COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Neural sequence-to-sequence models for machine translation

CMSC 470

Marine Carpuat

Machine Translation

- Translation system
  - Input: source sentence F
  - Output: target sentence E
  - Can be viewed as a function

$$\hat{E} = \mathrm{mt}(F)$$

- Modern machine translation systems

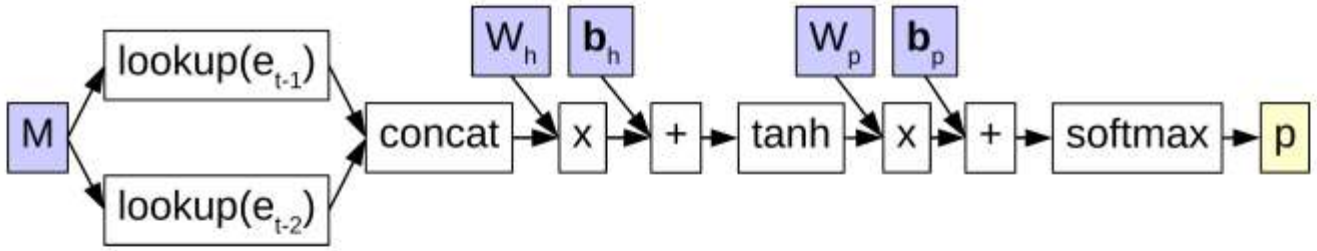
$$\hat{E} = \underset{E}{\operatorname{argmax}}\ P(E \mid F; \theta)$$

- 3 problems
  - Modeling
    - how to define P(.)?
  - Training/Learning
    - how to estimate parameters from parallel corpora?
  - Search
    - How to solve argmax efficiently?

Neural Machine Translation

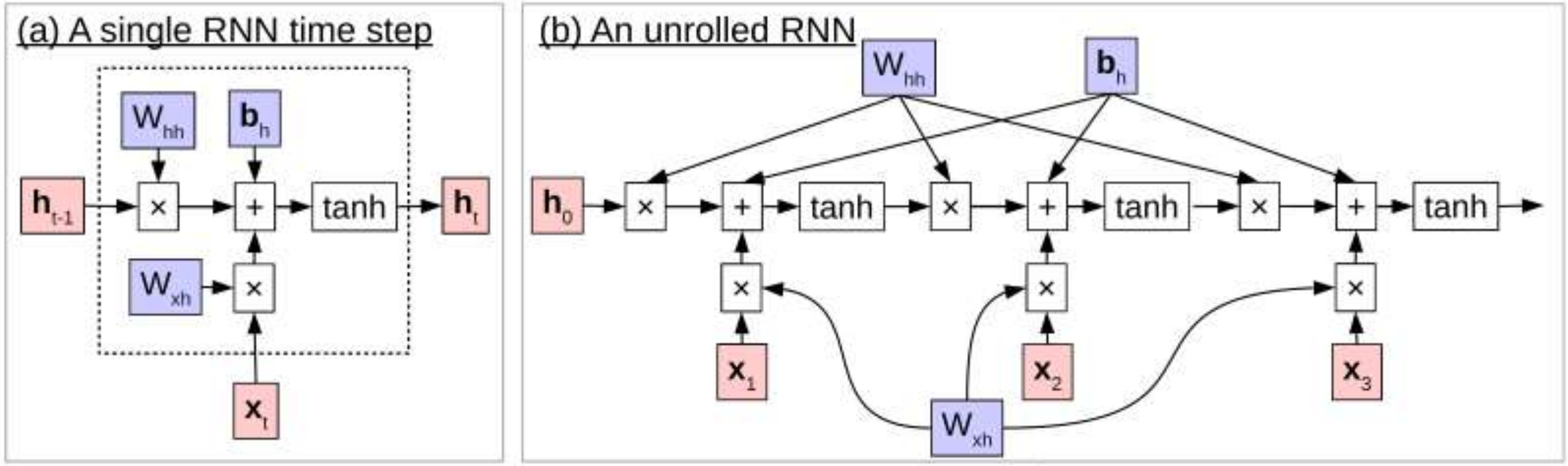
- Neural language models review
- Sequence to sequence models for MT
  - Encoder-Decoder
  - Sampling and search (greedy vs beam search)
  - Training
  - Practical tricks
- Sequence to sequence models for other NLP tasks

