CS 378 Lecture 18
Neural MT, seq2seq models

Today
- Seq2seq models
- Problems w/seq2seq models
- Attention

Announcements
- A3 grading
- A4 due Tues
- FP

Recap Phrase-based MT

Bitext → word alignment → phrase table → decoder
LM → decoder

Alignment with IBM Model 1

n target words

$\bar{a} = (a_1, \ldots, a_n)$ $\bar{t} = (t_1, \ldots, t_n)$

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

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

Inference: compute $P(\bar{a} \mid \bar{t}, \bar{s})$ (+argmax of that)

$P(a_i \mid \bar{t}, \bar{s})$ proportional to $P(t_i \mid s_{a_i})$

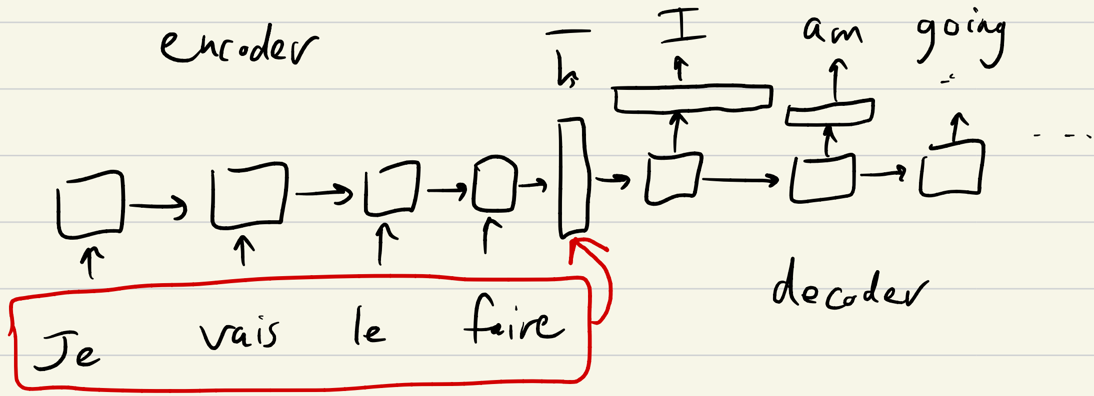
↳ Look up $P(t_i \mid s_{a_i})$ in table for each value of a_i

→ normalize that

Seq2seq models

Model distribution $P(\bar{t} \mid \bar{s})$?

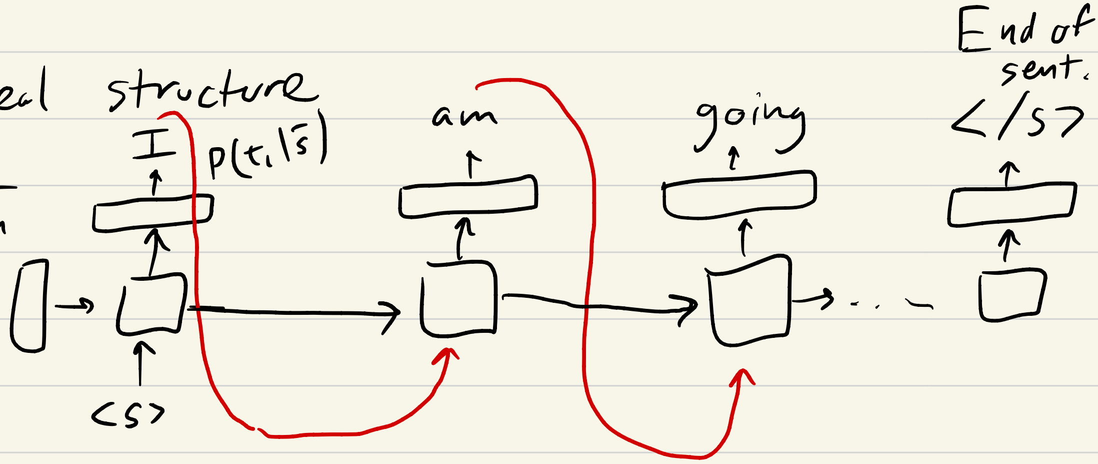
With neural models: encode $\bar{s}$ w/ RNN
"decode" $\bar{t}$ with another RNN
initialized from the state of the first



$$P(\bar{t} \mid \bar{s}) = P(t_1 \mid \bar{s})\, P(t_2 \mid \bar{s}, t_1)\, P(t_3 \mid \bar{s}, t_1, t_2) \cdots$$

