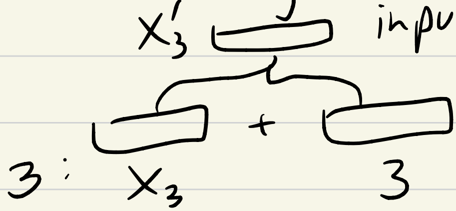
CS 378 Lecture 18: Machine Translation

Announcements

- FP proposals back
- A4 due in 1 week

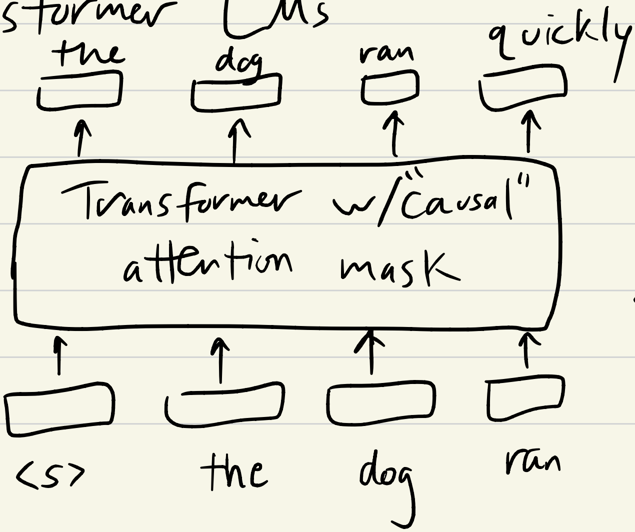
Recap Transformers

Position encoding

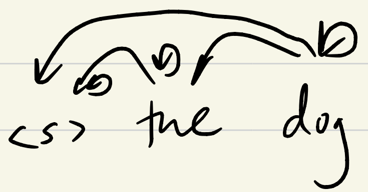


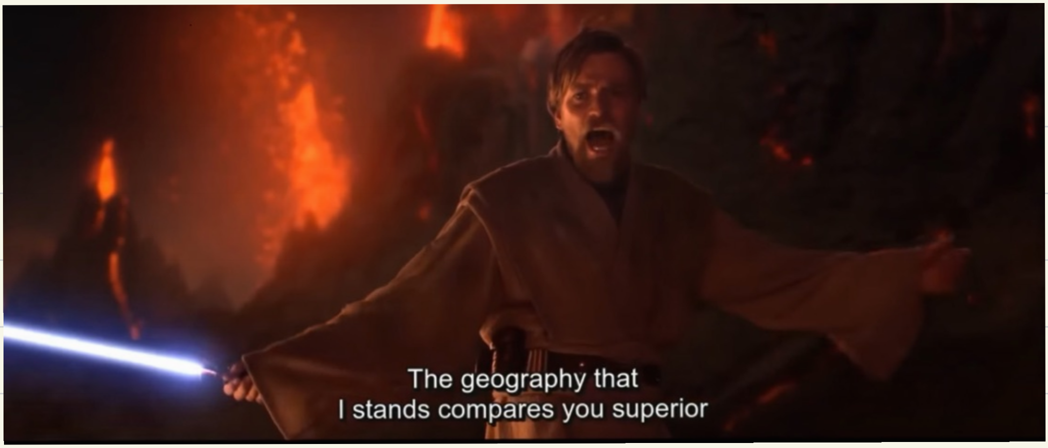
i = posn in sequence

Transformer LMs



causal attn:





Machine Translation

Language modeling: $P(\bar{w})$

MT: $P(\bar{y}|\bar{x})$

Today: phrase-based MT (pre-2015)

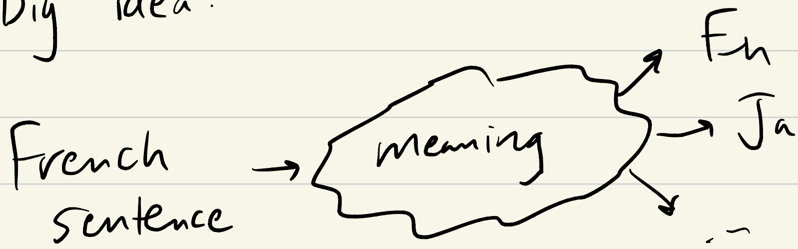
Word alignment: another perspective on attention