A feedforward neural 3-gram model



$$
\begin{aligned}
\boldsymbol{m} &= \mathrm{concat}(M_{\cdot,e_{t-2}}, M_{\cdot,e_{t-1}}) \\
\boldsymbol{h} &= \tanh(W_{mh}\boldsymbol{m} + \boldsymbol{b}_h) \\
\boldsymbol{s} &= W_{hs}\boldsymbol{h} + \boldsymbol{b}_s \\
\boldsymbol{p} &= \mathrm{softmax}(\boldsymbol{s})
\end{aligned}
$$

A recurrent language model

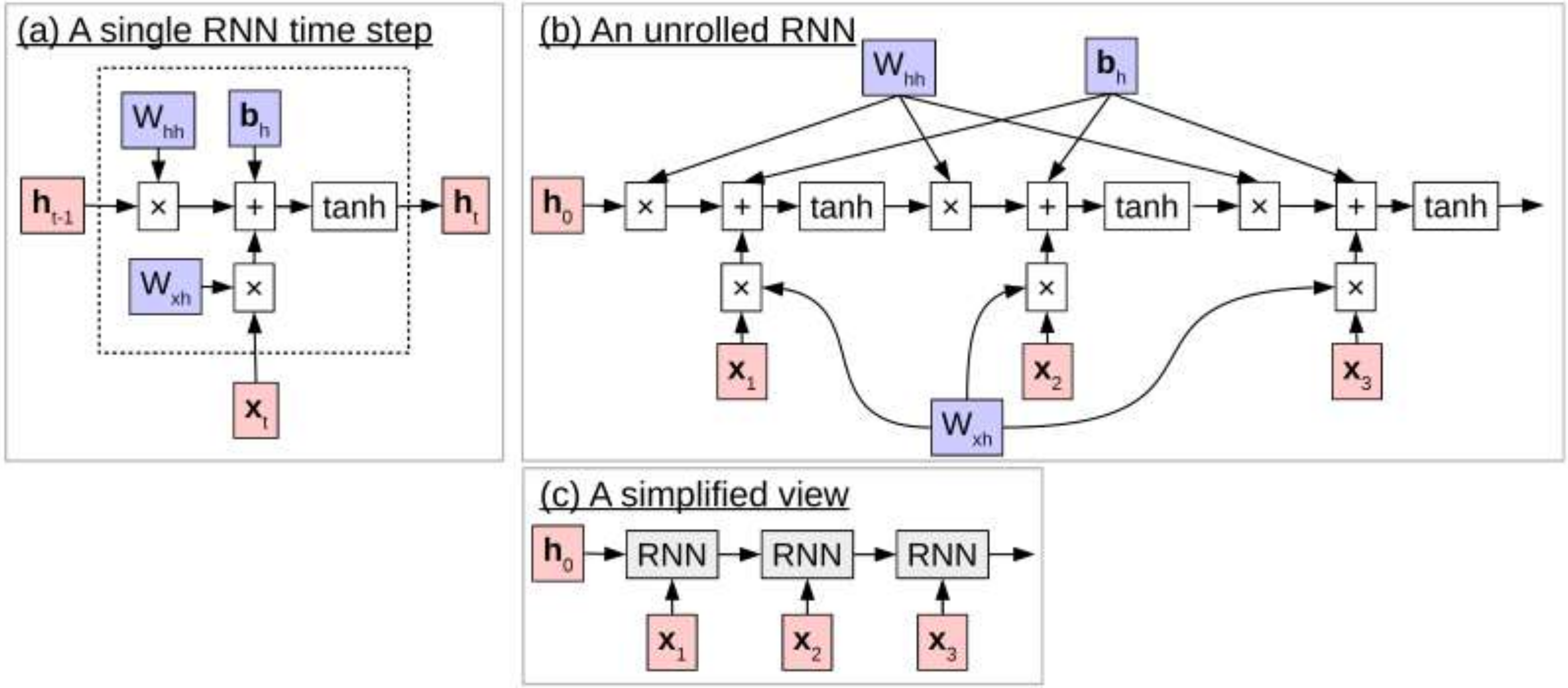


$$
\boldsymbol{m}_t = M_{\cdot, e_{t-1}}
$$

$$
\boldsymbol{h}_t = \begin{cases} \tanh(W_{mh}\boldsymbol{m}_t + W_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h) & t \geq \\ \mathbf{0} & \text{otherwise.} \end{cases}
$$

