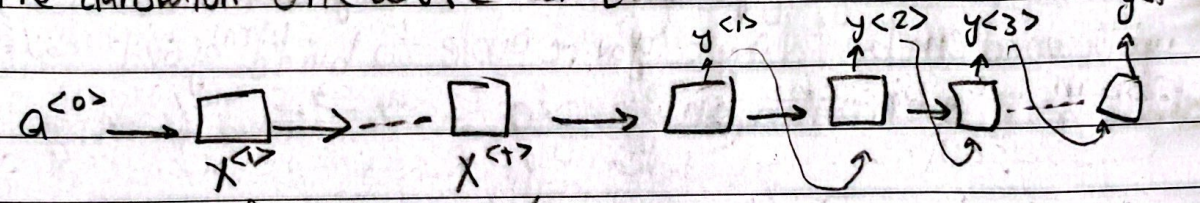
PAGE
DATE

(Week 3)

→ Basic models

- Sequence to sequence models which are useful for everything from machine translation to speech recognition.
- Seq to seq takes Jane visite l'Afrique → Jane is visiting
  $x^{<1>}$ $x^{<2>}$ $x^{<3>}$ y^1 y^2 y^3
- first the encoder network be built as a RNN & This could be GRU or LSTM feeding the input french words one word at time & after Then output a vector that represents the input sentence
- second we can built a decoder network which takes input (encoding output) & then can be trained to output the translation one word at time



→ Picking the most likely sentence.

- seq to seq. models (VS) language models.
- when using Model we try to use Model for Machine translation, we don't try to sample at random from distribution instead we would find the english sentence y, that maximizes that conditional probability $P(y^{<1>}, \ldots y^{<T>} | x)$
  english (y) french (x)

PAGE
DATE

Beam Search algorithm (used during decoding to generate seq. based on trained NN)

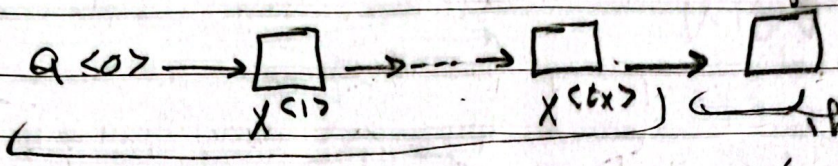
- French → english (or) speech recognition → txt transcript

ex of sentence:

10000 [a, !, in, jane, septemper]

- first thing beam search try to pick the first words of the english translation that's going to output

first we use network fragment encoder & decoder to try



to evalate the prob. of that first word

$P(y^{<1>} | X)$

↳ what prob. of $\hat{y}$ given X

- so in the first step we run the french sentence through the encoder network & then in decoder the softmax output overall 10000 possibilities, then we take those 10000 possible outputs & keep in memory top (3)

(B=3 beam width)

- second, as we picked jane & septemper as most likely choice of first word what beam search will do now is consider what should be in the second word

notes: ① initialization: the model starts from initial input seq ex <start> & computes all prob. for tokens that could appear after this

② selection: retain top K most prob. tokens

③ Expansion: for each retain token compute prob. for the next token

④ pruning: After expanding we have 9 possible sentences so we select top 3 sequences

- So beam search processes multiple sentences & choose one final seq based on cumulative prob. of all possible paths