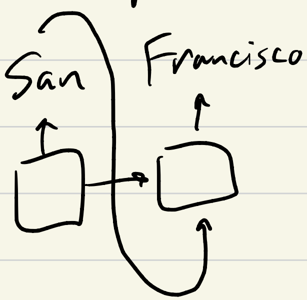
From $\bar{h}$, can reconstruct the source in the target language

Real structure



Feeding output t_i into step $i+1$

$\Rightarrow$ model to capture word – word dependencies



bitext s^*, t^*

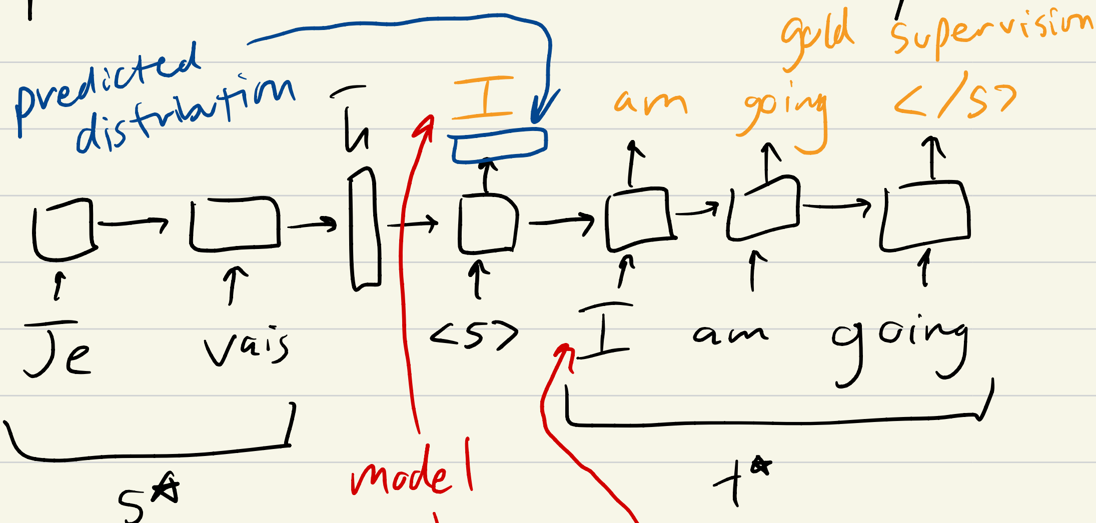
Training Given sent pairs $(\bar{s}, \bar{t})$

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

(sum over $\bar{t}$)