$$
\boldsymbol{p}_t = \operatorname{softmax}(W_{hs}\boldsymbol{h}_t + b_s).
$$

A recurrent language model



$$\boldsymbol{m}_t = M_{\cdot,e_{t-1}}$$

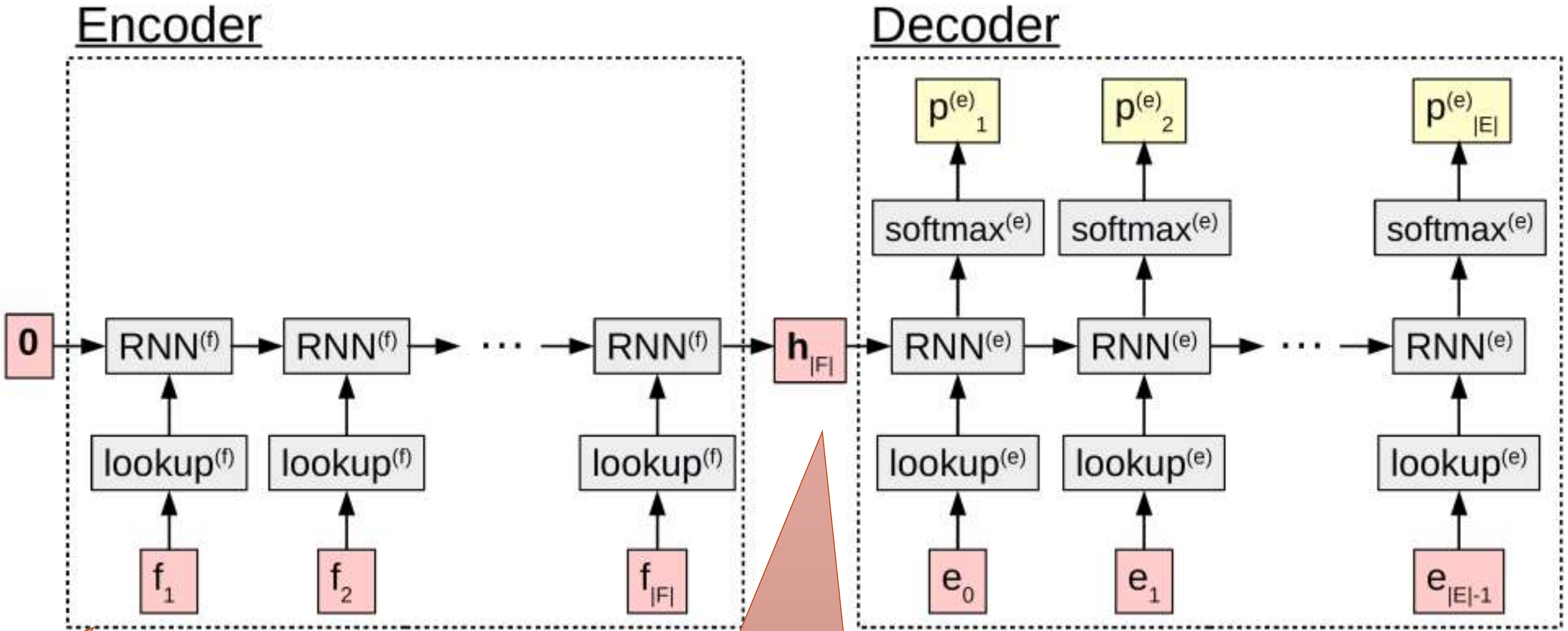
$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{m}_t, \boldsymbol{h}_{t-1})$$

$$\boldsymbol{p}_t = \mathrm{softmax}(W_{hs}\boldsymbol{h}_t + b_s).$$

Neural Machine Translation

- Neural language models review
- Sequence to sequence models for MT
  - Encoder-Decoder
  - Sampling and search (greedy vs beam search)
  - Practical tricks
- Sequence to sequence models for other NLP tasks

P(E|F) as an encoder-decoder model



The Encoder models the input/source entence F=(f1,….f|F|)

The decoder hidden state is initialized with the last hidden state of the encoder

The Decoder models the output/target sentence E=(e1,….e|E|).

