Machine Translation

[J&M'08 Ch. 25]

Machine Translation

- Anaphora & agreement

(1) a. **The monkey** ate the banana because **it** was hungry.
b. The monkey ate **the banana** because it was ripe.
c. The monkey ate the banana because **it** was tea-time

(2) EN: I put **the book** inside **the suitcase** so that **it** wouldn't get dirty.
PT: Eu pus **o livro** dentro d**a mala** para que **ele**/**ela** não se sujasse.

- Word order & discontinuities
  Japanese: Subject - Object - Verb
  Malagasy: Verb - Object - Subject
  Filipino: Verb - Subject - Object

- Lexical gaps across languages
  Schadenfreude (German): pleasure derived from another's misfortune
  Razbliuto (Russian): feeling one has for someone they used to love

Machine Translation

- Isolating vs. polysyntectic languages

(3) Sahonwanhotnkwahse (Mohawk)
sa-honwa-nhoton-kw-a-hse
again-PAST-she/him-opendoor-reversive-un-for
"she opened the door for him again"

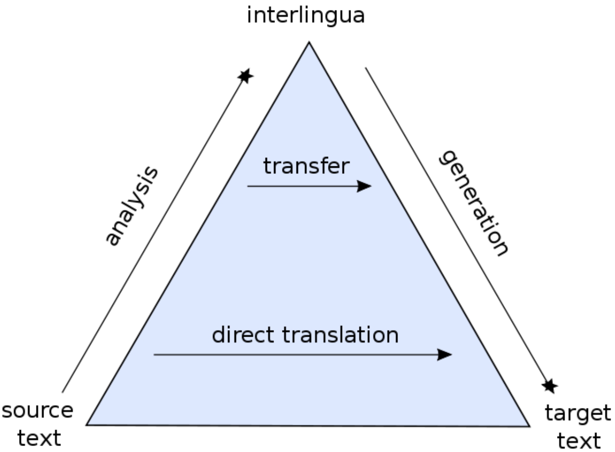
- Pro-drop

(4) Viste? (Portuguese)
'saw'
Did you see that?

- Lexical divergence

(5) a. I know Sam / math.
b. Eu sei matematica.
c. Eu conheo o Sam.

Vauquois Triangle



Machine Translation

Direct Translation typically involves 4 steps:

1. morphological parsing
2. bilingual dictionary
3. reordering
4. morphological generation

Very popular:

- Faster and cheaper to develop.
- Appropriate when rough translation is enough.
- Can be improved by human post-editing (computer-aided human translation)
- Specially good when applied to small sublanguage domains (weather reports, cooking recipes, etc.)

Machine Translation

Example: English to Spanish direct translation

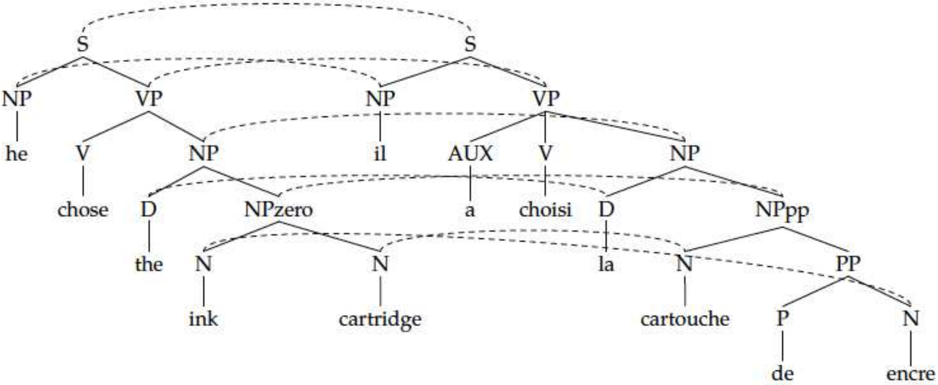
Input:	Mary didn't slap the green witch
Step 1:	Mary DO-PAST not slap the green witch
Step 2:	Maria PAST no dar una bofetada a la verde bruja
Step 3:	Maria no dar PAST una bofetada a la bruja verde
Step 4:	Maria no dió una bofetada a la bruja verde

