Decoding

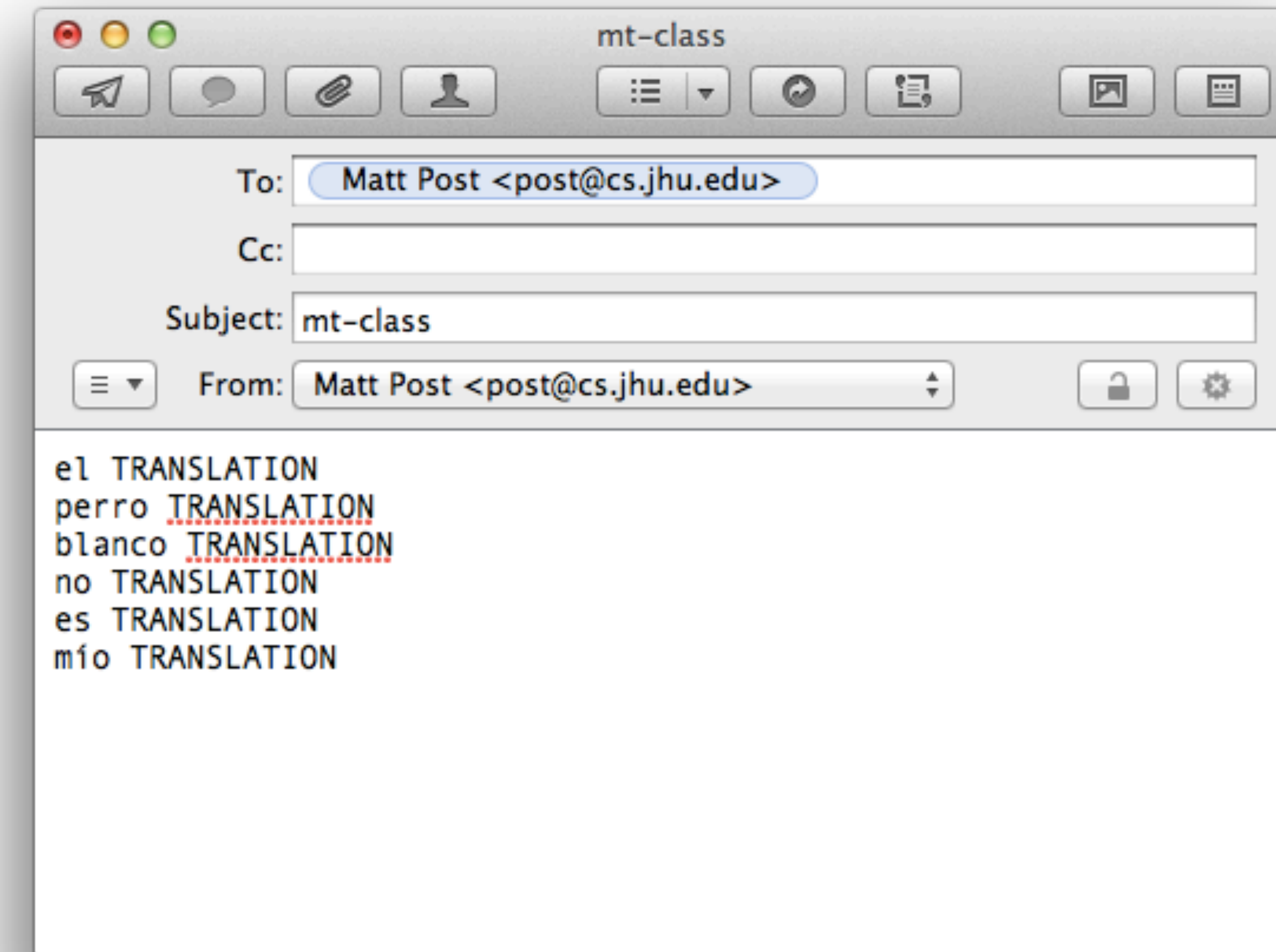
continued

Activity

Build a translation model that we'll use later today.

