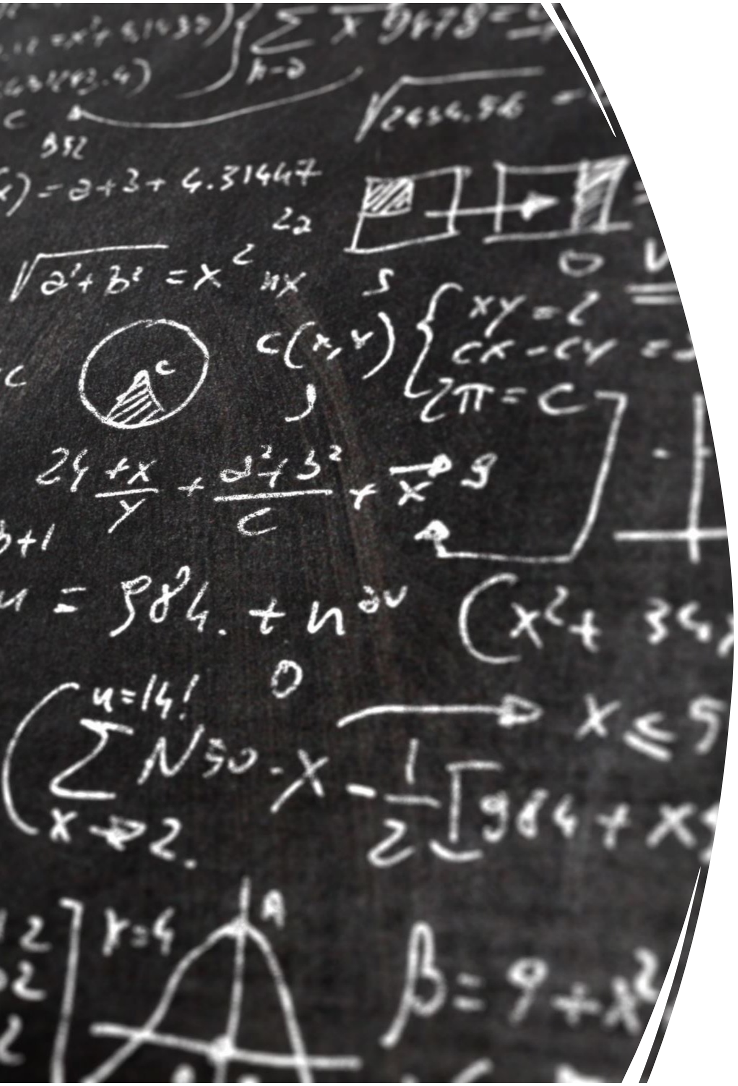


CS 383 – Computational Text Analysis

Lecture 19
Generating text

Adam Poliak

03/29/2023

Slides adapted from Philipp Koehn

Machine Learning in a nutshell

In a ML model, what are we training?

- **Parameters!**

How do we train parameters in supervised learning?

train parameters == figure out values for the parameters

- Update weights by using them to make predictions and seeing **how far off our predictions** are
    - **Loss function!**

Algorithm to learn weights?

- **SGD**
- Others exist but not covering them

Outline

Generating text

Evaluating Generated Text

When do we even want to generate text?

Machine Translation

Summarization

ChatBots

…

Generating text so far

Greedy approach:

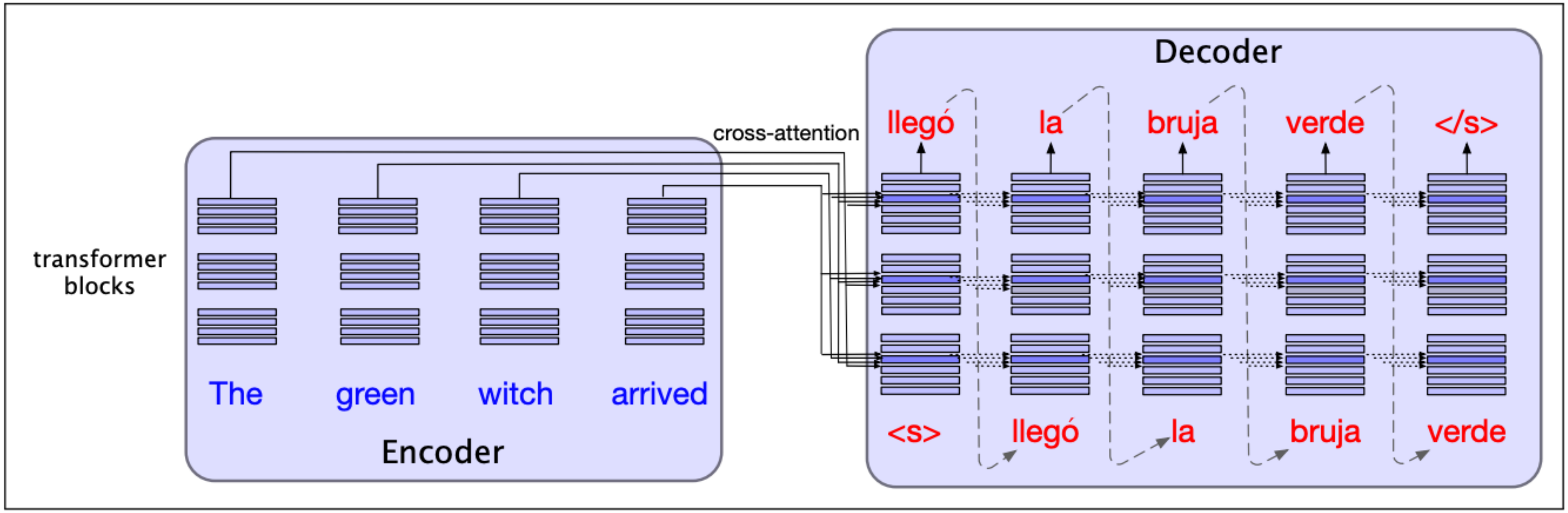
- At each time step, choose the word with the highest probability
- Optionally can condition on the previous generated words
  - Like in Language modeling!

Side: how do we prove correctness of greedy approaches?

Hint: use induction

https://jeffe.cs.illinois.edu/teaching/algorithms/book/04-greedy.pdf

Encoder-decoder for MT



Why might Greedy not be optimal?

$$\hat{y}_t = \text{argmax}_{w \in V} P(w|y_1 \ldots y_{t-1})$$

Choosing the best local word to generate, not necessarily the best sentence/paragraph globally to generate

Possible solutions?

Beam Search Decoding!

