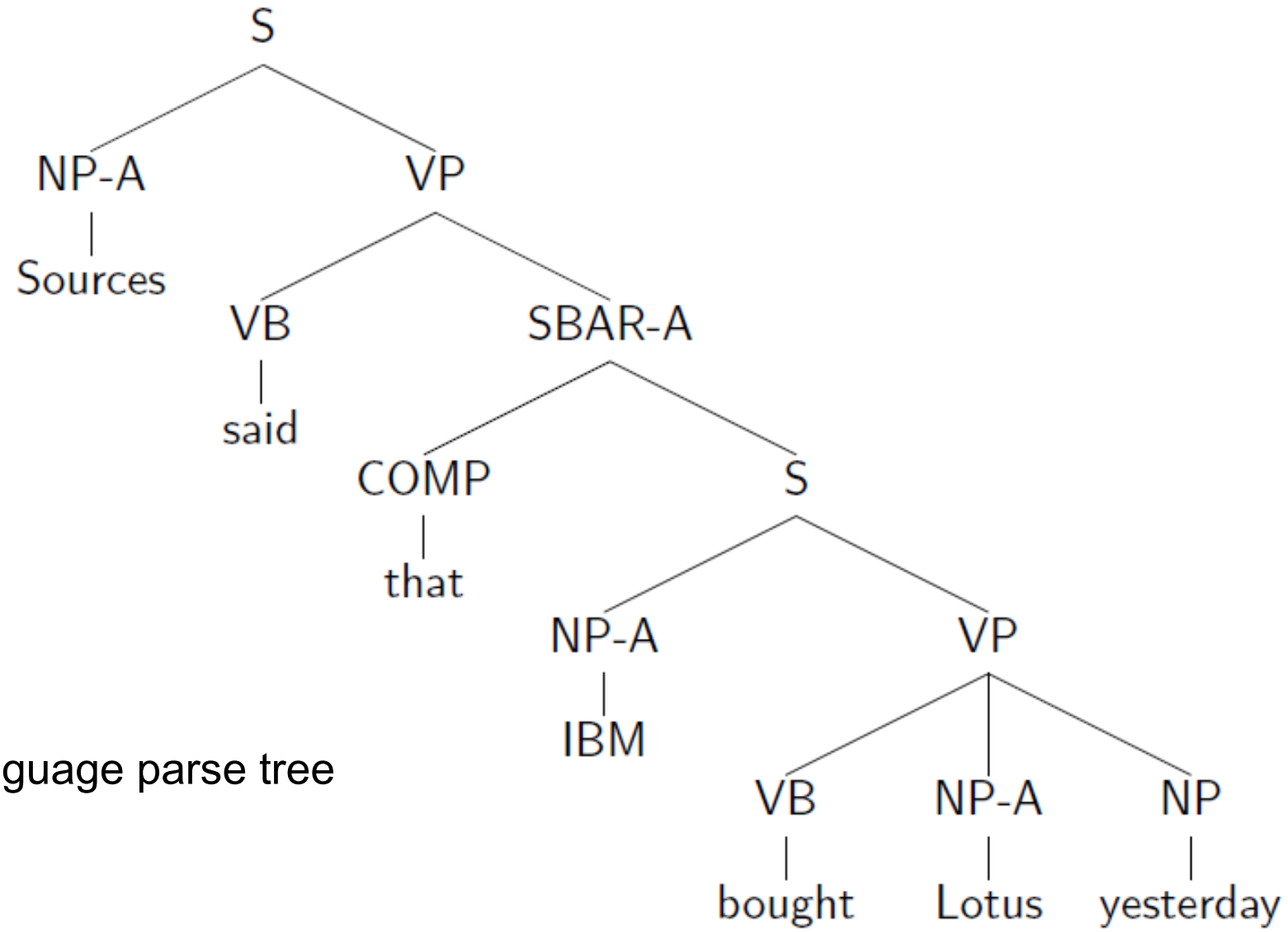
Three phases in translation:

- Analysis: Analyze the source language sentence; for example, build a syntactic analysis of the source language sentence.
- Transfer: Convert the source-language parse tree to a target-language parse tree.
- Generation: Convert the target-language parse tree to an output sentence.