Instructions

- Subject is "mt-class"
- The body has six lines
- There is one, one-word translation per line

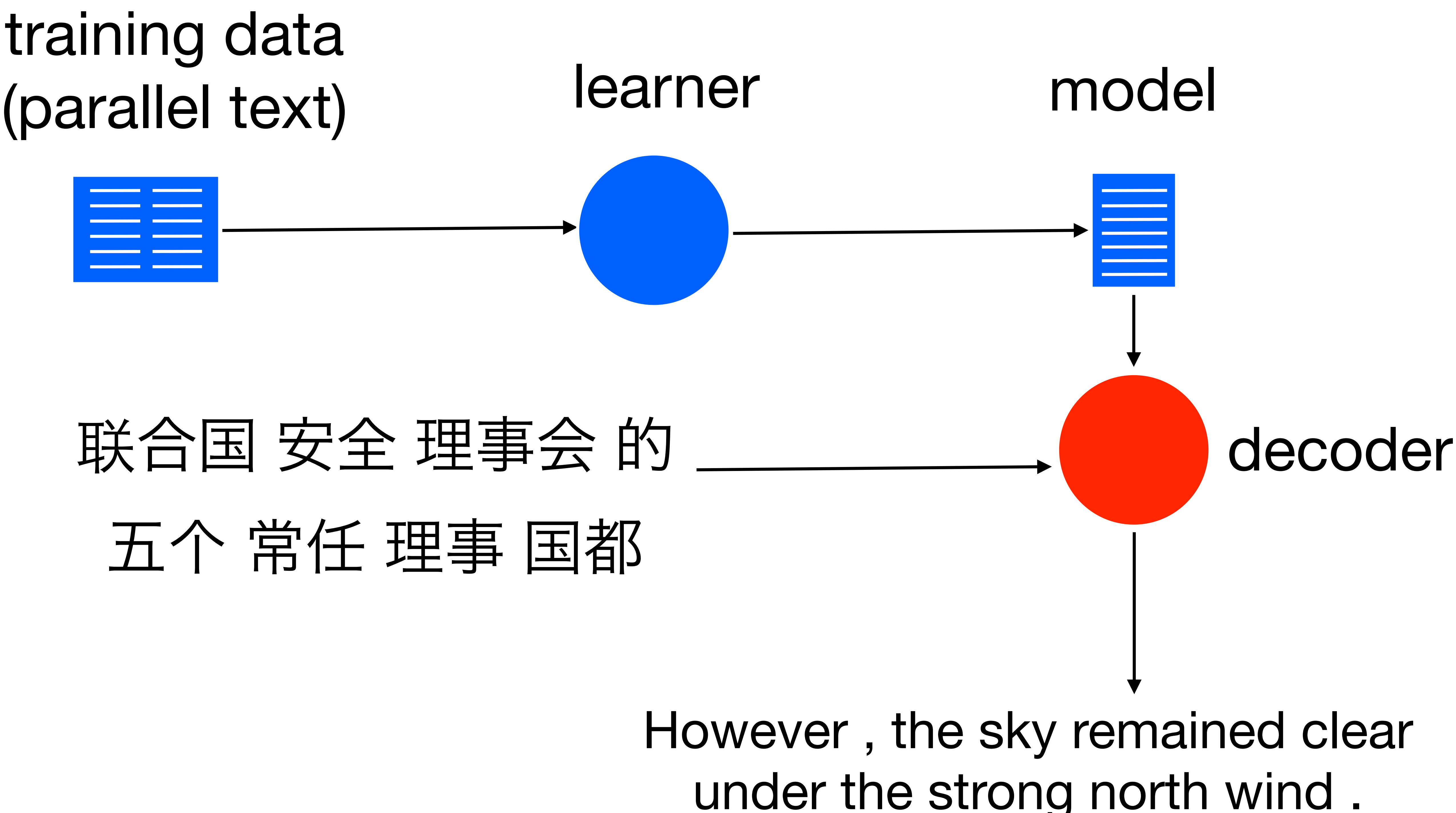


ADMINISTRATIVE

- Schedule for language in 10 minutes

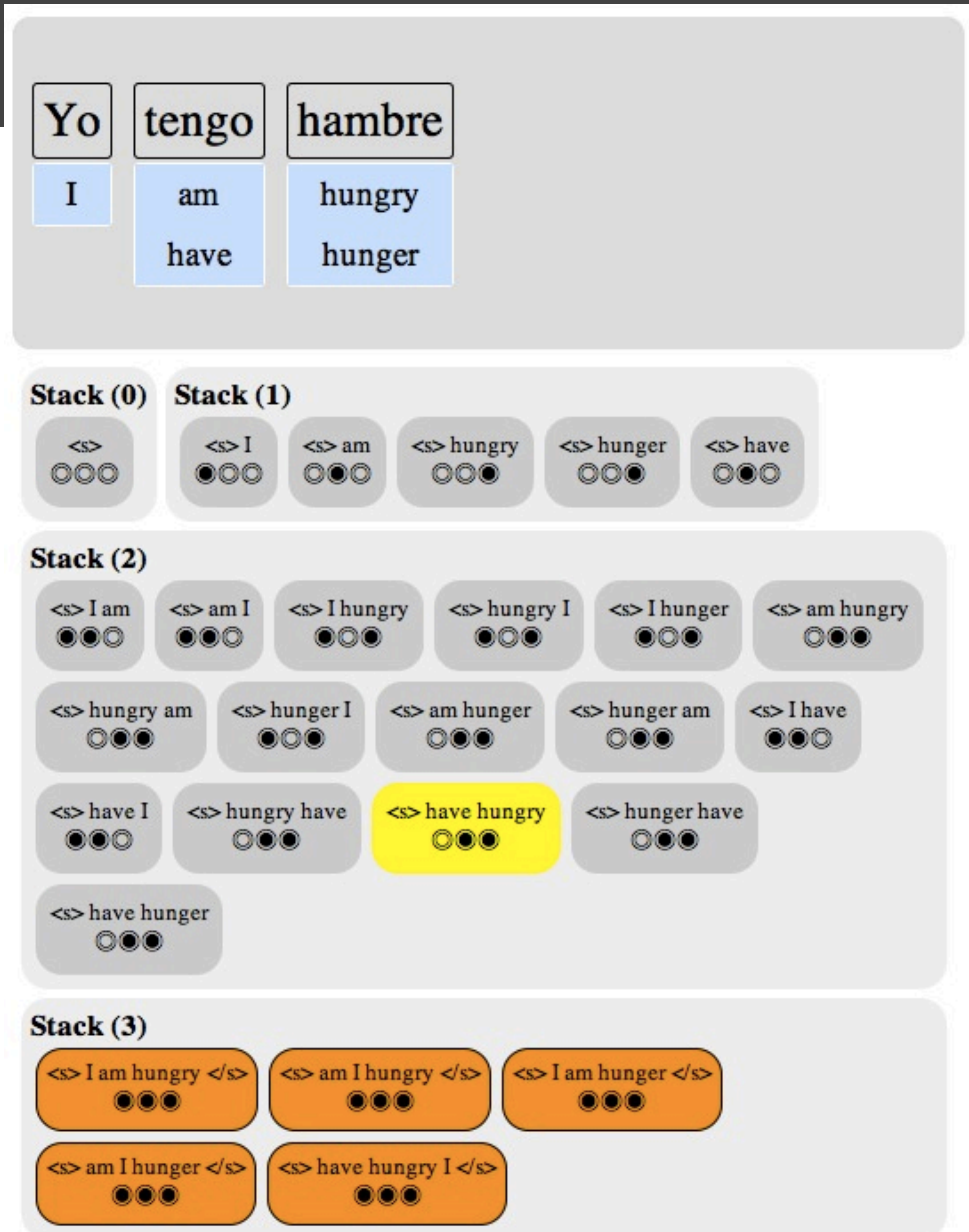
- Leaderboard

THE STORY SO FAR...



SCHEDULE

- TUESDAY
    - stack-based decoding in conception
- TODAY
    - stack-based decoding in practice
    - scoring, dynamic programming, pruning



DECODING

- *the process of producing a translation of a sentence*

- Two main problems:

  - **modeling** – given a pair of sentences, how do we assign a probability to them?

$$P_{(C \rightarrow E)}\left(\begin{array}{c} \text{他们还缺乏国际比} \\ \text{赛的经验.} \\ \text{They still lack experience in} \\ \text{international competitions} \end{array} \right) = \text{high}$$