Machine Translation

function DIRECT_TRANSLATE_MUCH/MANY(word) **returns** Russian translation

if preceding word is *how* **return** *skol'ko*
else if preceding word is *as* **return** *stol'ko zhe*
else if word is *much*
  if preceding word is *very* **return** nil
  else if following word is a noun **return** *mnogo*
else /* word is many */
  if preceding word is a preposition and following word is a noun **return** *mnogii*
  else return *mnogo*

Transfer MT


S
NP
he
VP
V
chose
NP
D
the
NPzero
N
ink
N
cartridge
S
NP
il
VP
AUX
a
V
choisi
NP
D
la
NPpp
N
cartouche
PP
P
de
N
encre

Machine Translation

English to Spanish:		
NP $\rightarrow$ Adjective$_1$ Noun$_2$	$\Rightarrow$	NP $\rightarrow$ Noun$_2$ Adjective$_1$
Chinese to English:		
VP $\rightarrow$ PP[+Goal] V	$\Rightarrow$	VP $\rightarrow$ V PP[+Goal]
English to Japanese:		
VP $\rightarrow$ V NP	$\Rightarrow$	VP $\rightarrow$ NP V
PP $\rightarrow$ P NP	$\Rightarrow$	PP $\rightarrow$ NP P
NP $\rightarrow$ NP$_1$ Rel. Clause$_2$	$\Rightarrow$	NP $\rightarrow$ Rel. Clause$_2$ NP$_1$

Transduction Grammar Transfer

English to Spanish:

$N \rightarrow$ witch / bruja

$NP \rightarrow$ Mary / Maria

$Nominal \rightarrow \langle Adj\ N \rangle$

$Adj \rightarrow$ green / verde

$S \rightarrow [NP\ VP]$

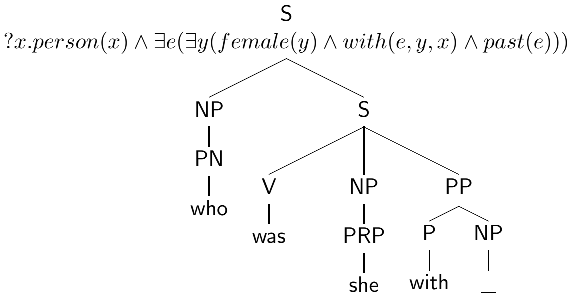
$VP \rightarrow [V\ PP]$

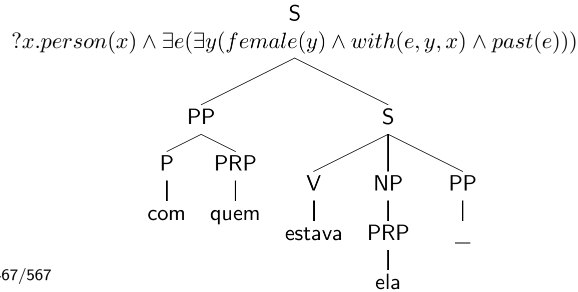
$VP \rightarrow [Negation\ VP]$

$Negation \rightarrow$ didn't / no

$V \rightarrow$ slap / dio una bofetada

Interlingua-based MT





Example:
Verbmobil: a large-scale, multilingual, robust, interlingua machine translation system for spontaneous speech in business domains.

