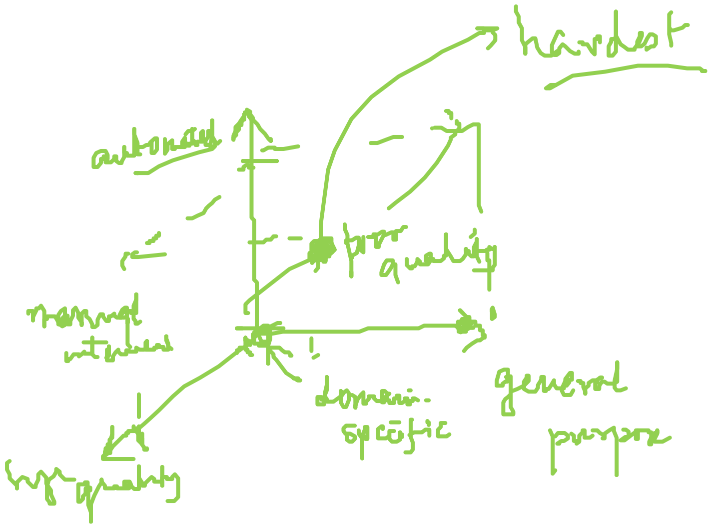
Machine Translation

Some slides are based on Kevin Knight's "A Statistical MT Tutorial Workbook"

Machine Translation



- MT is useful
  - Overcoming the digital divide
  - An ~~imaginary~~ application : MT interface in your cell phone camera

- MT is Hard
  - Limited successes in restricted domains

- Three goals : generating general-purpose, automatic, high quality translations

Problems in Machine Translation

- Word Order
- Word Sense
- Pronoun Resolution
- Idioms
- Ambiguity

English to Russian

"The spirit is willing but the flesh is weak"

“The vodka is good, but the meat is rotten”

Characteristics of Indian Languages

- Subject-Object-Verb
- Relatively Free Word Order
- Morphological change based on number and gender
- Post-position markers instead of prepositions
- Pronouns have no gender information
- Verb complexes – tense, gender information

MT Approaches

- Direct MT
- Rule based translation
- Corpus-based translation
- Knowledge-based translation

Direct MT

- No intermediate representation
- Steps :
  - Remove morphological inflections to get roots
  - Look up bilingual dictionary to get target language words
  - Change the word order to match target language order
- Limitations
  - No. of translators
  - Quality of translation

Morphology → Inflections

Morphology → Derivations

go → going

educate → education

Finite State Transducer

XFST

Rule Based Translation

- Two kinds
  - Transfer based
  - Interlingua
- Transfer based
  - Two steps:
    - Structural transfer (parse tree rewrite)
    - Target language lexicalization
  - Modular, can handle ambiguities
- Interlingua
  - Less no. of components
  - Defining an interlingua may be hard