Search Tree

Nodes: states

The potential word to generate

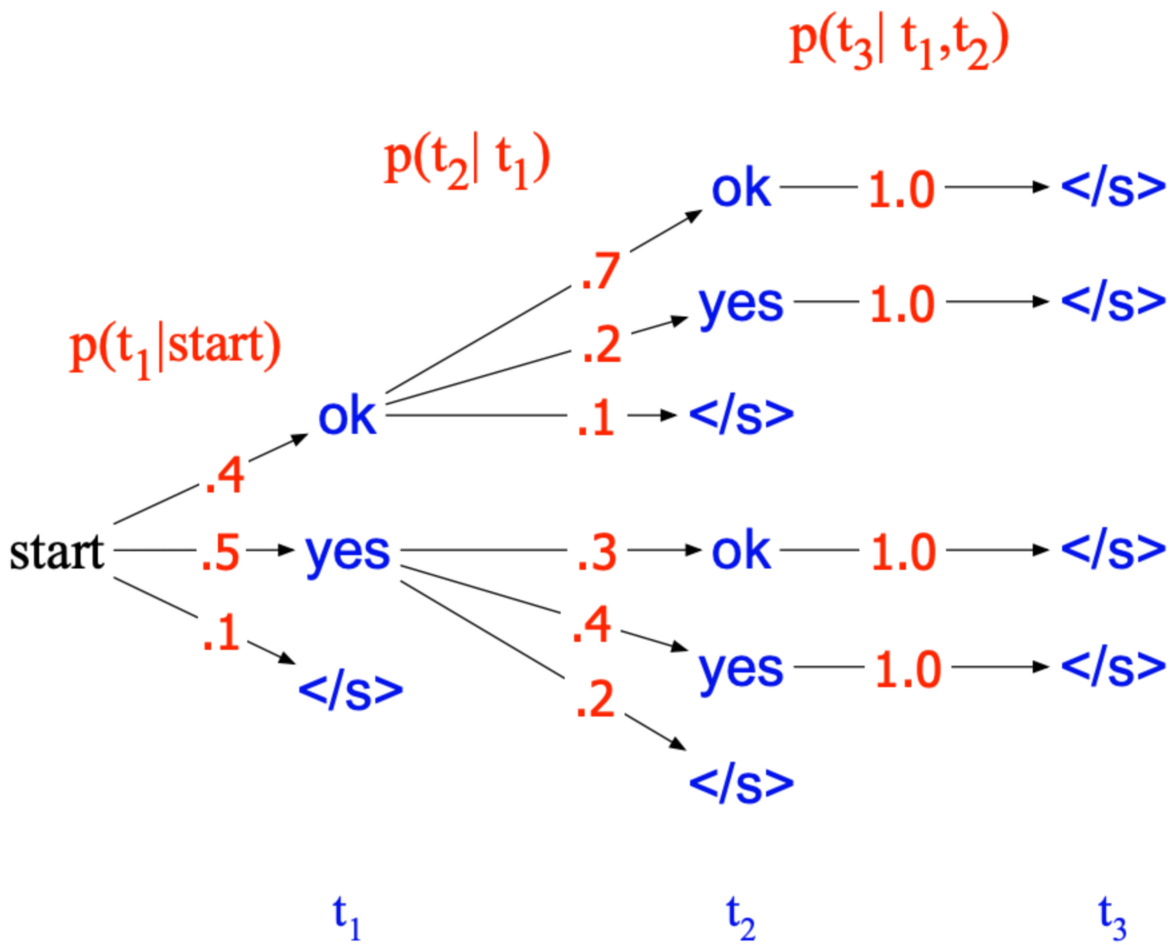
Branches/edges: actions

Edges have weights

Score for each word

Probability of generating each word

Search Tree Example



How long to search through the tree?

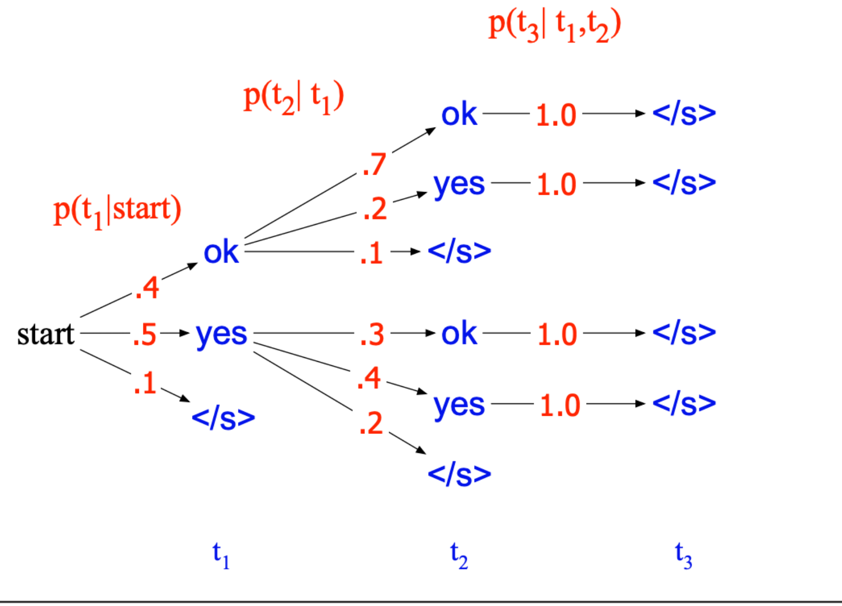
We need to consider all paths

How many paths are there?

$$V^T$$

Solution:

Beam search!