Transfer Based Approaches

- The “parse trees” involved can vary from shallow analyses to much deeper analyses (even semantic representations).
- The transfer rules might look quite similar to the rules for direct translation systems. But they can now operate on syntactic structures.
- It's easier with these approaches to handle long-distance reorderings
- The *Systran* systems are a classic example of this approach

Example

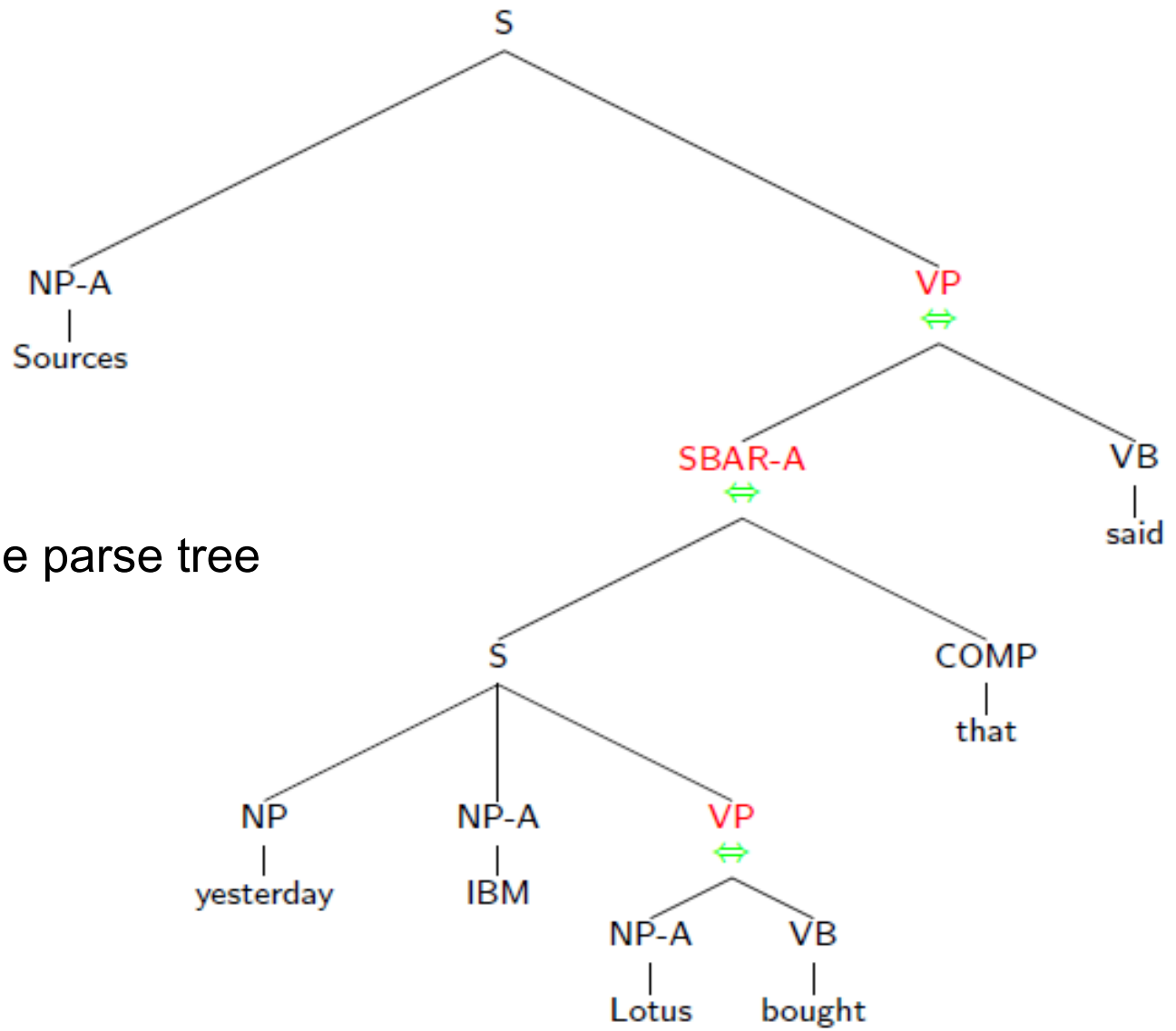


Source language parse tree

English: Sources said that IBM bought Lotus yesterday

Example

Target language parse tree



→ Japanese: Sources yesterday IBM Lotus bought that said

Interlingua-Based Translation

Two phases in translation:

- Analysis: Analyze the source language sentence into a (language-independent) representation of its meaning.
- Generation: Convert the meaning representation into an output sentence.