Interlingua

m n

$O(mn)$

$(m+n)$

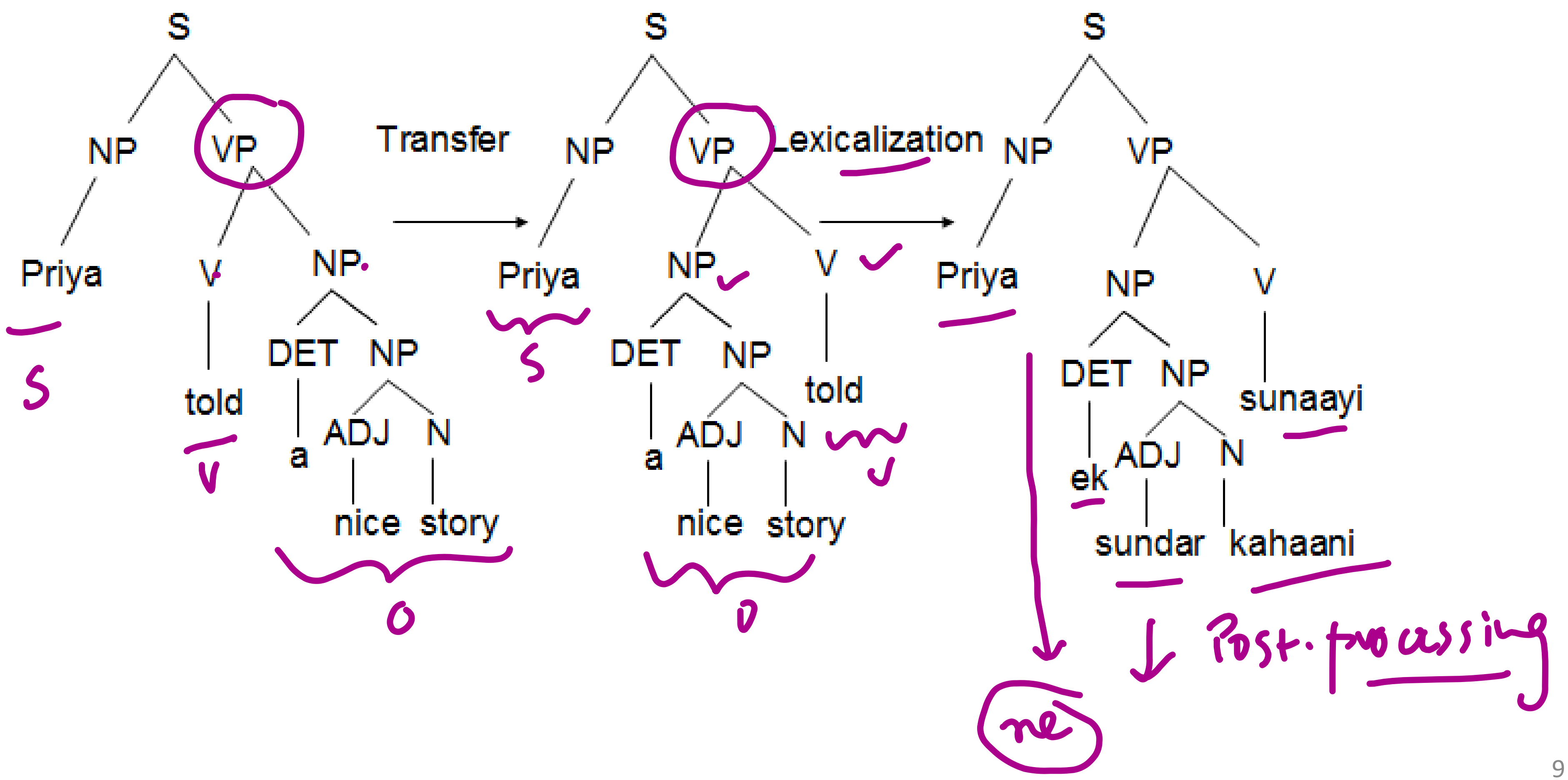
Inter-lingua

$O(2m)$

UNL

Universal Network Language

Transfer based Translation



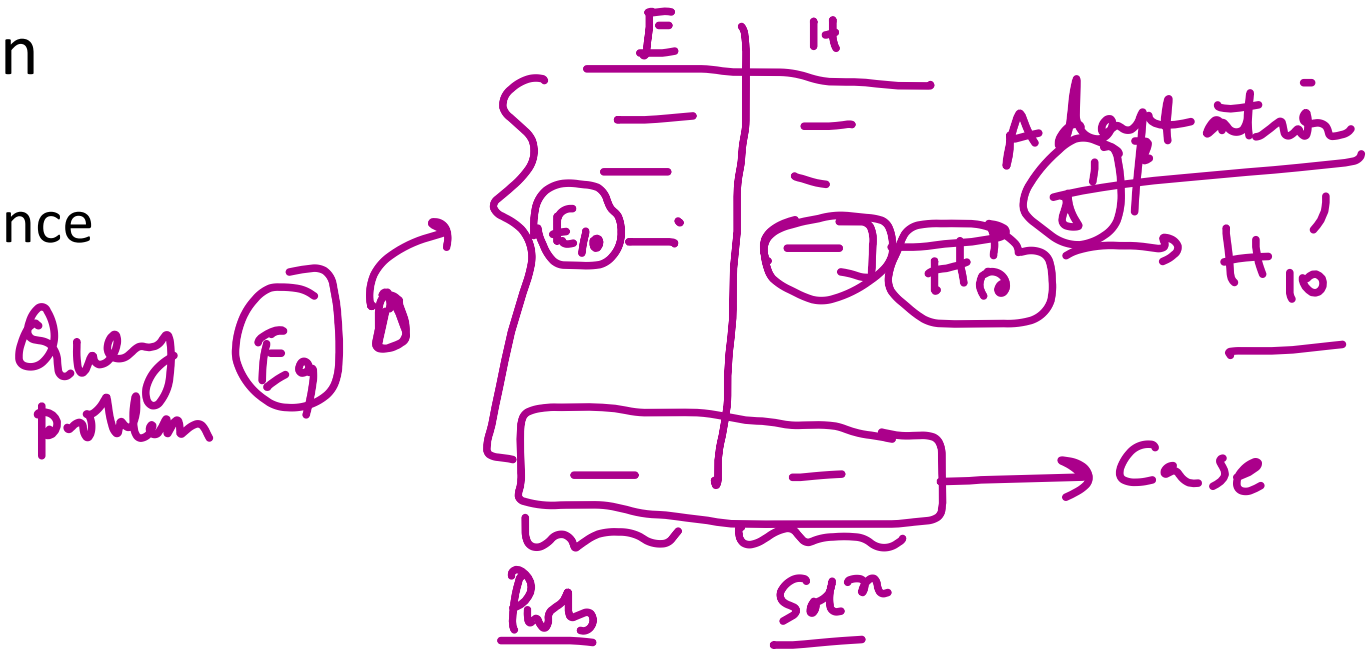
Corpus Based Translation

Model-Based
Rule-Based
Case-Based

- Statistical Machine Translation
  - Three steps in translating from English to Hindi:
    - Estimate language model P(h)
    - Estimate translation language model P(e/h)
    - Devise an efficient search for Hindi text that maximizes their product

Case Based Reasoning

- Example-based Translation
  - Retrieval + Adaptation
  - Catch : translation divergence



He eats shoots & leaves.

Translation involving Indian Languages

- AnglaBharati
  - Pseudo-interlingua + examples + post-editing
- Shakti
  - transfer-based
- MaTra
  - Frame representations
- MANTRA (Machine Assisted Translation Tool)
- Anusaarak

Bayesian stuff again…

$\boxed{F \rightarrow E}$

- Given a French sentence f, we seek the English sentence e that maximizes P(e|f).

$$\underbrace{\operatorname*{argmax}_e \; P(e \mid f)} = \operatorname*{argmax}_e \; P(e) * P(f \mid e)$$

Source model   Channel model

language model

diseases

translation model

F | E

E_1, E_2, …, E_n → NOISY CHANNEL → F

generative

observat

Symptom

B.I.

Inferencing

The Noisy Channel Idea

Why not use P(e|f) directly?

If we reason directly about translation using P(e|f), then our probability estimates had better be very good.

On the other hand, if we break things apart using Bayes Rule, then we can theoretically get good translations even if the probability numbers aren't that accurate.

The factor P(f|e) will ensure that a good e will have words that generally translate to words in f.

- Various "English" sentences will pass this test. For example, if the string "the boy runs" passes, then "runs boy the" will also pass. Some word orders will be grammatical and some will not.