Teacher Forcing

We assume everything is correct prior to the current timestep

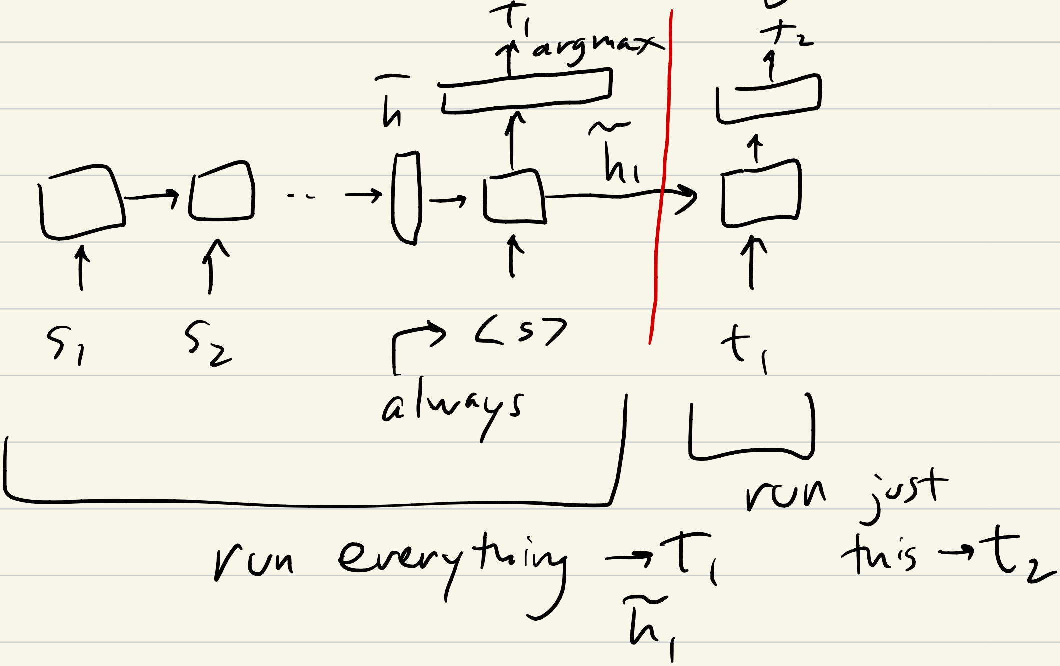


model distribution here could be wrong

still feed in correct first token regardless

Inference

At test time: given $\bar{s}$



Train: single forward() + backward()

Test: n forward passes (until </s>)

save encoder state, don't rerun

Problems with seq2seq

① Model can repeat itself

Je fais un bureau ⇒ I make a desk
a desk a desk ...

RNN doesn't have a notion of "coverage"

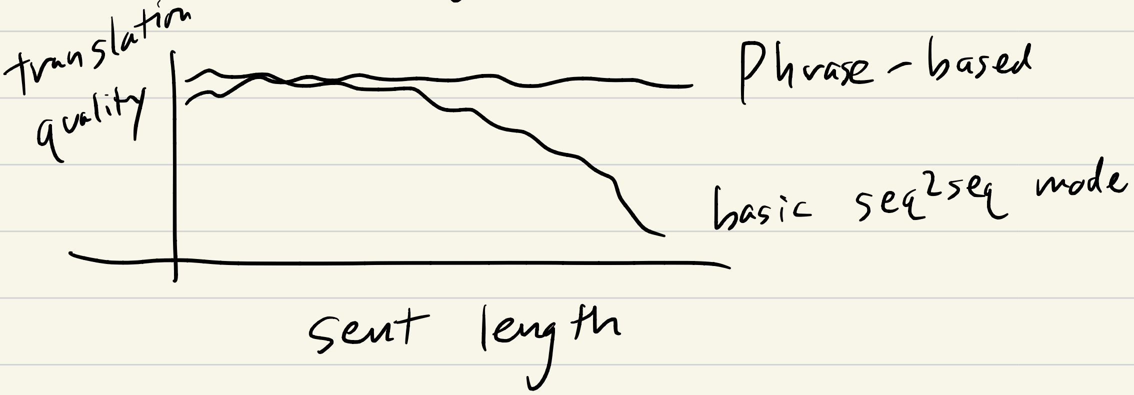
② Fixed vocabulary

Elle est allée à Pont-de-Bois ⇒ She went to UNK

decoder has $|V|$ words in it
($\sim 30{,}000$)

No notion of "copy-paste"

③ Bad at long sentences



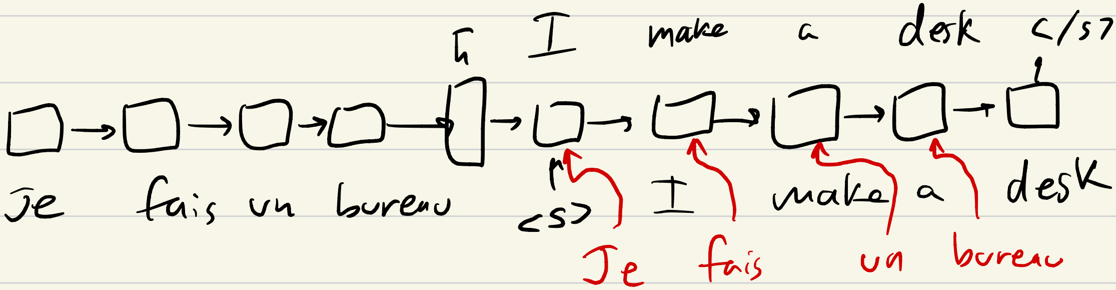
Two reasons: ① h is a fixed size

② Long distances are hard in LSTMs

(trick: reverse the input)