Input: $\bar{s}$ source sentence

Output: $\bar{t}$ target sentence

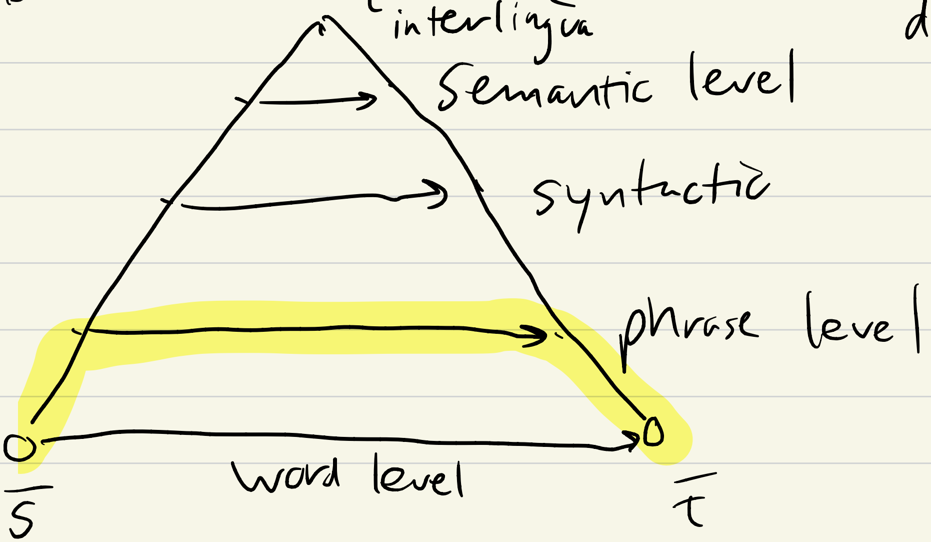
Data: <u>bitext</u>. Set of $(\bar{s}, \bar{t})$ pairs

Big idea:



One view: neural MT systems do this... but not really

Bernard Vauquois (1968) ways of doing MT



Phrase-based MT (PBMT)

+LM

Bitext → word-aligned bitext → phrase table → MT system

alignments

Je fais   I make

Je fais un bureau   I am making a desk

Tu fais   You make

phrase

Je fais un bureau

I am making a desk

alignments tell us word-level correspondences

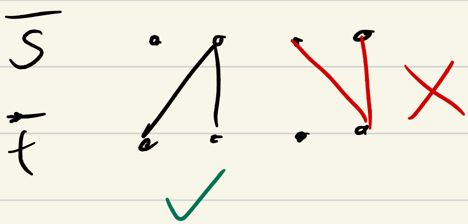
Decoder: search over phrase-by-phrase translations to find the best one

"best": LM score → translation model score

Word alignment

Input: bitext $(\bar{s}, \bar{t})$ pairs

Output: one-to-many alignments from $\bar{s}$ to $\bar{t}$



many to one

placeholder ↓

$\bar{s} =$ Je vais le faire NULL

$\bar{a} = 1 \quad 2 \quad 2 \quad 4 \quad 4 \quad 3$ $\quad a_6 = 3$

$\bar{t} =$ I am going to do it

$a_1 = 1 \quad a_2 = 2 \quad a_3 = 2$

Define $a_i =$ the index in $\bar{s}$ that t_i aligns to

Alignment model: $P(\bar{t}, \bar{a} \mid \bar{s})$

$\bar{t} \approx$ words in HMM

$\bar{a} \approx$ tags

In HMM: we wanted $P(\underset{\text{tags}}{\bar{y}} \mid \underset{\text{words}}{\bar{x}})$

but we modeled $P(\bar{x}, \bar{y})$

Here: we want $P(\bar{a} \mid \bar{s}, \bar{t})$

IBM Model 1 (1993)