DECODING

- *the process of producing a translation of a sentence*

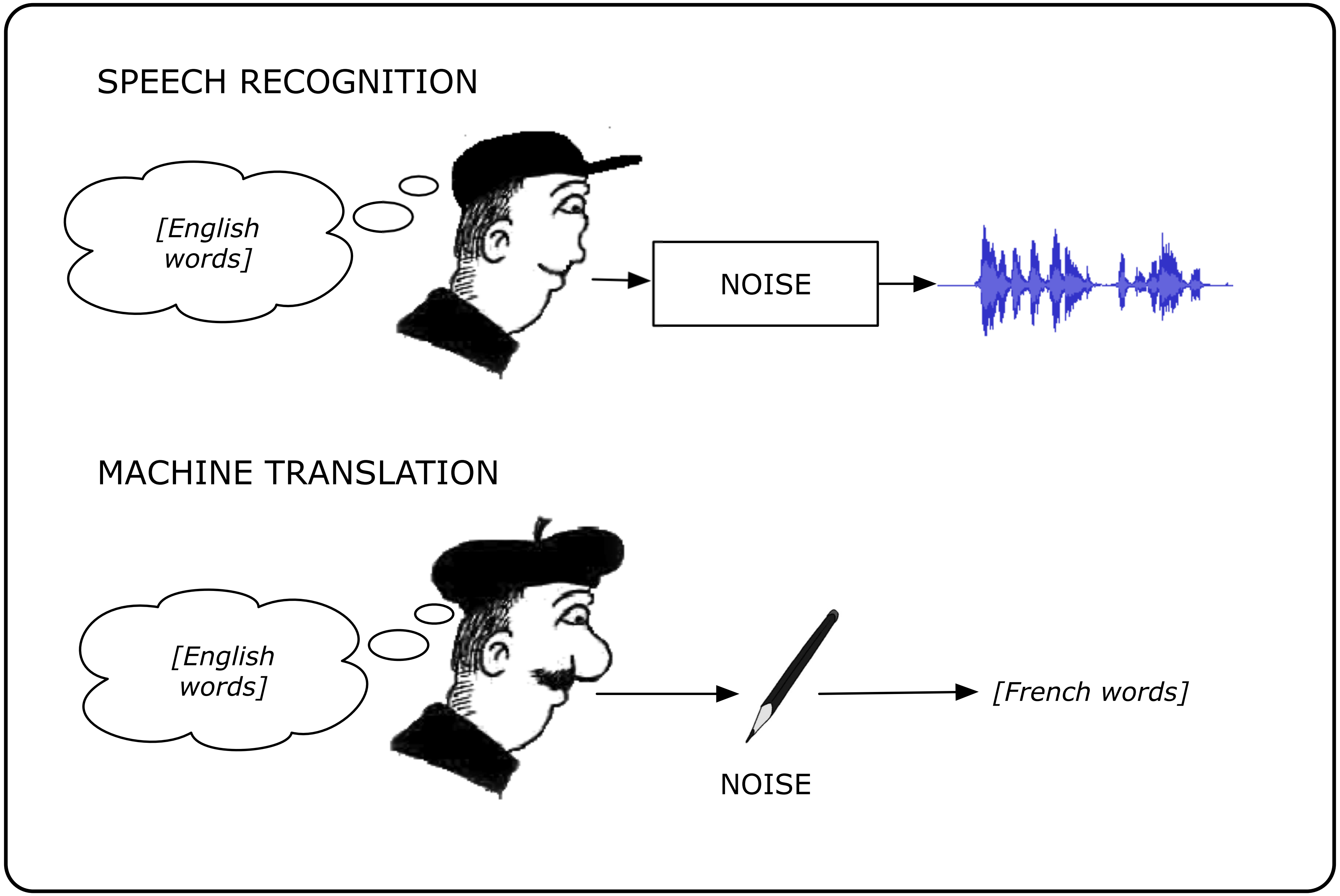
- Two main problems:

  - **modeling** – given a pair of sentences, how do we assign a probability to them?

$$P_{(C \rightarrow E)}\left(\begin{array}{c} \text{他们还缺乏国际比赛的经验.} \\ \text{This is not a good translation of the above sentence.} \end{array} \right) = \text{low}$$