Beam search

Only keep *k* vocabulary terms are each time step

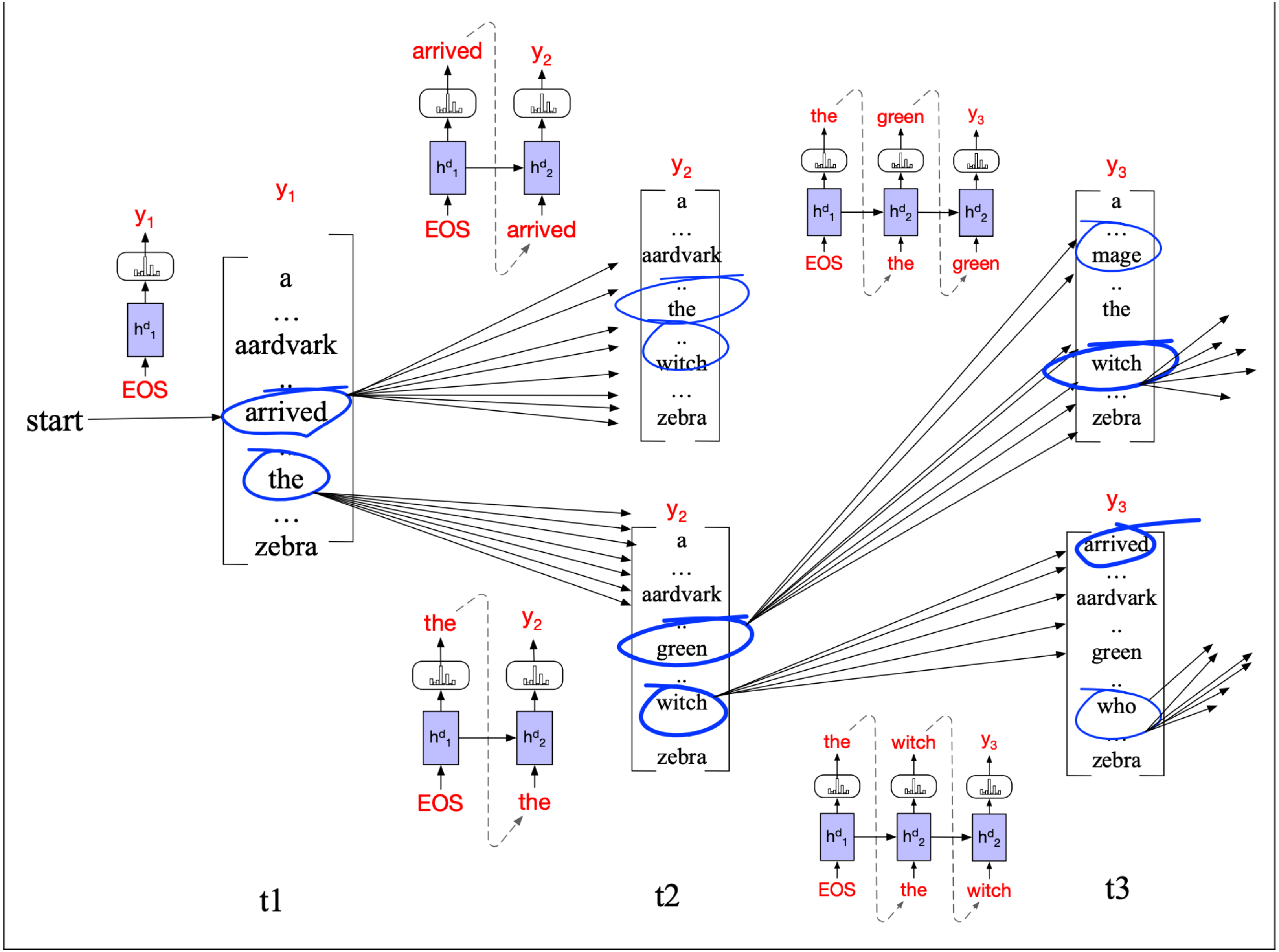
beam width

Apply softmax over vocabulary, keep just top *k* terms

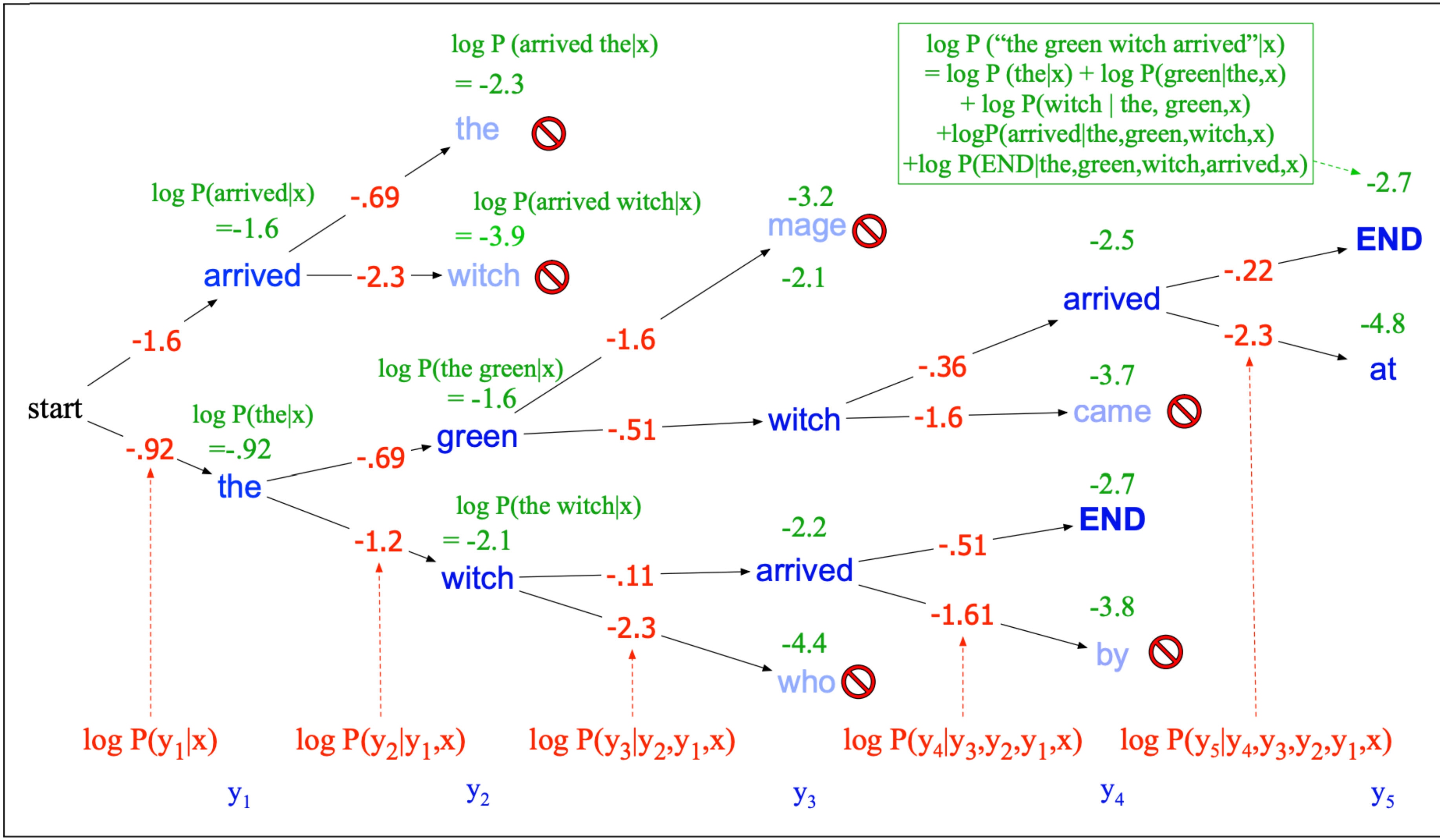
search frontier

k-hypotheses

Incrementally repeat this until we generate </s>



start
y1
EOS
a
…
aardvark
arrived
the
zebra
arrived
y2
EOS
arrived
y2
a
aardvark
the
witch
zebra
the
y2
EOS
the
y2
a
aardvark
green
witch
zebra
the
green
y3
EOS
the
green
y3
a
mage
the
witch
zebra
y3
arrived
aardvark
green
who
zebra
the
witch
y3
EOS
the
witch
t1
t2
t3