$\bar{a} = (a_1 \ldots a_n)$ $\quad \bar{t} = (t_1 \ldots t_n)$ $\quad$ n target words

$\bar{s} = (s_1 \ldots s_m, \text{NULL})$ $\quad$ m source words

Model 1:

"transitions" "emissions"

$$P(\bar{t}, \bar{a} \mid \bar{s}) = \prod_{i=1}^{n} \underbrace{P(a_i)} \, \underbrace{P(t_i \mid s_{a_i})}$$

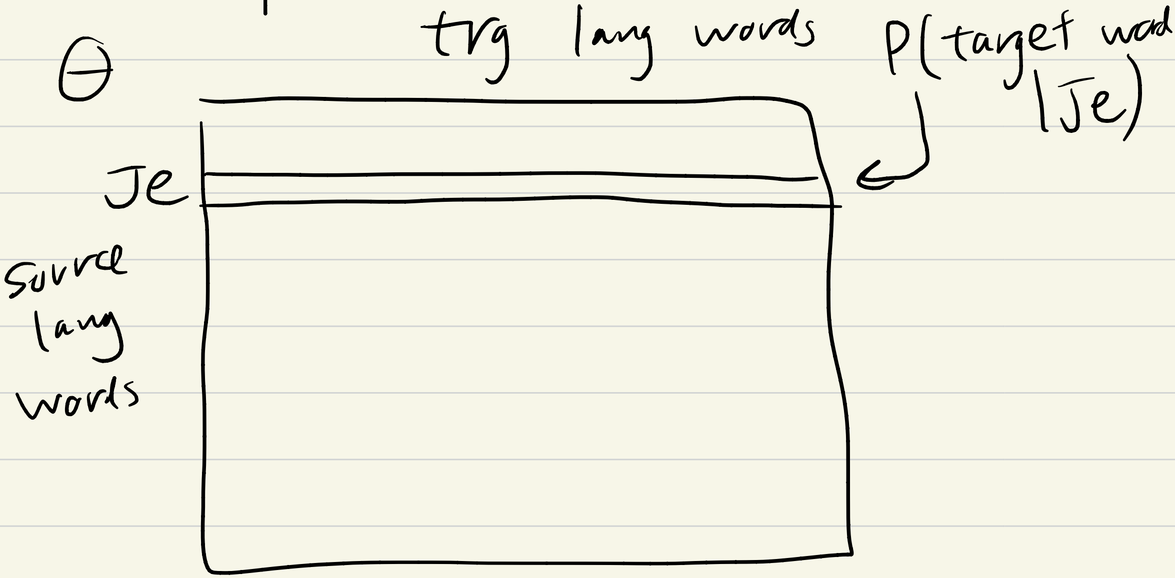
$P(a_i = j) = \frac{1}{m+1}$ uniform over choices

Generate t_i conditioned on s_{a_i}

a_ith source word

Model params: translation dictionary

Θ



$$P(a_i^j \mid \bar{t}, \bar{s}) \propto P(t_i \mid s_j)$$

Θ	I	like	eat
Je	0.8	0.1	0.1
J'	0.8	0.1	0.1
mange	0	0	1.0
aime	0	1.0	0
NULL	0.4	0.3	0.3

Ex 1 $\quad \bar{s} = \text{Je NULL}$

$\bar{t} = \text{I}$

$P(\bar{t}, \bar{a} \mid \bar{s}) \quad \begin{cases} P(\text{I}, a=1 \mid \text{Je}) \\ P(\text{I}, a=2 \mid \text{Je}) \end{cases}$