- However, the factor P(e) will lower the score of ungrammatical sentences.

Goal : F→ E

Word Choice in Translation

The P(e) model can also be useful for selecting English translations of French words.

Example :

- suppose there is a French word that either translates as “in” or “on.” Then there may be two English strings with equally good P(f | e) scores: (1) she is in the end zone, (2) she is on the end zone. (Let's ignore other strings like “zone end the in is she” which probably also get good P(f | e) scores).

- the first sentence is much better English than the second, so it should get a better P(e) score, and therefore a better P(e) * P(f | e) score.

Goal : F→ E

P(e) : Language Model

Bigram Model

p(y | x) = number-of-occurrences(“xy”) / number-of-occurrences(“x”)

P(I like snakes that are not poisonous) ~
p(I | start-of-sentence) *
p(like | I) *
p(snakes | like) *
...
p(poisonous | not) *
p(end-of-sentence | poisonous)

Trigram Model

p(z | x y) = number-of-occurrences(“xyz”) / number-of-occurrences(“xy”)

P(I like snakes that are not poisonous) ~
p(I | start-of-sentence start-of-sentence) *
p(like | start-of-sentence I) *
p(snakes | I like) *
...
p(poisonous | are not) *
p(end-of-sentence | not poisonous) *
p(poisonous | end-of-sentence end-of-sentence)

Smoothing

Instead of

p(z | x y) = number-of-occurrences("xyz") / number-of-occurrences("xy")

we can use

p(z | x y) = 0.95 * number-of-occurrences("xyz") / number-of-occurrences("xy") +
0.04 * number-of-occurrences ("yz") / number-of-occurrences ("z") +
0.008 * number-of-occurrences("z") / total-words-seen +
0.002

It's handy to use different <u>smoothing coefficients</u> in different situations. You might want 0.95 in the case of xy(z), but 0.85 in another case like ab(c). For example, if "ab" doesn't occur very much, then the counts of "ab" and "abc" might not be very reliable.