Interlingua-Based Translation

One Advantage: If we want to build a translation system that translates between n languages, we need to develop n analysis and generation systems. With a transfer based system, we'd need to develop $O(n^2)$ sets of translation rules.

Disadvantage: What would a language-independent representation look like?

Interlingua-Based Translation

- How to represent different concepts in an interlingua?
- Different languages break down concepts in quite different ways:
  - German has two words for *wall*: one for an internal wall, one for a wall that is outside
  - Japanese has two words for *brother*: one for an elder brother, one for a younger brother
  - Spanish has two words for *leg*: pierna for a human's leg, pata for an animal's leg, or the leg of a table

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).
- Idea goes back to Warren Weaver (1949): suggested applying statistical and cryptanalytic techniques to translation.

Introduction to Statistical MT

... 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, 1949, in a letter to Norbert Wiener)

The Noisy Channel Model

- Goal: translation system from French to English
- Have a model $p(e \mid f)$ which estimates conditional probability of any English sentence e given the French sentence f. Use the training corpus to set the parameters.
- A Noisy Channel Model has two components:
  - $p(e)$ the language model
  - $p(f \mid e)$ the translation model
- Which gives us:

$$p(e \mid f) = \frac{p(e, f)}{p(f)} = \frac{p(e)\, p(f \mid e)}{\sum_e p(e)\, p(f \mid e)}$$

and

$$\text{argmax}_e\, p(e \mid f) = \text{argmax}_e\, p(e)\, p(f \mid e)$$

The Noisy Channel Model

- The language model $p(e)$ could be a trigram model, estimated from any data (parallel corpus not needed to estimate the parameters)
- The translation model $p(f \mid e)$ is trained from a parallel corpus of French/English pairs.
- Note:
  - The translation model is backwards!
  - The language model can make up for deficiencies of the translation model.
  - Later we'll talk about how to build $p(f \mid e)$
  - Decoding, i.e. finding $\text{argmax}_e \, p(e) \, p(f \mid e)$ is also a challenging problem.

Example from Koehn and Knight tutorial

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

Que hambre tengo yo

What hunger have	$p(s\|e)$ = 0.000014
Hungry I am so	$p(s\|e)$ = 0.000001
I am so hungry	$p(s\|e)$ = 0.0000015
Have i that hunger	$p(s\|e)$ = 0.000020

Example from Koehn and Knight tutorial

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

Que hambre tengo yo

What hunger have	$p(s\|e)p(e)$ = 0.000014 × 0.000001
Hungry I am so	$p(s\|e)p(e)$ = 0.000001 × 0.0000014
I am so hungry	$p(s\|e)p(e)$ = 0.0000015 × 0.0001
Have i that hunger	$p(s\|e)p(e)$ = 0.000020 × 0.00000098

IBM Models

- The earliest statistical machine translation models
- Used to help us model $p(f|e)$
- We will discuss only two models:
  - IBM Model 1
  - IBM Model 2
- They can be trained without supervision from a parallel corpus

Alignments

- How do we model $p(f|e)$?

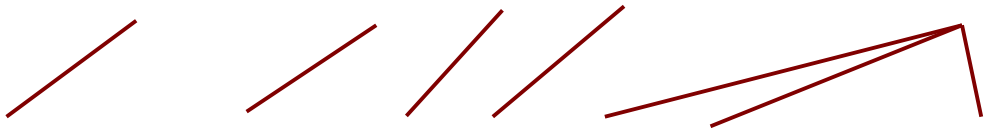
  English: The dog eats
  French: Le chien mange

- English sentence e has l words $e_1, \ldots, e_l$
- French sentence f has m words $f_1, \ldots, f_m$
- An alignment a identifies the source of each *french* word
  - Above: $\{1, 2, 3\}$
- Formally, an alignment a is $\{a_1, \ldots, a_m\}$ where each $a_i \in \{0, \ldots, l\}$. Why 0?
  - French words with no English equivalent ($NULL$ word)
- How many possible alignments?
  - $(l+1)^m$