- Bundesministerium für Forschung und Technologie, Germany's Federal Ministry of Research and Technology
- Project duration: 1993 – 2000
- Cooperative effort between industry and science
- Cost: approx. 160 million DM (107 million USD)

Commercial advertisement

Statistical MT (noisy channel model)

$$\hat{E} = \text{arg} \quad \text{max}_{E \in English} P(F|E)P(E)$$

- Where F is a sentence from a foreign language
- E is the corresponding target language sentence

... but this is very difficult, so instead we estimate:

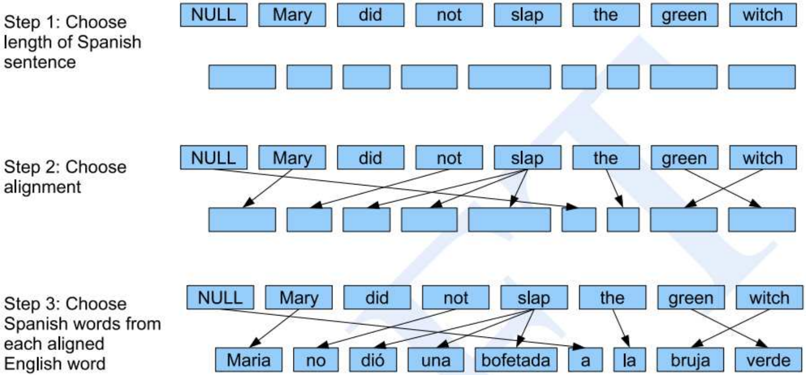
$$\hat{E} = \text{arg} \quad \text{max}_{E \in English} P(F, A|E, m)P(E)$$

- A is an alignment from F to E, eg. $A = 0, 1, 3, 3$ means 'word 1 in F is paired with nothing, word 2 in F is paired with word 1in E, and words 3 and 4 in F are paired with word 3 in E'. For $|F| = n$ and $|E| = m$, there are $(m+1)^n$ alignments.

... and then we can marginalize out A:

$$P(F|E) = \sum_{a \in A} P(F, A|E, m)$$

IBM alignment model



Goal is to find a probability distribution for all alignments.

$$P(\text{la bruja verde}, \langle 1,3,2\rangle | \text{the green witch}, 3)$$

Machine Translation

1. Pick an alignment randomly, say $\langle 1, 3, 2 \rangle$
2. Using a paralel corpus (e.g. EuroParl), pick foreign words according to their translation probability:
   $t(la|the) \times t(bruja|witch) \times t(verde|green)$

$$P(F|A, E, M) = \prod_{i=1}^{n} t(f_i|e_{a_i})$$

f_i is the $i-$th word in F, and e_{a_i} is the english word aligned with it.

IBM Model 2

$$P(F|A,E,M) = \prod_{i=1}^{n} t(f_i|e_{a_i}) \times d(a_i|i,m,n)$$

For example, in *la bruja verde*, $\langle 1,3,2 \rangle$, *the green witch*:

$$d(1|1,3,3)$$

$$d(3|2,3,3)$$

$$d(2|3,3,3)$$

Machine Translation

Once t and d are estimated, it becomes possible to compute the most likely alignments:

$$a_i = \text{arg max}_{a \in A} d(a|i, m, n) \times t(f_i|e_a)$$

How to estimate t and d parameters?

Maximum Likelihood Estimates

Input: set of **aligned** sentence pairs.

$$t(f|e) = \frac{\text{Count}(e, f)}{\text{Count}(e)}$$

$$d(a|i, l, m) = \frac{\text{Count}(a, i, m, n)}{\text{Count}(i, m, n)}$$

... but there aren't many aligned corpora. For those cases, the **Expectation Maximization** algorithm is used.

Expectation Maximization (basic intuition):

1. Randomly initialize t and q
2. Compute (estimate) alignments from t and q (**E-step**)
3. Use alignments to re-estimate t and q (**M-step**)
4. Go to step 2 and repeat for 10 to 20 iterations.

Machine Translation

Tiny paralel corpus:
The witch! / La bruja!
The green witch! / La bruja verde!