Goal : F→ E

Translation Model

- Storyline:

  - Words in an English sentence are replaced by French words, which are then scrambled around.

  - P(f | e) doesn't necessarily have to turn English into good French. Some of the slack will be taken up by the independently-trained P(e) model.

Goal : F→ E

Example

- Mary did not slap the green witch (input)

Goal : F→ E

Example

- Mary did not slap the green witch (input)
- Mary not slap slap slap the green witch (choose fertilities)

Goal : F→ E

Example

- Mary did not slap the green witch (input)
- Mary not slap slap slap the green witch (choose fertilities)
- Mary not slap slap slap NULL the green witch (choose number of spurious words)

Goal : F→ E

Example

- Mary did not slap the green witch (input)
- Mary not slap slap slap the green witch (choose fertilities)
- Mary not slap slap slap NULL the green witch (choose number of spurious words)
- Mary no daba una botefada a la verde bruja (choose translations)

Example

- Mary did not slap the green witch (input)
- Mary not slap slap slap the green witch (choose fertilities)
- Mary not slap slap slap NULL the green witch (choose number of spurious words)
- Mary no daba una botefada a la verde bruja (choose translations)
- Mary no daba una botefada a la bruja verde (choose target positions)

Goal : F→ E

Example

- Mary did not slap the green witch (input)
- Mary not slap slap slap the green witch (choose fertilities)
- Mary not slap slap slap NULL the green witch (choose number of spurious words)
- Mary no daba una botefada a la verde bruja (choose translations)
- Mary no daba una botefada a la bruja verde (choose target positions)

Goal : F→ E

Reference slide: Parameters

- Parameters like t(daba | slap), are translation probabilities, which gives the probability of producing “daba” from “slap”
- Fertility parameters like n(1 | house), which gives the probability that “house” will produce exactly one French word, whenever “house” appears.
- Distortion parameters like d(5 | 2) which gives the probability that an English word in position 2 (of an English sentence) will generate a French word in position 5 (of a French translation).

In practice, a richer distortion parameter like d(5 | 2, 4, 6) is used in IBM Model 3, which is just like d(5 | 2), except also given that the English sentence has four words and French sentence has six words. Also, an additional set of parameters may be needed to capture the fact that a French word may appear out of nowhere, i.e. when there is no corresponding English word.

Goal : F→ E

IBM Model 3 parameters

- The model has four types of parameters: n, t, p, and d.

Goal : F→ E

Word-for-word Alignments

- First, how can we automatically obtain parameter values from data? Second, armed with a set of parameter values, how can we compute P(f | e) for any pair of sentences?

- First let's think about automatically obtaining values for the n, t, p, and d parameters from data. If we had a bunch of English strings and a bunch of step-by-step rewritings into French, then life would be easy.
  - To compute n(0 | did), we would just locate every instance of the word "did" and see what happens to it during the first rewriting step. If "did" appeared 15,000 times and was deleted during the first rewriting step 13,000 times, then n(0 | did) = 13/15.

Goal : F→ E

Word-for-word Alignments

```
NULL   And   the   program   has   been   implemented
  |     |     |       |        |     |          |
              |       |        |     |       +-+---+
              |       |        |     |       | |   |
             Le programme      a    ete    mis en application
```

To compute t(maison | house), we count up all the French words generated by all the occurrences of “house,” and see how many of those words are “maison.”

Every French word is connected to exactly one English word (either a regular word or NULL). This is not intuitive.

We can represent the sample word alignment above as [2, 3, 4, 5, 6, 6, 6].

Goal : F→ E

EM for bootstrapping

- To get good parameter value estimates, we may need a very large corpus of translated sentences.
- Such large corpora do exist, sometimes, but they do not come with word-for-word alignments. However, it is possible to obtain estimates from non-aligned sentence pairs.