Alignments

e.g., l = 6, m = 7

e = And the program has been implemented



f = Le programme a ete mis en application

One alignment is

{2, 3, 4, 5, 6, 6, 6}

Another (bad!) alignment is

{1, 1, 1, 1, 1, 1, 1}

Alignments in IBM Models

- We'll define models for $p(a|e, m)$ and $p(f | a, e, m)$, giving

$$p(f, a|e, m) = p(a|e, m)\, p(f|a, e, m)$$

Example: e = the dog eats m = 3

f = f_1 f_2 f_3

We can estimate $p(\text{le chien mange}, \{1, 2, 3\}|\text{the dog eats}, 3)$

- Also,

$$p(f|e, m) = \sum_{a \in A} \overbrace{p(a|e,m)\, p(f|a, e, m)}^{p(f, a|e, m)}$$

where A is the set of all possible alignments

By-product: Most Likely Alignments

- Once we have a model for $p(f, a \mid e, m) = p(a \mid e, m)\, p(f \mid a, e, m)$, we can calculate

$$p(a \mid f, e, m) = \frac{p(f, a \mid e, m)}{\underbrace{\sum_{\alpha \in A} p(f, \alpha \mid e, m)}_{p(f \mid e, m)}}$$

  for any alignment a.

- For a given f, e pair, we can also compute the most likely alignment

$$a^* = \text{argmax}_a\, p(a \mid f, e, m)$$

- Nowadays, these IBM models are rarely used for translation, but are used for recovering alignments

Example Alignment

French:

le conseil a rendu son avis , et nous devons à présent adopter un nouvel avis sur la base de la première position .

English:

the council has stated its position , and now , on the basis of the first position , we again have to give our opinion .

Alignment:

the/le council/conseil has/à stated/rendu its/son position/avis ,/, and/et now/présent ,/NULL on/sur the/le basis/base of/de the/la first/première position/position ,/NULL we/nous again/NULL have/devons to/a give/adopter our/nouvel opinion/avis ./.

Alignment from *English* to *French*

IBM Model 1: Alignments

- Recall: $p(f, a|e, m) = p(a|e, m)\, p(f|a, e, m)$

- In IBM Model 1, all alignments are equally likely:

$$p(a|e, m) = \frac{1}{(l+1)^m}$$

- This is a major simplifying assumption …

IBM Model 1: Translation Probabilities

- Recall: $p(f, a|e, m) = p(a|e, m)\, p(f|a, e, m)$

- In IBM Model 1, this is:

$$p(f|a, e, m) = \prod_{i=1}^{m} t(f_i|e_{a_i})$$

Example: e = the dog eats

f = le chien mange

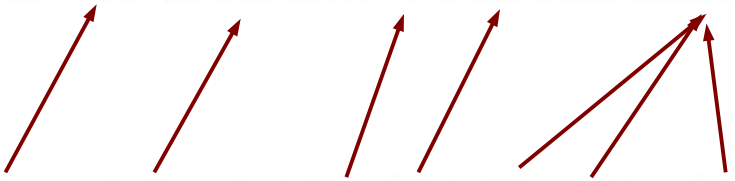
m = 3, a = {1, 2, 3}

$$p(f|a, e, m) = t(\text{le}|\text{the}) \times t(\text{chien}|\text{dog}) \times t(\text{mange}|\text{eats})$$

Another Example

e.g., $l = 6$, $m = 7$

$e =$ And the program has been implemented



$f =$ Le programme a ete mis en application

$a = \{2, 3, 4, 5, 6, 6, 6\}$

$$
\begin{aligned}
p(f \mid a, e, m) = \; & t(Le \mid the) \times \\
& t(programme \mid program) \times \\
& t(a \mid has) \times \\
& t(ete \mid been) \times \\
& t(mis \mid implemented) \times \\
& t(en \mid implemented) \times \\
& t(application \mid implemented)
\end{aligned}
$$

IBM Model 1: The Generative Process

To generate a French string *f* from an English string *e*

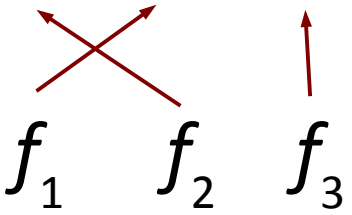
- *Step 1*: Pick an alignment with probability

$$p(a|e,m)=\frac{1}{(l+1)^m}$$

- *Step 2*: Given the alignment, pick the french words with probability

$$p(f|a,e,m)=\prod_{i=1}^{m} t(f_i|e_{a_i})$$

Example: the dog eats



generate f_1 from $t(\text{-}|\text{dog})$, f_2 from $t(\text{-}|\text{the})$ … etc