P(E|F) as an encoder-decoder model

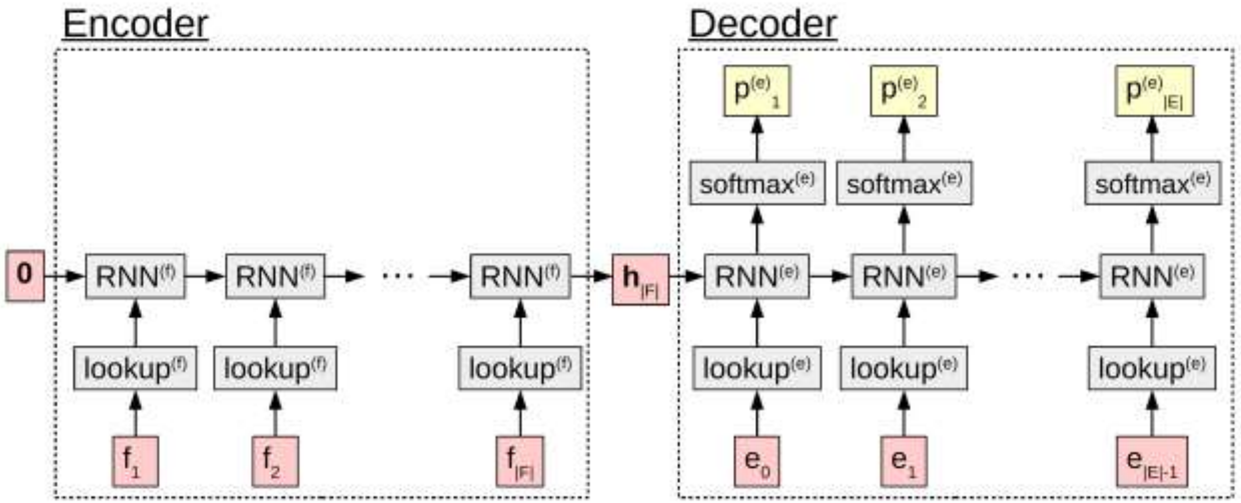
$$\boldsymbol{m}_t^{(f)} = M_{\cdot,f_t}^{(f)}$$

$$\boldsymbol{h}_t^{(f)} = \begin{cases} \mathrm{RNN}^{(f)}(\boldsymbol{m}_t^{(f)}, \boldsymbol{h}_{t-1}^{(f)}) & t \geq 1, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

$$\boldsymbol{m}_t^{(e)} = M_{\cdot,e_{t-1}}^{(e)}$$

$$\boldsymbol{h}_t^{(e)} = \begin{cases} \mathrm{RNN}^{(e)}(\boldsymbol{m}_t^{(e)}, \boldsymbol{h}_{t-1}^{(e)}) & t \geq 1, \\ \boldsymbol{h}_{|F|}^{(f)} & \text{otherwise.} \end{cases}$$

$$\boldsymbol{p}_t^{(e)} = \mathrm{softmax}(W_{hs}\boldsymbol{h}_t^{(e)} + b_s)$$



Generating Output

- We have a model $P(E|F)$, how can we generate translations?

- 2 methods

  - **Sampling**: generate a random sentence according to probability distribution

  - **Argmax**: generate sentence with highest probability

Ancestral Sampling

While $e_{j-1} != \text{</s>}$
$$e_j \sim \mathrm{P}(e_j|\mathrm{F}, e_1, \ldots, e_{j-1})$$

- Randomly generate words one by one
- Until end of sentence symbol
- Done!