EM for parameter estimation in SMT

$$P(a \mid \underbrace{e, f}) = \frac{P(a, f \mid e)}{P(f \mid e)}$$

obtained after preparation step of E

$$\frac{P(a, e, f)}{P(e, f)} = \frac{P(a, f \mid e) \cdot P(e)}{P(f \mid e) \cdot P(e)}$$

$$= \frac{P(a, f \mid e)}{P(f \mid e)} = \frac{P(a, f \mid e)}{\sum_a P(a, f \mid e)}$$

for colouring

colouring ↔ assignment

Example

Step 1 : M0

$t(x|b) = 1/2$

$t(y|b) = 1/2$

$t(x|c) = 1/2$

$t(y|c) = 1/2$

English French

$b\ c \longleftrightarrow x\ y$

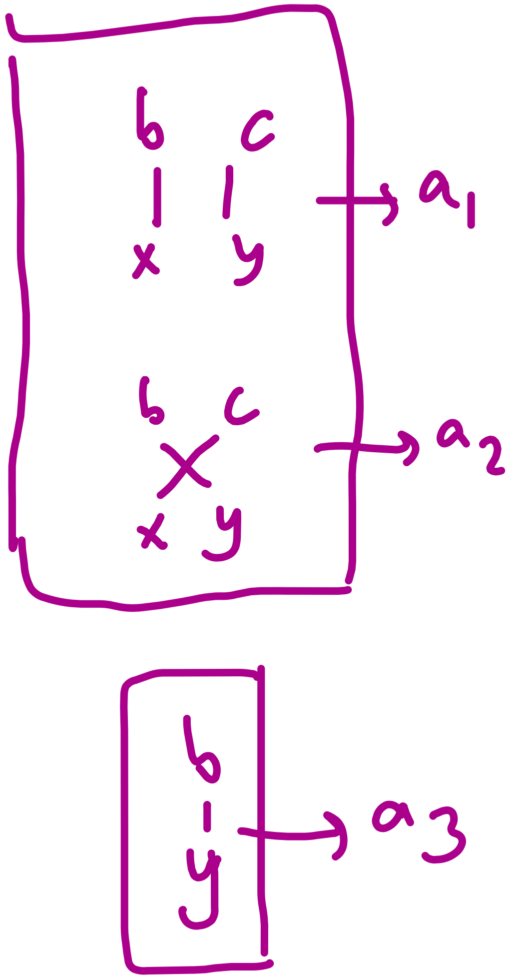
$b \longleftrightarrow y$

Assumptions:

→ No nulls.

→ Each word has fertility 1

Step 2: E_1 : preparation step.



$$P(a_1, f \mid e) = \underbrace{\frac{1}{2} \times \frac{1}{2}}_{} = \frac{1}{4}$$

$$P(a_2, f \mid e) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

$$P(a_3, f \mid e) = \frac{1}{2}$$

Step 3: E_1 Colouring step

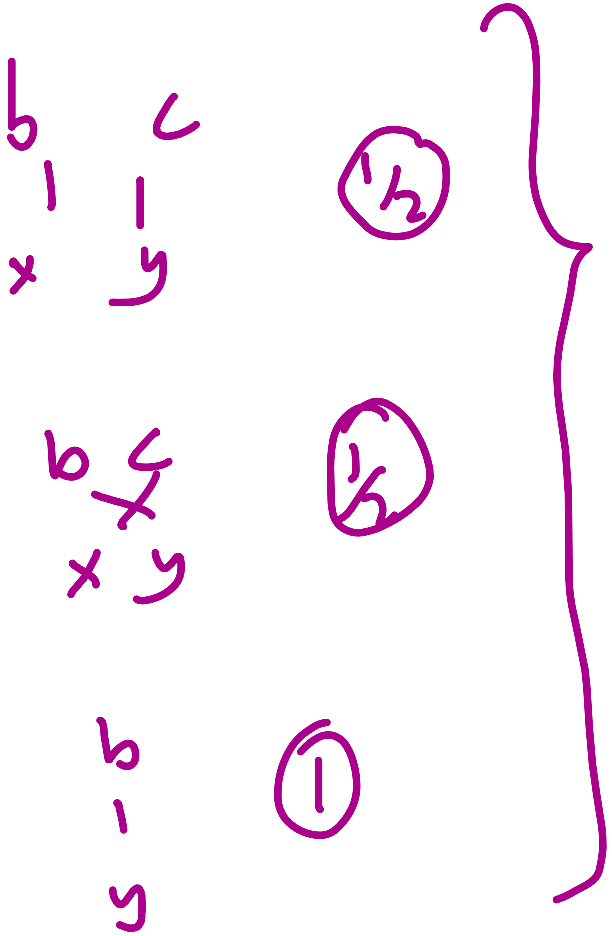
$$p(a_1|f,e) = \frac{p(a_1,f|e)}{p(a_1,f|e) + p(a_2,f|e)} = \frac{1/4}{1/4 + 1/4} = 1/2$$

$$p(a_2|f,e) = \frac{p(a_2,f|e)}{p(a_1,f|e) + p(a_2,f|e)} = \frac{1/4}{1/4 + 1/4} = 1/2$$

$$p(a_3|f,e) = 1.$$

Step 4:

M, step



$$t(x|b) = \frac{1/2}{1/2 + \frac{1}{2} + 1} = \frac{1}{4}$$

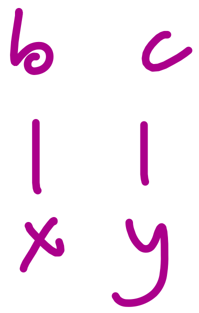
$$t(y|b) = \frac{1 + 1/2}{\frac{1}{2} + \frac{1}{2} + 1} = 3/4$$

$$t(x|c) = \frac{1/2}{1/2 + 1/2} = 1/2$$

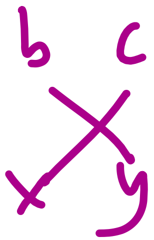
$$t(x|b) = 1/2$$

Step 5

E step : preparation.



$$P(a_1, f|e) = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$$



$$p(a_2, f|e) = \frac{3}{4} \times \frac{1}{2} = \frac{3}{8}$$

$$P(a_3, f|e) = \frac{3}{4}$$

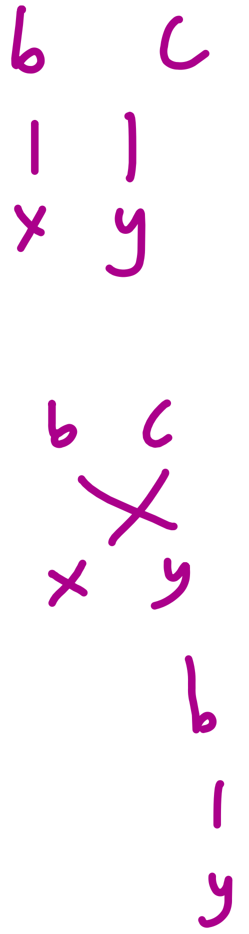
Step 6 : E_2 : colouring

$$p(a_1 | e, f) = \frac{1/8}{1/8 + 3/8} = 1/4$$

$$p(a_2 | e, f) = \frac{3/8}{1/8 + 3/8} = 3/4$$

$$p(a_3 | e, f) = 1$$

Step 7: M_2 Step.



$\frac{1}{4}$

$\frac{3}{4}$

1

$$t(x|b) = \frac{1/4}{1/4 + 3/4 + 1} = 1/8$$

$$t(y|b) = \frac{3/4 + 1}{1/4 + 3/4 + 1} = \frac{7}{8}$$

$$t(x|c) = \frac{3/4}{1/4 + 3/4} = 3/4$$

$$t(y|c) = \frac{1/4}{1/4 + 3/4} = 1/4$$

After a few more iterations

$$\begin{cases} t(x|b) = 0.0001 \\ t(y|b) = 0.9999 \\ t(x|c) = 0.9999 \\ t(y|c) = 0.0001 \end{cases}$$

$$\begin{array}{ccc} \underline{F} & & E \\ b\ c & \leftrightarrow & x\ y \\ b & \leftrightarrow & y \end{array}$$

Generative process in preparation for E step

Instead of alignment weights, we will start thinking in terms of alignment probabilities.

$p(a \mid e, f)$ = prob. of a particular alignment given a particular sentence pair.

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

Proof: $$p(a \mid e,f) = \frac{p(a,e,f)}{p(e,f)} = \frac{P(a,f \mid e) \, . \, P(e)}{p(f \mid e) . P(e)}$$

$$= \frac{p(a,f \mid e)}{P(f \mid e)} = \frac{p(a,f \mid e)}{\sum_a p(a,f \mid e)}$$

Example: M_0 step

Corpus has two sentence pairs :

English		French
b c	$\longleftrightarrow$	x y
b	$\longleftrightarrow$	y

Step 1 (M_0)

$t(x|b) = 1/2$

$t(y|b) = 1/2$

$t(x|c) = 1/2$

$t(y|c) = 1/2$

Assumptions:

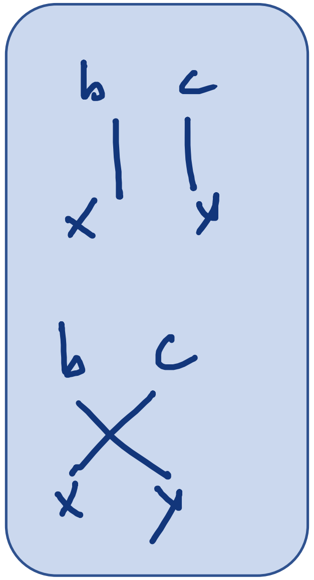
- NO NULL
- Every word has fertility 1

Example: preparation for E_1 step

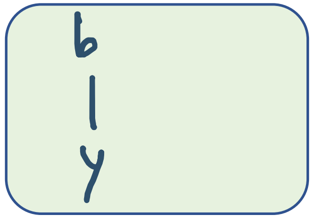
Step 2

Compute $P(a, f \mid e)$ for all alignments

Two alignments corresponding to ambiguous pair bc <-> xy



Only alignment Corresponding to unambiguous pair b <-> y



$$P(a_1, f \mid e) = \tfrac{1}{2} \times \tfrac{1}{2} = \tfrac{1}{4}$$

$$p(a_2, f \mid e) = \tfrac{1}{2} \times \tfrac{1}{2} = \tfrac{1}{4}$$

$$p(a_3, f \mid e) = \tfrac{1}{2}$$

Similar to coin tossing example where we estimated

$$(\hat{\theta}_A)^m (1-\hat{\theta}_A)^{N-m} = \alpha$$

and

$$(\hat{\theta}_B)^m (1-\hat{\theta}_B)^{N-m} = \beta$$

Example: E_1 step

Step 3

$$p(a_1 \mid f, e) = \frac{p(a_1, f \mid e)}{p(a_1, f \mid e) + p(a_2, f \mid e)} = \frac{1/4}{1/4 + 1/4} = 1/2$$

$$p(a_2 \mid f, e) = \frac{p(a_2, f \mid e)}{p(a_1, f \mid e) + p(a_2, f \mid e)} = \frac{1/4}{1/4 + 1/4} = 1/2$$

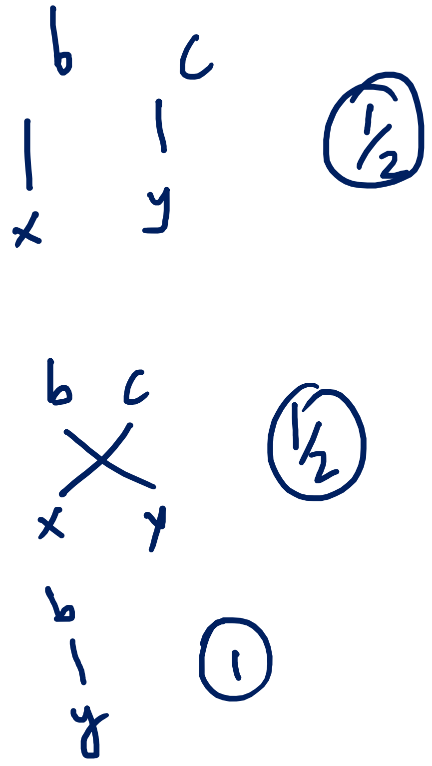
$$p(a_3 \mid f, e) = 1$$

Coin tossing example again!!!

Example: M_1 step

Step 4

Collect fractional counts to estimate parameters again



$$t(x|b) = \frac{1/2}{1/2 + 1/2 + 1} = 1/4$$

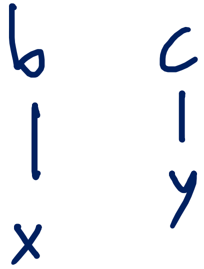
$$t(y|b) = \frac{1 + 1/2}{1/2 + 1/2 + 1} = \frac{3/2}{2} = 3/4$$

$$t(x|c) = \frac{1/2}{1/2 + 1/2} = 1/2$$

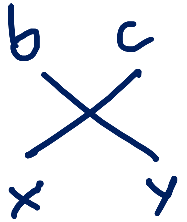
$$t(y|c) = \frac{1/2}{1/2 + 1/2} = 1/2$$

Example: Preparation for E_2 step

Step 5



$$p(a_1, f \mid e) = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$$



$$p(a_2, f \mid e) = \frac{3}{4} \times \frac{1}{2} = \frac{3}{8}$$

$$p(a_3, f \mid e) = \frac{3}{4}.$$

Example: E_2 step

Step 6

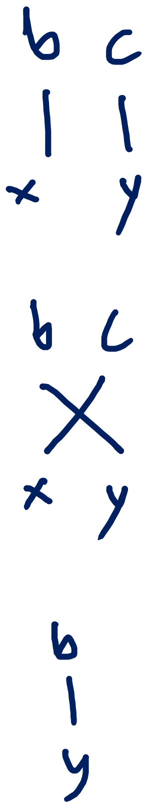
$$p(a_1 | e, f) = \frac{1/8}{1/8 + 3/8} = 1/4$$