IBM Model 1: The Generative Process

To generate a French string *f* from an English string *e*

- *Step 1*: Pick an alignment with probability

$$p(a|e,m)=\frac{1}{(l+1)^m}$$

- *Step 2*: Given the alignment, pick the french words with probability

$$p(f|a,e,m)=\prod_{i=1}^{m} t(f_i|e_{a_i})$$

The final result for IBM Model 1:

$$p(f,a|e,m)=p(a|e,m)\,p(f|a,e,m)=\frac{1}{(l+1)^m}\prod_{i=1}^{m} t(f_i|e_{a_i})$$

An Example Lexical Entry

English	French	Probability
position	position	0.756715
position	situation	0.0547918
position	mesure	0.0281663
position	vue	0.0169303
position	point	0.0124795
position	attitude	0.0108907

$t(\text{-} \mid \text{position})$

. . . de la situation au niveau des négociations de l ' ompi . . .
. . . of the current position in the wipo negotiations . . .

nous ne sommes pas en mesure de décider , . . .
we are not in a position to decide , . . .

. . . le point de vue de la commission face à ce problème complexe .
. . . the commission 's position on this complex problem .

IBM Model 2

- Only difference: *alignment* or *distortion* parameters

$$q(j|i,l,m)$$

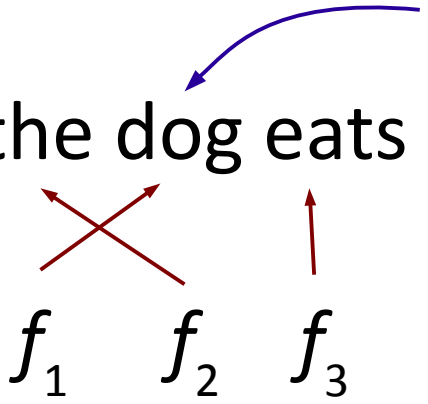
Probability that i^{th} French word is connected to j^{th} English word, given sentence lengths of e and f are l and m

- Define $$p(a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)$$

where $a = \{a_1, \ldots, a_m\}$

- Example:

$$a=\{2,1,3\}$$



$$q(a_1=2|i=1,l=3,m=3)$$

$$p(a|e,m)=q(2|1,3,3)\times q(1|2,3,3)\times q(3|3,3,3)$$

IBM Model 2

- Only difference: *alignment* or *distortion* parameters

$$q(j|i,l,m)$$

Probability that j^{th} French word is connected to i^{th} English word, given sentence lengths of e and f are l and m

- Define $$p(a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)$$

where $a = \{a_1, \ldots, a_m\}$

- The final result for IBM Model 2:

$$p(f,a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)\,t(f_i|e_{a_i})$$

Another Example

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

$$
\begin{aligned}
p(a \mid e, 7) = \; & \mathbf{q}(2 \mid 1, 6, 7) \times \\
& \mathbf{q}(3 \mid 2, 6, 7) \times \\
& \mathbf{q}(4 \mid 3, 6, 7) \times \\
& \mathbf{q}(5 \mid 4, 6, 7) \times \\
& \mathbf{q}(6 \mid 5, 6, 7) \times \\
& \mathbf{q}(6 \mid 6, 6, 7) \times \\
& \mathbf{q}(6 \mid 7, 6, 7)
\end{aligned}
$$

Another Example

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

$$
\begin{aligned}
p(f \mid a, e, 7) = \; & \mathbf{t}(Le \mid the) \times \\
& \mathbf{t}(programme \mid program) \times \\
& \mathbf{t}(a \mid has) \times \\
& \mathbf{t}(ete \mid been) \times \\
& \mathbf{t}(mis \mid implemented) \times \\
& \mathbf{t}(en \mid implemented) \times \\
& \mathbf{t}(application \mid implemented)
\end{aligned}
$$

IBM Model 2: The Generative Process

To generate a French string *f* from an English string *e*

- *Step 1*: Pick an alignment with probability

$$p(a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)$$

- *Step 2*: Given the alignment, pick the french words with probability

$$p(f|a,e,m)=\prod_{i=1}^{m} t(f_i|e_{a_i})$$

The final result:

$$p(f,a|e,m)=p(a|e,m)\,p(f|a,e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)\,t(f_i|e_{a_i})$$

