Announcements

- A4 due tonight
- A5 out tonight, due next Tuesday

Recap Phrase-based MT

1. Word alignment (src → trg / trg → src, intersect)
2. Phrase extraction
3. Decoding : search over phrase lattice + LM

Today Seq-to-seq models / Neural MT

↳ Connections to RNN LMs

↳ Training

↳ Problems

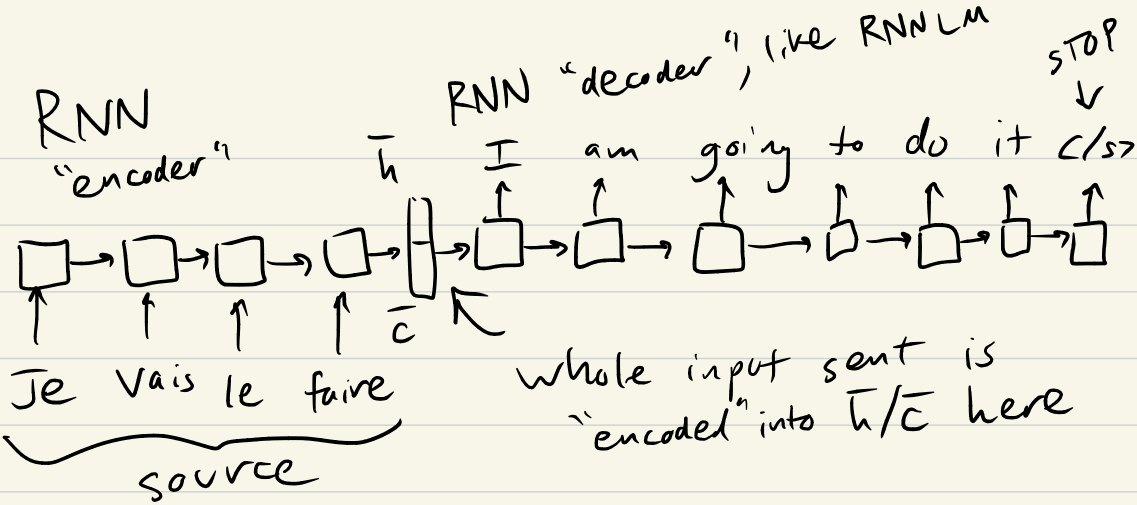
Attention

↳ Mechanism

↳ Connections to alignment

Seq-to-seq models for MT

Sutskever et al. NeurIPS 2014



NMT: encode src w/RNN, "decode" trg with another RNN (diff params)

$$P(\bar{t}\,|\,\bar{s}) = P(t_1\,|\,\bar{s})\,P(t_2\,|\,t_1, \bar{s})\,P(t_3\,|\,t_{1:2}, \bar{s})\ldots$$

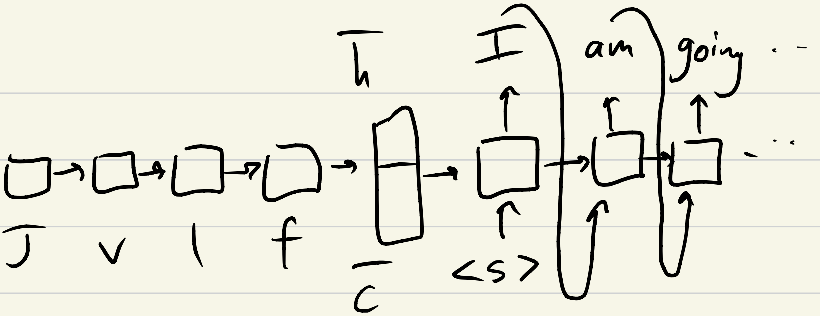
$$P(t_i\,|\,t_{1:i-1}, \bar{s}) = \mathrm{softmax}\left(W^{(d)}\bar{h}_i^{(d)}\right)$$

hidden state at ith step of decoder

$W^{(d)}$: trg vocab $\times$ hidden size mat

Like an RNN LM additionally conditioned on $\bar{s}$

Decoder: feed the previous predicted word into next decoder timestep



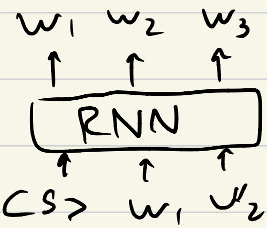
Decoding: $t_0 = \langle s \rangle$

while $t_i \neq \text{STOP}$

$$t_i = \operatorname{argmax}_t P(t \mid t_{1:i-1}, \bar{s})$$

<u>Training</u> Looks like LM training

Recall LM:

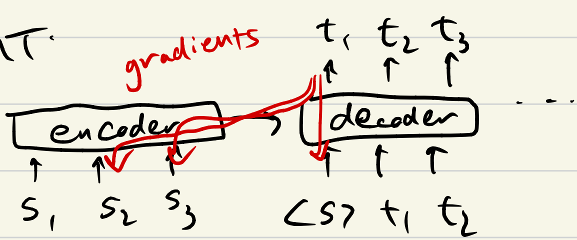


$n \leftarrow$ trg len

t^* gold trg

$$\text{loss} = \sum_{i=1}^{n} \log P(t_i = t_i^* \mid \bar{s}, t^*_{1:i-1})$$

NMT:



encoder trained too!

teacher forcing: feed in true target tokens

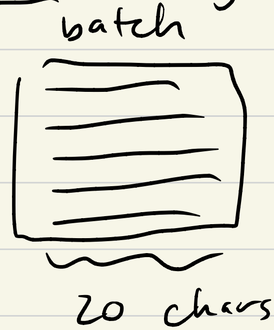
Teaching the model to predict $t_i \mid$ the correct $t_{1:i-1}$ up to this point

What do we really want? Model's predicted sequence $\hat{t}_1 \ldots \hat{t}_n$ has high BLEU score

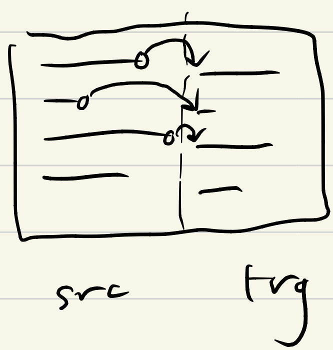
Requires reinforcement learning

<u>Details</u> Lengths can vary, need to pad

LM:



MT:



need padding + smart handling

What goes wrong?

① Repetition

Je vais le faire → I am going to do it going to going to going to ...

Won't happen in PBMT (every word translated as part of one phrase)

Need a notion of coverage in NMT

② Unknown words

Decoder has a fixed vocab $|V_t|$

PBMT: "copy" rules

Pont-de-Buis → copy to output

NMT: produce UNK

1. Repetition (going to going to...)
2. UNKs
3. Poor performance on long sents

Bahdanau et al. (2015)
(attention)

NMT: fixed size $\bar{h}$ / $\bar{c}$ vectors don't scale arbitrarily
LSTMs can't remember for too long

Sutskever: reverse input as a trick to address this

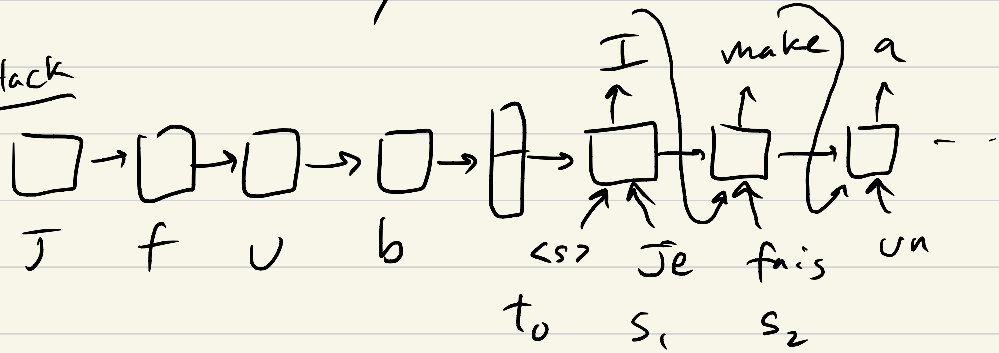
$s_m \; s_{m-1} \ldots s_1 \to t_1 \; t_2 \ldots$

at least remember the first few $\bar{s}$

Attention: Je fais un bureau → I make a desk

What if we know translation should be word-by-word?

Hack



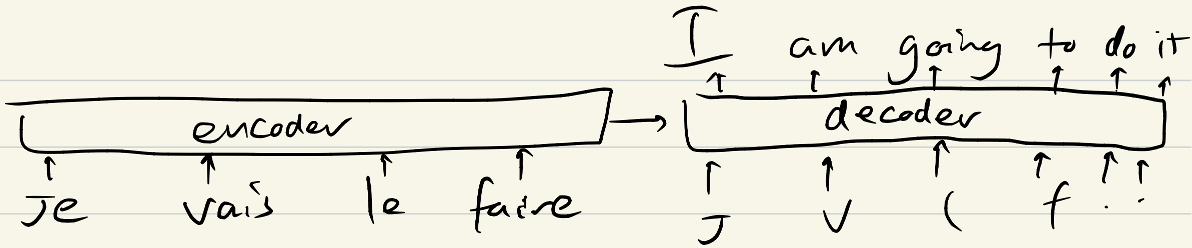
Manually feed s_i into ith decoder step (along with t_{i-1})

Q: What problems can this fix? (1-3)