$E = \{\text{the, green, witch}\}$
$F = \{\text{la, bruja, verde}\}$

$$t(la|the) = \frac{1}{3} \quad t(la|green) = \frac{1}{3} \quad t(la|witch) = \frac{1}{3} \quad \ldots$$

Step 1: compute $P(a, f|e)$ for all word pairs (let's assume $3! = 6$ possible alignments)

$P(a, f|e) = t(\text{la}|\text{the}) \times t(\text{bruja}|\text{green}) \times t(\text{verde}|\text{witch}) = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$
$P(a, f|e) = t(\text{la}|\text{the}) \times t(\text{bruja}|\text{witch}) \times t(\text{verde}|\text{green}) = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$

...

$$P(A, F|E) = \prod_{i=1}^{m} t(f_i|e_{a_i})$$

Step 2: normalize $P(a, f|e)$ to get $P(a|e, f)$:

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

For example (assuming that there are $3!$ possible alignments)

$$P(a|f,e) = \frac{\frac{1}{27}}{\frac{6}{27}} = 0.16$$

Machine Translation

Step 3: Compute expected counts, weighing each count by $P(a|e,f)$

The witch! / La bruja!
The green witch! / La bruja verde!

$$tcount(\text{la}|\text{the}) = \tfrac{3}{3} \;\; tcount(\text{bruja}|\text{the}) = \tfrac{2}{3} \;\; tcount(\text{verde}|\text{the}) = \tfrac{2}{3} \;\; \text{total(the)} = 2.3$$
$$tcount(\text{la}|\text{green}) = \tfrac{2}{3} \;\; tcount(\text{bruja}|\text{green}) = \tfrac{3}{3} \;\; tcount(\text{verde}|\text{green}) = \tfrac{2}{3} \;\; \text{total(green)} = 2.3$$
$$tcount(\text{la}|\text{witch}) = \tfrac{3}{3} \;\; tcount(\text{bruja}|\text{witch}) = \tfrac{3}{3} \;\; tcount(\text{verde}|\text{witch}) = \tfrac{2}{3} \;\; \text{total(witch)} = 2.3$$

Step 4: normalize the counts (MLE) to sum to 1:

$$t(\text{la}|\text{the}) = \frac{\frac{3}{3}}{2.3} = 0.43 \quad t(\text{bruja}|\text{the}) = \frac{\frac{2}{3}}{2.3} = 0.28 \quad t(\text{verde}|\text{the}) = \frac{\frac{2}{3}}{2.3} = 0.28$$
$$t(\text{la}|\text{green}) = \tfrac{\frac{2}{3}}{2.3} = 0.43 \quad t(\text{bruja}|\text{green}) = \frac{\frac{3}{3}}{2.3} = 0.43 \quad t(\text{verde}|\text{green}) = \frac{\frac{2}{3}}{2.3} = 0.43$$

Step 5: re-compute $P(a|e,f)$ again with the new t probabilities

$$P(a,f|e) = t(\text{la}|\text{the}) \times t(\text{bruja}|\text{green}) \times t(\text{verde}|\text{witch})$$
$$P(a,f|e) = t(\text{la}|\text{the}) \times t(\text{bruja}|\text{witch}) \times t(\text{verde}|\text{green})$$

The HMM approach to finding optimal alignments is similar:

- HMM that aligns Language1 to Language2
- HMM that aligns Language2 to Language1
- Take the intersection of the two aligments

$$P(f_1^J, a_1^J | e_1^I) = P(J|e_1^I) \times \prod_{j=1}^{J} P(f_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \times P(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I)$$

f_1^J = sequence of size J of foreign words
a_1^J = sequence of word alignments of size J
e_1^I = sequence of size I of source language words

Machine Translation

$$P(f_1^J, a_1^J | e_1^I) = P(J|e_1^I) \times \prod_{j=1}^{J} P(f_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \times P(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I)$$

$P(F|A,E)$ is computable from the length probability $P(J|e_1^I)$, and alignment probability $P(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I)$, and a lexicon probability $P(f_j|f_1^{j-1}, a_1^{j-1}, e_1^I)$.

Expectation Maximization algorithm for HMM finds optimal alignment.