Greedy search

While $e_{j-1} != </s>$
$e_j = \text{argmax}\, \mathrm{P}(e_j | \mathrm{F}, e_1, \ldots, e_{j-1})$

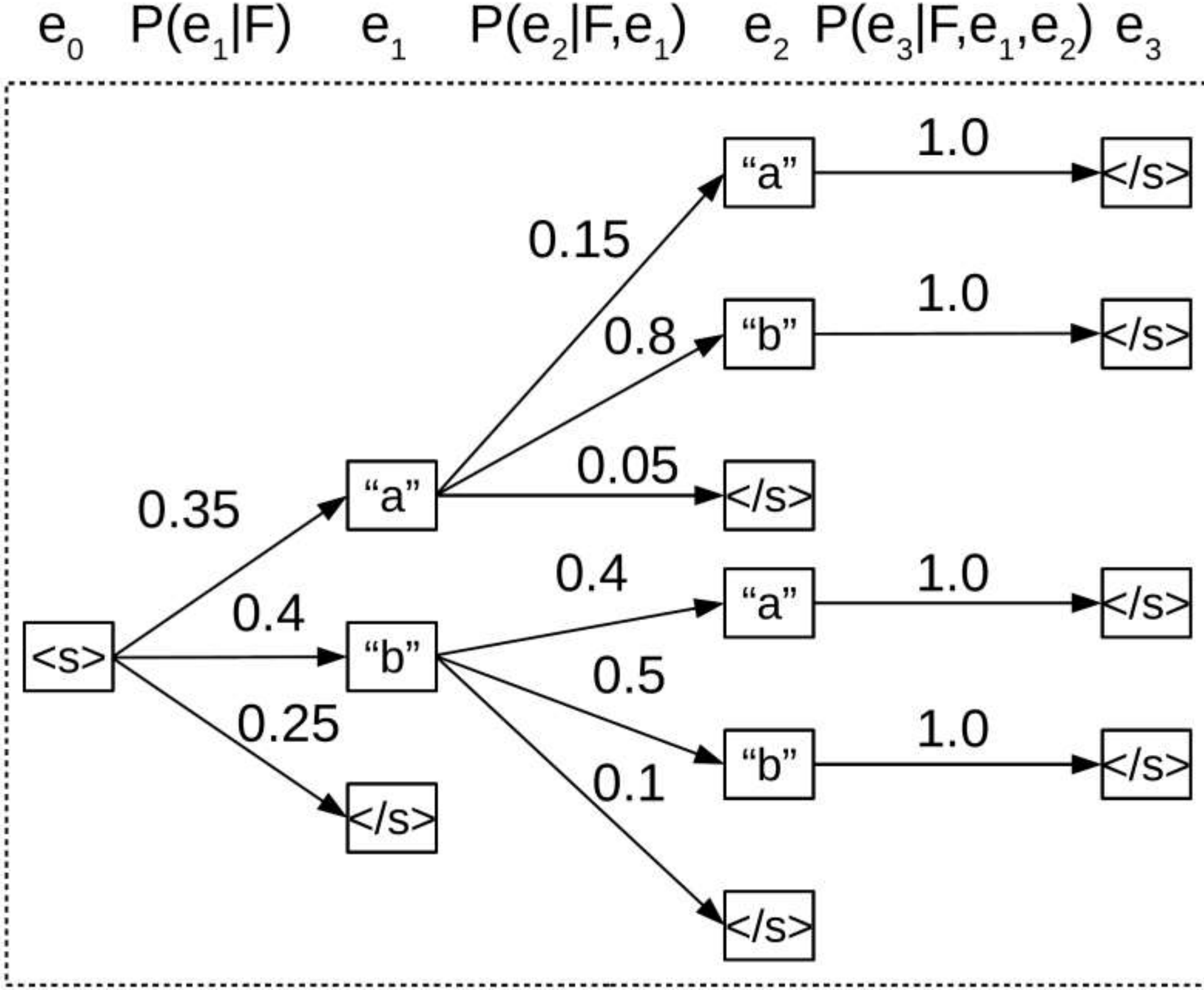
- One by one, pick single highest probability word
- Problems
  - Often generates easy words first
  - Often prefers multiple common words to rare words

Greedy Search Example

Consider this complete search graph for a model with vocabulary {a,b,</s>}

What sequence does greedy search produces?