$t = \text{I}$

$a = 1$ or 2

$\rightarrow P(a=1)\, P(\text{I} \mid \text{Je}) = \frac{1}{2} \cdot 0.8$

$P(a=2)\, P(\text{I} \mid \text{NULL}) = \frac{1}{2} \cdot 0.4$

What we want: $P(\bar{a} \mid \bar{t}, \bar{s})$

$$P(\bar{a} \mid \bar{t}, \bar{s}) = \frac{P(\bar{t}, \bar{a} \mid \bar{s})}{P(\bar{t} \mid \bar{s})}$$

Constant w.r.t. $\bar{a}$

$$P(\bar{a} \mid \bar{t}, \bar{s}) \overset{\text{prop. to}}{\propto} P(\bar{t}, \bar{a} \mid \bar{s})$$

$P(a)$ $P(t \mid s_a)$

$$\text{Ex 1: } P(a \mid I, Je) \propto \begin{cases} \frac{1}{2} \cdot 0.8 \\ \frac{1}{2} \cdot 0.4 \end{cases}$$

normalize these

$$= \begin{array}{ll} 2/3 & a=1 \\ 1/3 & a=2 \end{array}$$

$$P(a_i = j \mid \bar{t}, \bar{s}) \propto P(t_i \mid s_j)$$

Ex 2 J' aime NULL $\bar{s}$

I like $\bar{t}$

You compute $P(a_1 | \bar{s}, \bar{t})$

$a_1 = 1$ J'

$a_1 = 2$ aime

$a_1 = 3$ NULL

$$P(a_1 | \bar{s}, \bar{t}) \propto \begin{cases} 1: \ P(I | J') & 0.8 \\ 2: \ P(I | \text{aime}) & 0 \\ 3: \ P(I | \text{NULL}) & 0.4 \end{cases}$$

$$= \begin{cases} 1 & 2/3 \\ 2 & 0 \\ 3 & 1/3 \end{cases}$$

Learning

Expectation Maximization

Start with $\bar{s}, \bar{t}$, random θ

"E step": compute $P(\bar{a} \mid \bar{s}, \bar{t})$
for each sent. pair

"M step": re-estimate θ by
counting + normalizing

EM maximizes $$\sum_{i=1}^{D} \log \underbrace{\sum_{\bar{a}} P(\bar{t}^{(i)}, \bar{a} \mid \bar{s}^{(i)})}_{P(\bar{t}^{(i)} \mid \bar{s}^{(i)})}$$

Bitext   I   Je

I like   J' aime

even w/ random θ, we get
Count(I | J') is pretty high

IBM Model 2: uses fancier $P(a)$
to capture position in sentence

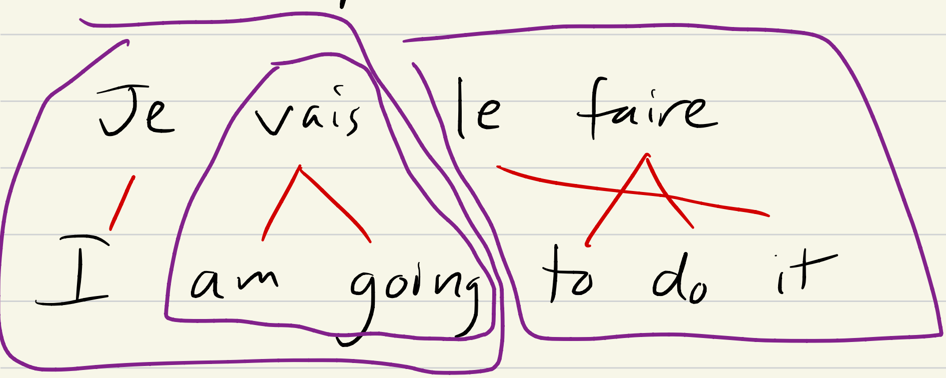
HMM Model: $P(a_{i+1} | a_i)$ models the "gap" in alignments

Models 3+

PBMT

$(\bar{s}, \bar{t}) \to$ aligned

Extract phrases from aligned sents



phrases		scores
Je fais ||| I make		0.9
Je fais ||| I am making		0.6

To translate: choose phrase translations w/ highest score + LM score