$$p(a_2 | e, f) = \frac{3/8}{1/8 + 3/8} = 3/4$$

$$p(a_3 | e, f) = 1$$

Example: M_2 step

Step 7



$1/4$

$3/4$

1

$$t(x|b) = \frac{1/4}{1/4 + 3/4 + 1} = 1/8$$

$$t(y|b) = \frac{3/4 + 1}{1/4 + 3/4 + 1} = \frac{7}{8}$$

$$t(x|c) = \frac{3/4}{3/4 + 1/4} = 3/4$$

$$t(y|c) = \frac{1/4}{3/4 + 1/4} = 1/4$$

After a few more iterations: M step outcome

Repeating these steps for a few more iterations gives us the following estimates (on convergence)

$$t(x|b) = 0.0001$$

$$t(y|b) = 0.9999$$

$$t(x|c) = 0.9999$$

$$t(y|c) = 0.0001$$

English	French
bc $\leftrightarrow$	x y
b $\leftrightarrow$	y

CL Olympiad

```
1a.  ok-voon ororok sprok .

1b.  at-voon bichat dat .
---------------------------------------------------
2a.  ok-drubel ok-voon anok plok sprok .

2b.  at-drubel at-voon pippat rrat dat .
---------------------------------------------------
3a.  erok sprok izok hihok ghirok .

3b.  totat dat arrat vat hilat .
---------------------------------------------------
4a.  ok-voon anok drok brok jok .

4b.  at-voon krat pippat sat lat .
---------------------------------------------------
5a.  wiwok farok izok stok .

5b.  totat jjat quat cat .
---------------------------------------------------
6a.  lalok sprok izok jok stok .

6b.  wat dat krat quat cat .
```

```
---------------------------------------------------
7a.  lalok farok ororok lalok sprok izok enemok .

7b.  wat jjat bichat wat dat vat eneat .
---------------------------------------------------
8a.  lalok brok anok plok nok .

8b.  iat lat pippat rrat nnat .
---------------------------------------------------
9a.  wiwok nok izok kantok ok-yurp .

9b.  totat nnat quat oloat at-yurp .
---------------------------------------------------
10a. lalok mok nok yorok ghirok clok .

10b. wat nnat gat mat bat hilat .
---------------------------------------------------
11a. lalok nok crrrok hihok yorok zanzanok .

11b. wat nnat arrat mat zanzanat .
---------------------------------------------------
12a. lalok rarok nok izok hihok mok .

12b. wat nnat forat arrat vat gat .
---------------------------------------------------
```

```
Translation dictionary:

ghirok - hilat              ok-yurp - at-yurp
ok-drubel - at-drubel       zanzanok - zanzanat
ok-voon - at-voon
```

Why does EM work?

- Ambiguity is resolved by exploiting knowledge from unambiguous mappings

Parallel corpus

TATA

டாடா

Parallel corpus

	K-means	GMM	Biased Coins	PCFG	Statistical MT
Source					
Observations					
Generative storyline (preparation for E step)					
Parameters Estimated					
Why it works					

Homework problem

Consider a Machine Translation parallel corpus having two sentence pairs. The first sentence pair is “Read books”/”Kitaab padho”. The second sentence pair is “read”/”padho”. (a) Show how the first three iterations of EM (starting M_0) useful in learning word alignments from this corpus. (b) Make clear any simplifying assumptions (with respect to IBM Model 3) that you use. (c) Why does EM succeed in resolving ambiguity, in case it does?

Reference Material

A Statistical MT Tutorial Workbook

by Kevin Knight

Appendix: The generative process

e = English sentence
f = French sentence
ei = the ith English word
fj = the jth French word
l = number of words in the English sentence
m = number of words in the French sentence
a = alignment (vector of integers a1 ... am, where each aj ranges from 0 to l)
aj = the English position connected to by the jth French word in alignment a
eaj = the actual English word connected to by the jth French word in alignment a
phi-i = fertility of English word i (where i ranges from 0 to l), given the alignment a

$$P(a,f \mid e) = \prod_{i=1}^{l} n(\text{phi-i} \mid ei) * \prod_{j=1}^{m} t(fj \mid eaj) * \prod_{j=1}^{m} d(j \mid aj, l, m)$$

Stack Decoding algorithm

- https://www.youtube.com/watch?v=oWVmphEaHZI