Recovering Alignments

- If we have estimates for the parameters *q* and *t*, we can easily recover the most likely alignment for any sentence pair

- Given a sentence pair $e_1, e_2, \ldots, e_l$ and $f_1, \ldots, f_m$, define

$$a_i = \text{argmax}_{a \in \{0,\ldots,l\}} q(a|i,l,m)\, t(f_i|e_a)$$

for $i = 1, \ldots, m$

Recovering Alignments

$$a_i = \text{argmax}_{a \in \{0,\ldots,l\}} q(a|i,l,m)\, t(f_i|e_a)$$

e = And the program has been implemented

f = Le programme a ete mis en application

Focus on computing a_3

NULL:	$q(0\|3, 6, 7)\ t(\text{a} \mid \text{NULL})$
And:	$q(1\|3, 6, 7)\ t(\text{a} \mid \text{And})$
the:	$q(2\|3, 6, 7)\ t(\text{a} \mid \text{the})$
program:	$q(3\|3, 6, 7)\ t(\text{a} \mid \text{program})$
has:	$q(4\|3, 6, 7)\ t(\text{a} \mid \text{has})$
been:	$q(5\|3, 6, 7)\ t(\text{a} \mid \text{been})$
implemented:	$q(6\|3, 6, 7)\ t(\text{a} \mid \text{implemented})$

Choose as a_3 the best value

EM Training

- Till now we saw IBM Models 1 & 2
- The models need the parameters *t* and *q*
- Using the parameters, we can find the “best” alignment
- Now, how do we get these parameters?

The Parameter Estimation Problem

- Input: to the parameter estimation algorithm $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- Output: parameters $t(f|e)$ and $q(j \mid i, l, m)$
- Challenge: we do not have alignments on our training examples

- For example:

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

Parameter Estimation if Alignments are Observed

- If alignments are observed in the training data:

  $e^{(100)}$ = And the program has been implemented

  $f^{(100)}$ = Le programme a ete mis en application

  $a^{(100)}$ = {2, 3, 4, 5, 6, 6, 6}

- Training data is $(e^{(k)}, f^{(k)}, a^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence, each $a^{(k)}$ is the alignment.

- Maximum likelihood parameter estimation in this case is easy:

$$t_{ML}(f|e) = \frac{\text{Count}(e, f)}{\text{Count}(e)} \qquad q_{ML}(j|i, l, m) = \frac{\text{Count}(j|i, l, m)}{\text{Count}(i, l, m)}$$

Parameter Estimation if Alignments are Observed

- Maximum likelihood parameter estimation in this case is easy:

$$t_{ML}(f|e) = \frac{\text{Count}(e, f)}{\text{Count}(e)} \qquad q_{ML}(j|i, l, m) = \frac{\text{Count}(j|i, l, m)}{\text{Count}(i, l, m)}$$

- Example:

$$t_{ML}(\text{le} \mid \text{the}) = \frac{\text{Count}(\text{le, the})}{\text{Count}(\text{the})}$$

Number of times "the" and "le" were aligned

Number of times "the" was aligned to anything

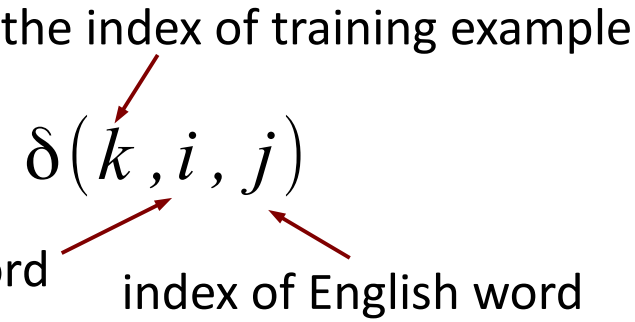
$$q_{ML}(3|1,6,7) = \frac{\text{Count}(3|1,6,7)}{\text{Count}(1,6,7)}$$

Number of times position 1 (in French) was aligned with position 3 (in English) for l = 6 and m = 7

Number of times position 1 (in French) was aligned with anything for l = 6 and m = 7

Algorithm

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.