1+3 ✓ ① Less likely to repeat if we see explicit words

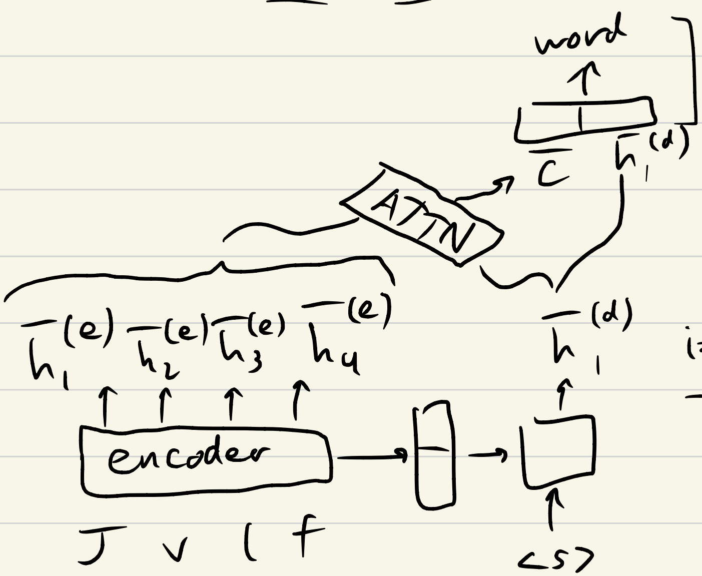
× ② Requires changing output layer

✓ ③ Anchored more to input



- Run out of s before end of t?
- Ordering is inflexible

<u>Attention</u> Target decoder picks <u>where to look</u> in source

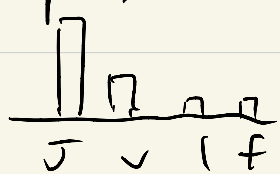


ATTN: inputs $\bar{h}_j^{(e)}$, $\bar{h}_i^{(d)}$

$i=1$

$$\bar{\alpha}^{(t)} = \mathrm{softmax}\begin{bmatrix} \bar{h}_1^{(e)} \cdot \bar{h}_i^{(d)} \\ \bar{h}_2^{(e)} \cdot \bar{h}_i^{(d)} \\ \vdots \\ \bar{h}_m^{(e)} \cdot \bar{h}_i^{(d)} \end{bmatrix}$$

α distribution over source posns



$$\bar{c}_i = \sum_{j=1}^{m} \underset{\text{scalar}}{\alpha_j^{(i)}} \cdot \underset{\text{vector}}{\bar{h}_j^{(e)}}$$

weighted average of $\bar{h}_j^{(e)}$ with $\bar{\alpha}$ as weights

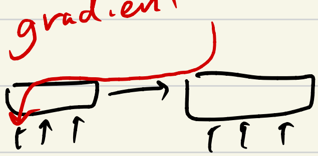
$$P(t_i \mid \bar{s}, t_{1:i-1}) = \text{softmax}\left(W^{(d)} \underbrace{\begin{bmatrix} \bar{h}_i^{(d)} \\ \bar{c}_i \end{bmatrix}}_{\text{concat}}\right)$$

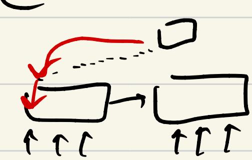
$\bar{\alpha}$ can also use other fcns to compare $\bar{h}^{(d)} / \bar{h}^{(e)}$ besides dot

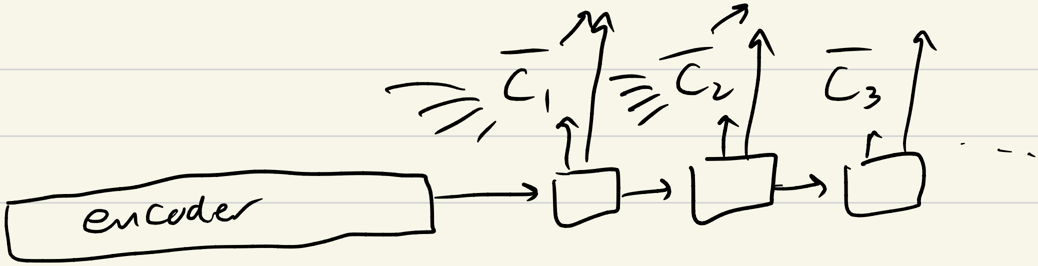
Think about Je fais un bureau

$$\alpha_1 = [0.99, 0.003, 0.003, 0.003]$$

Output: $\text{softmax}\left(W^{(d)} \begin{bmatrix} \bar{h}_1^{(d)} \\ \underbrace{\approx \bar{h}_1^{(e)}}_{\approx \text{Je}} \end{bmatrix}\right)$

No attn: 

w/attn 



Je vais le faire

I am going to do it

<u>Learning mapping</u> 1 2 2 4 4 3

— Not input at train time!

Je vais le faire → 1 2 2 4 4 3

- easier than translation
- regular mapping
- This <u>is</u> learnable!