log P (arrived the|x)
= -2.3
the
log P (“the green witch arrived”|x)
= log P (the|x) + log P(green|the,x)
+ log P(witch | the, green,x)
+logP(arrived|the,green,witch,x)
+log P(END|the,green,witch,arrived,x)
-2.7
log P(arrived|x)
=-1.6
-.69
log P(arrived witch|x)
= -3.9
-3.2
mage
END
-2.5
arrived
-2.3
witch
-2.1
-.22
arrived
-1.6
-1.6
-2.3
-4.8
start
log P(the green|x)
= -1.6
-.36
at
-3.7
log P(the|x)
=-.92
-.92
-.69
green
-.51
witch
-1.6
came
the
-2.7
log P(the witch|x)
= -2.1
END
-2.2
-1.2
-.51
arrived
witch
-.11
-1.61
-3.8
-2.3
-4.4
by
who
log P(y1|x)
log P(y2|y1,x)
log P(y3|y2,y1,x)
log P(y4|y3,y2,y1,x)
log P(y5|y4,y3,y2,y1,x)
y1
y2
y3
y4
y5

Scoring hypotheses

$$
\begin{aligned}
score(y) &= \log P(y|x) \\
&= \log \left(P(y_1|x)P(y_2|y_1,x)P(y_3|y_1,y_2,x)\ldots P(y_t|y_1,\ldots,y_{t-1},x)\right) \\
&= \sum_{i=1}^{t} \log P(y_i|y_1,\ldots,y_{i-1},x)
\end{aligned}
$$

What issues might there be?

Longer sentences will have higher scores

Solution:

Normalize by length of generated sentence

Algorithm

```
function BeamDecode(c, beam_width) returns best paths

  y0, h0 ← 0
  path ← ()
  complete_paths ← ()
  state ← (c, y0, h0, path)          ;initial state
  frontier ← ⟨state⟩        ;initial frontier

  while frontier contains incomplete paths and beamwidth > 0
     extended_frontier ← ⟨⟩
     for each state ∈ frontier do
          y ← Decode(state)
          for each word i ∈ Vocabulary do
               successor ← NewState(state, i, y_i)
               extended_frontier ← AddToBeam(successor, extended_frontier,
                                                               beam_width)

     for each state in extended_frontier do
          if state is complete do
               complete_paths ← Append(complete_paths, state)
               extended_frontier ← Remove(extended_frontier, state)
               beam_width ← beam_width - 1
     frontier ← extended_frontier

   return completed_paths

function NewState(state, word, word_prob) returns new state

function AddToBeam(state, frontier, width) returns updated frontier

  if Length(frontier) < width then
     frontier ← Insert(state, frontier)
  else if Score(state) > Score(WorstOf(frontier))
     frontier ← Remove(WorstOf(frontier))
     frontier ← Insert(state, frontier)
  return frontier
```

Beam search decoding

What data structure might you use to store hypotheses?

- Priority Queue
  - Maintain the order of the best hypotheses

Beam search in action:

http://mt-class.org/jhu/stack-decoder/

Other decoding methods

A*

Hill-Climbing

Greedy approach

Finite State Transducers

Outline

Generating text

Evaluating Generated Text

How would you determine if a generated text is *good*

Why cannot we not use accuracy (or other stats from the confusion matrix)?

Compare the generated text to some human generated text

Comparing to human generated text

Why is this a hard problem?

many different generated text as acceptable → semantic equivalence / similarity

How might you do this?

Metrics:

BLEU

Embedding approaches (e.g. BertScore)

BLEU

N-gram overlap between machine translation output and reference translation

Compute precision for n-grams of size 1 to 4

Add brevity penalty (for too short translations)

$$\text{BLEU} = \min\left(1, \frac{\textit{output-length}}{\textit{reference-length}}\right) \left(\prod_{i=1}^{4} \textit{precision}_i\right)^{\frac{1}{4}}$$

Typically computed over the entire corpus, not single sentence

BLEU Example

SYSTEM A: Israeli officials responsibility of airport safety

2-GRAM MATCH (Israeli officials) 1-GRAM MATCH (airport)

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security Israeli officials are responsible

2-GRAM MATCH (airport security) 4-GRAM MATCH (Israeli officials are responsible)

Metric	System A	System B
precision (1gram)	3/6	6/6
precision (2gram)	1/5	4/5
precision (3gram)	0/4	2/4
precision (4gram)	0/3	1/3
brevity penalty	6/7	6/7
BLEU	0%	52%

Plan for next week

Statistical Inference

- Quantifying uncertainty
  - Hypothesis testing
  - Null hypothesis
  - p-value
  - Confidence interval
  - Bootstraping

- Forecast (time-series prediction)