Algorithm:

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$,

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output: $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i, l, m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Algorithm

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

Algorithm:

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$,

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output: $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i, l, m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Example 90:

$e^{(90)}$ = the dog

$f^{(90)}$ = le chien

$a^{(90)}$ = {1, 2}

δ(90, 1, 1) = 1

δ(90, 2, 2) = 1

δ(90, *i*, *j*) = 0

c(the, le) ++

c(the) ++

c(1|1,2,2) ++

c(1,2,2) ++

c(dog, chien) ++

c(dog) ++

c(2|2,2,2) ++

c(2,2,2) ++

Parameter Estimation with the EM Algorithm

- The alignments are not observed in the training data:

  $e^{(100)}$ = And the program has been implemented

  $f^{(100)}$ = Le programme a ete mis en application

- Training data is $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence

- Related to previous algorithm, but with two key differences:
  - The algorithm is *iterative*. Start with some initial choice for *q* and *t*. At each iteration, compute some "counts" based on the data and current estimates. Re-estimate the parameters using the new counts
  - We use the following definition for $\delta(k, i, j)$ at each iteration:

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k)\, t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k)\, t(f_i^{(k)}|e_j^{(k)})}$$

EM Algorithm

Input: A training corpus $(f^{(k)}, e^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$.

Initialization: Initialize $t(f|e)$ and $q(j|i, l, m)$ parameters (e.g., to random values).

10 – 20 iterations

EM Algorithm

For $s = 1 \ldots S$

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

Identical to previous algorithm

where

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}$$

- Recalculate the parameters:

$$t(f|e) = \frac{c(e, f)}{c(e)} \qquad q(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}$$

EM Algorithm

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k)\, t(f_i^{(k)} | e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k)\, t(f_i^{(k)} | e_j^{(k)})}$$

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

$\delta(100, 3, 0) = q(0|3,6,7) \times t(\text{a} \mid \text{NULL}) / X$

$X = (q(0|3,6,7) \times t(\text{a} \mid \text{NULL}) + q(1|3,6,7) \times t(\text{a} \mid \text{And}) + q(2|3,6,7) \times t(\text{a} \mid \text{the}) + \ldots)$

$\delta(100, 3, 1) = q(1|3,6,7) \times t(\text{a} \mid \text{And}) / X$

$\delta(100, 3, 2) = q(2|3,6,7) \times t(\text{a} \mid \text{the}) / X$

…

$\delta(100, 3, 6) = q(6|3,6,7) \times t(\text{a} \mid \text{implemented}) / X$

EM Algorithm

$$\delta(k,i,j)=\frac{q(j|i,l_k,m_k)\,t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)\,t(f_i^{(k)}|e_j^{(k)})}$$

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

$\sum_{j=0}^{l_k} \delta(k,i,j)=1$ They form a probability distribution over *j*

$$\delta(k,i,j)=P(a_i^{(k)}=j|e^{(k)},f^{(k)};t,q)$$

Probability that i^{th} French word is aligned with j^{th} English word under the current estimation parameter values

So we are trying to estimate the "best" alignment and use that because we don't have the actual alignment

EM Algorithm

For $s = 1 \ldots S$

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

where

$$
\delta(k, i, j) = \frac{q(j|i, l_k, m_k)\, t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k)\, t(f_i^{(k)}|e_j^{(k)})}
$$

- Recalculate the parameters:

$$
t(f|e) = \frac{c(e, f)}{c(e)} \qquad q(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}
$$

Random (q, t) values

⇩

Compute Counts

⇩

Re-estimate (q, t)

⇩

Compute Counts

⇩

Re-estimate (q, t)

⇩

…

Justification for the EM Algorithm

- Training data is $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- The log-likelihood function

$$L(q, t) = \sum_{k=1}^{n} \log p(f^{(k)} | e^{(k)}) = \sum_{k=1}^{n} \log \sum_{a} p(f^{(k)}, a | e^{(k)})$$

- The maximum likelihood estimates:

$$\text{argmax}_{q,t} L(q, t)$$

- The EM algorithm converges to a *local* maximum of the likelihood function (it is not *convex*)

Summary

- Use alignments to simplify model
- Once parameters are estimated, we can recover the most probable alignment
- Iterative EM algorithm for estimating parameters
- IBM Model 2 no longer used for translation, but rather for recovering alignments, which are used in other MT methods

Recap

- Introduction
- Challenges of Machine Translation (MT)
- Classical Approaches
  - Direct MT
  - Transfer Based MT
  - Interlingua-Based MT
- Introduction to Statistical Machine Translation (SMT)
- IBM Models
- Training of IBM Models