PAGE
DATE

→ Refinements to beam search

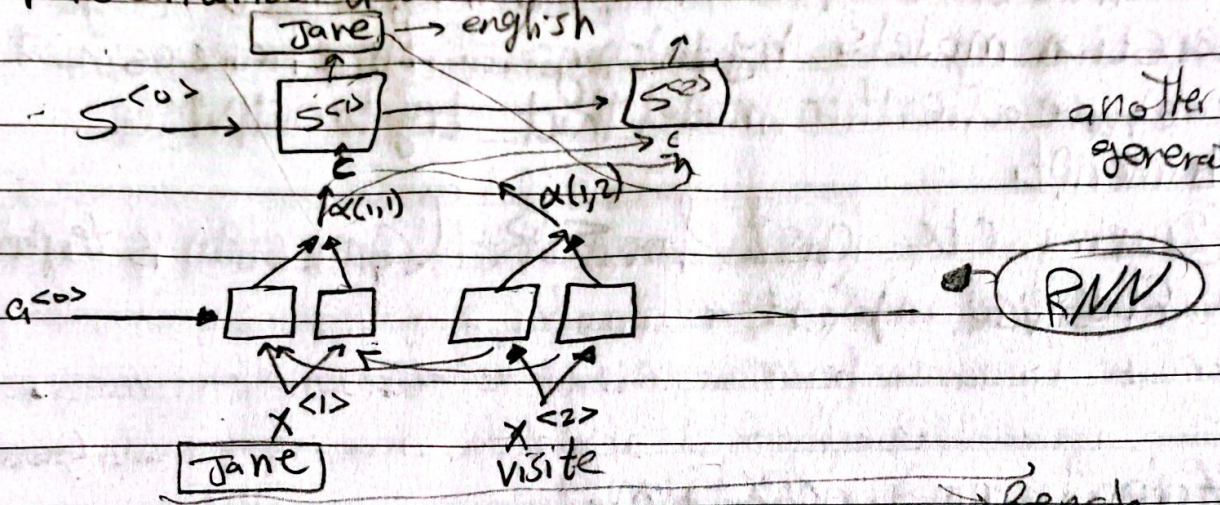
- If the beamwidth is to large Then we consider a lot of possibilities & tend to get better result Because we consider a lot of different options but it will be slower & memory req. will grow also computationally slower & vice versa with low beamwidth
- Beam search is unlike search alg. like BFS or DFS Beam search run faster but is not guaranteed to find exact max $\arg\max P(y|x)$

→ error analysis in Beam search

- How error analysis interact with beam search & How you can figure out whether it is the beam search alg. causing prob. or might be the RNN model
- if we have french sentence human translate y^* & algor. translate $\hat{y}$. So RNN(encoder-decoder) computes $P(y^*|x)$ $P(\hat{y}|x)$
  - if $P(y^*|x) > P(\hat{y}|x)$ Beamsearch ch $\hat{y}$ But y^* higher prob So this means that Beam alg didn't does its job max prob.
  - if $P(y^*|x) \leq P(\hat{y}|x)$ y^* is better than $\hat{y}$ This means that RNN model is at fault

→ Attention model intuition.

- In the encoder Decoder architecture it works well for short sentences so we might achieve high Blev score but for long sentences the performance comes down Bec, it is different for the network to memorize super long sequences. So the Attention model translates maybe like humans looking at part of the sentence at time



- Since we are trying to generate June in english we need to focus on the part that has Jane in french. So what the attension model would be computing is a set of attension weights ($\alpha_{1,}$) to denote when you're generating the first words how much should you pay attention to the first piece of info

→ Attenstion model.

- The alpha (α) tells us how much attention @ the context (C) would depend on the features we're getting @ activations we're getting
- $\alpha^{<t,t_1>}$ is the amount of attention $y^{<t>}$ should pay to $a^{<t>}$

→ Blev score is a metric that evaluate the quality of machine translated text by comparing it to a reference translation

PAGE
DATE

→ Speech recognition

- How seq2seq are applied to audio data
- in speech recg. problem

X (input audio) — y (transcript)

- How can we build speech recognition system ① using attenstion models by taking diff time frame as input & we have attention model that try to output the transcript.

② using ctc cost for SR (Connection is temporal classification) & it

→ trigger word detection.

- we take audio clip, Compute spectrogram features, then generates x_1, x_2 x_3 audio features, then we pass through RNN and so all that remains to be done is to define the target labels.
- the trainy set you can set all the target labels to 0 for every thing before the tergetpoint(siri) then all after to be 1