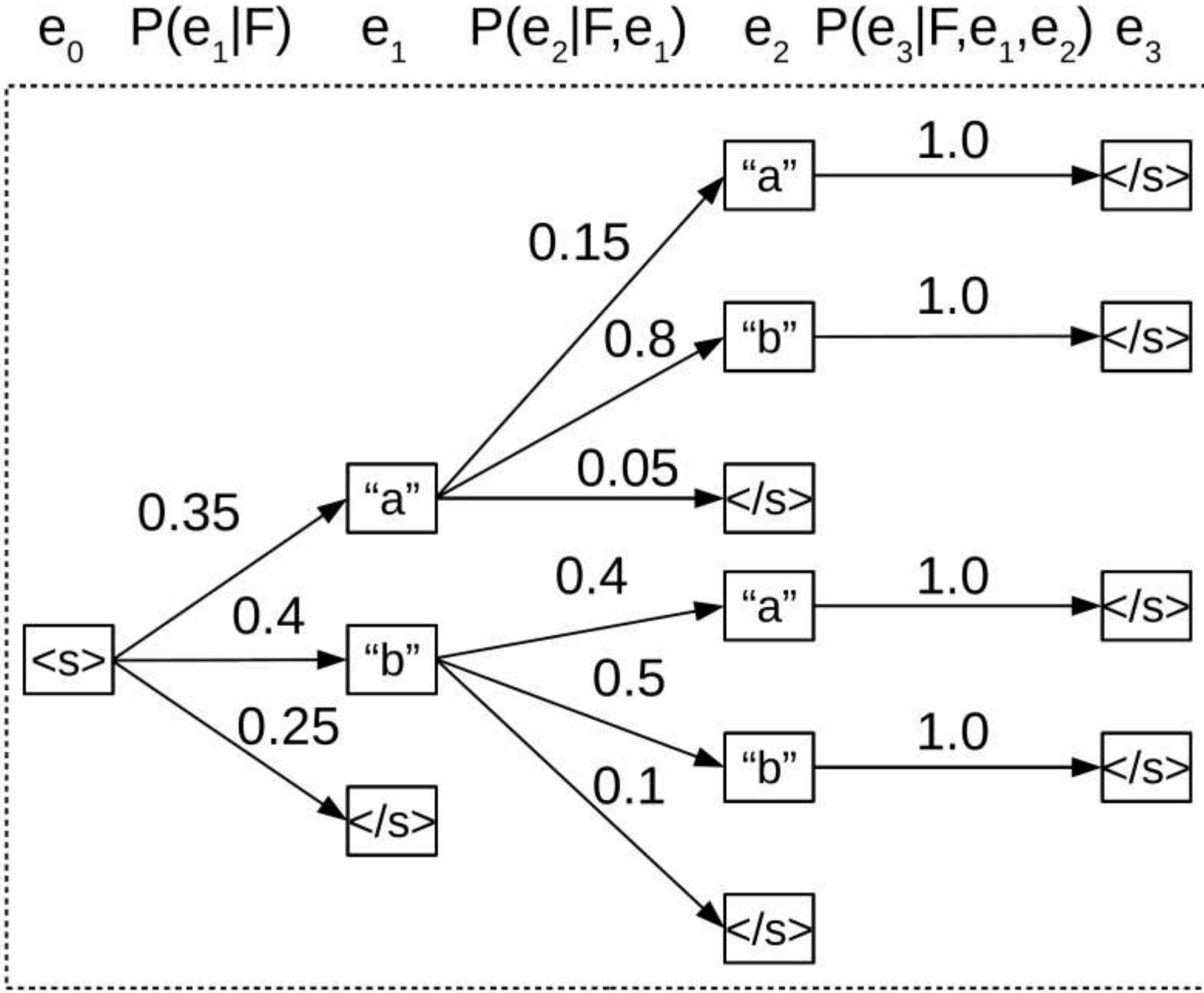
What is the best scoring sequence in the search space?



Greedy Search Example

Consider this complete search graph for a model with vocabulary {a,b,</s>}

Here greedy search fails to discover the best scoring output!