$$\underbrace{s_m \cdots s_1 \to t_1\, t_2 \cdots}_{\text{do well on this}}$$

Attention



This is a word-by-word translation

What if...

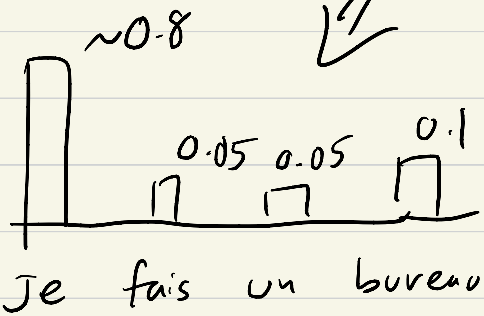
modify LSTM to take 2 inputs

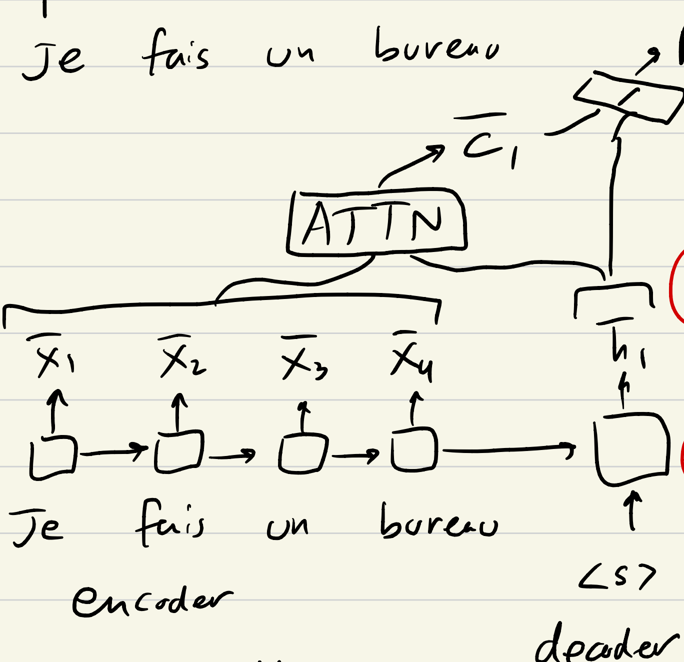
Now it's easy, scales to long seqs

Problem: not always word-by-word
different parts of input may be relevant

Solution: <u>softly</u> pick where we look in the input

Attention: predict weighted average of input, use in the decoder





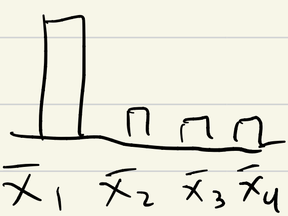
$\bar{X}$ = enc hidden state

Compute

① $e_i = \bar{h}_1^{T} W \bar{x}_i$

attention score

② $\alpha = \text{softmax}(\bar{e})$



③ $\bar{c}_1 = \sum_{i=1}^{m} \alpha_i \bar{x}_i$

weighted sum of $\bar{x}_i$

Then:

$$p(t_1 | \bar{s}) = \text{FFNN}(\bar{h}_1, \bar{c}_1)$$

concatenate

$\bar{c}_1 \approx$ "Je" → makes t_1 = "I" easy to predict

Training: random init + backprop

<u>Details</u> Many ways to do this

pred
↑
attn
↑
decoder
↑
What we saw

Luong

pred
↑
decoder
↑
attn
↑
word

Bahdanau