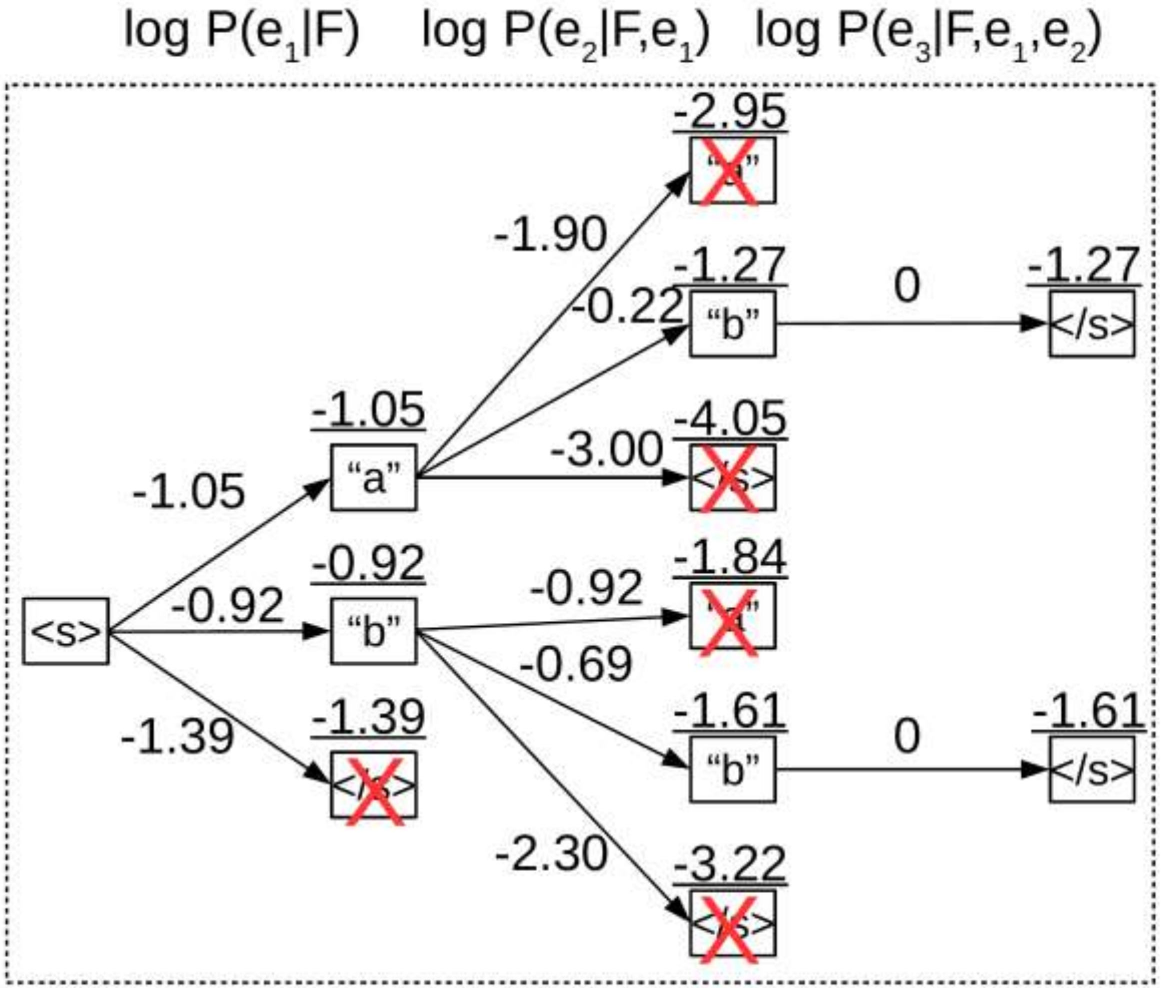


Beam Search

Idea: consider b top hypotheses at each time step

At each time step:

- Expand the b hypotheses for all words in the vocabulary
- Prune down to the top b hypotheses
- Move to next step



Example beam search with b = 2

Neural Machine Translation

- Neural language models review
- Sequence to sequence models for MT
  - Encoder-Decoder
  - Sampling and search (greedy vs beam search)
  - Training
  - Practical tricks
- Sequence to sequence models for other NLP tasks