MODEL

- Noisy Channel model $P(e \mid f) \propto P(f \mid e)P(e)$



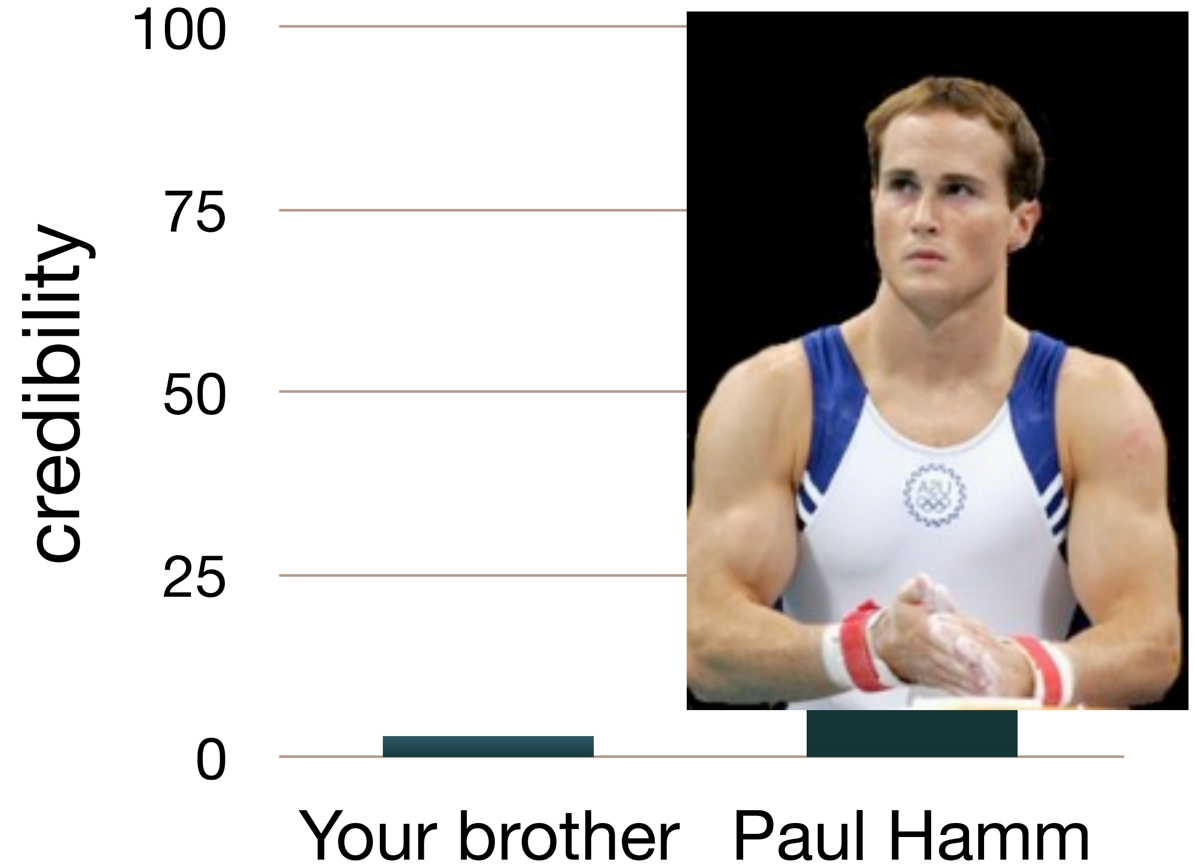
Model Transforms

- Add weights

$$
\begin{aligned}
P(e \mid f) &\propto P(f \mid e) P(e) \\
&\propto P(f \mid e)^{\lambda_1} P(e)^{\lambda_2}
\end{aligned}
$$

Weights

- Why?
    - Just like in real life, where we trust people's claims differently, we will want to learn how to trust different models



"I can do a backflip off this pommel horse"

Model Transforms

- Log space transform

$$
\begin{aligned}
P(e \mid f) &\propto P(f \mid e)P(e) \\
&\propto P(f \mid e)^{\lambda_1} P(e)^{\lambda_2} \\
&= \lambda_1 \log P(f \mid e) + \lambda_2 \log P(e)
\end{aligned}
$$

- Because:

0.0001 * 0.0001 * 0.0001 = **0.000000000001**

log(0.0001) + log(0.0001) + log(0.0001) = **-12**

Model Transforms

- Generalization

$$
\begin{aligned}
P(e \mid f) &\propto P(f \mid e)P(e) \\
&\propto P(f \mid e)^{\lambda_1} P(e)^{\lambda_2} \\
&= \lambda_1 \log P(f \mid e) + \lambda_2 \log P(e) \\
&= \lambda_1 \phi_1(f, e) + \lambda_2 \phi_2(f, e) \\
&= \sum_i \lambda_i \phi_i(f, e)
\end{aligned}
$$

Model

weight

$$e^*, a^* = \underset{e,a}{\operatorname{argmax}} \ \Pr(e, a \mid c)$$

feature function

search

how do we find it?

model

what is a good translation?

A better "fundamental equation" for MT

DECODING

- *the process of producing a translation of a sentence*
- Two main problems:
  - **search** – given a model and a source sentence, how do we find the sentence that the model likes best?
  - impractical: enumerate all sentences, score them
  - stack decoding: assemble translations piece by piece

STACK DECODING

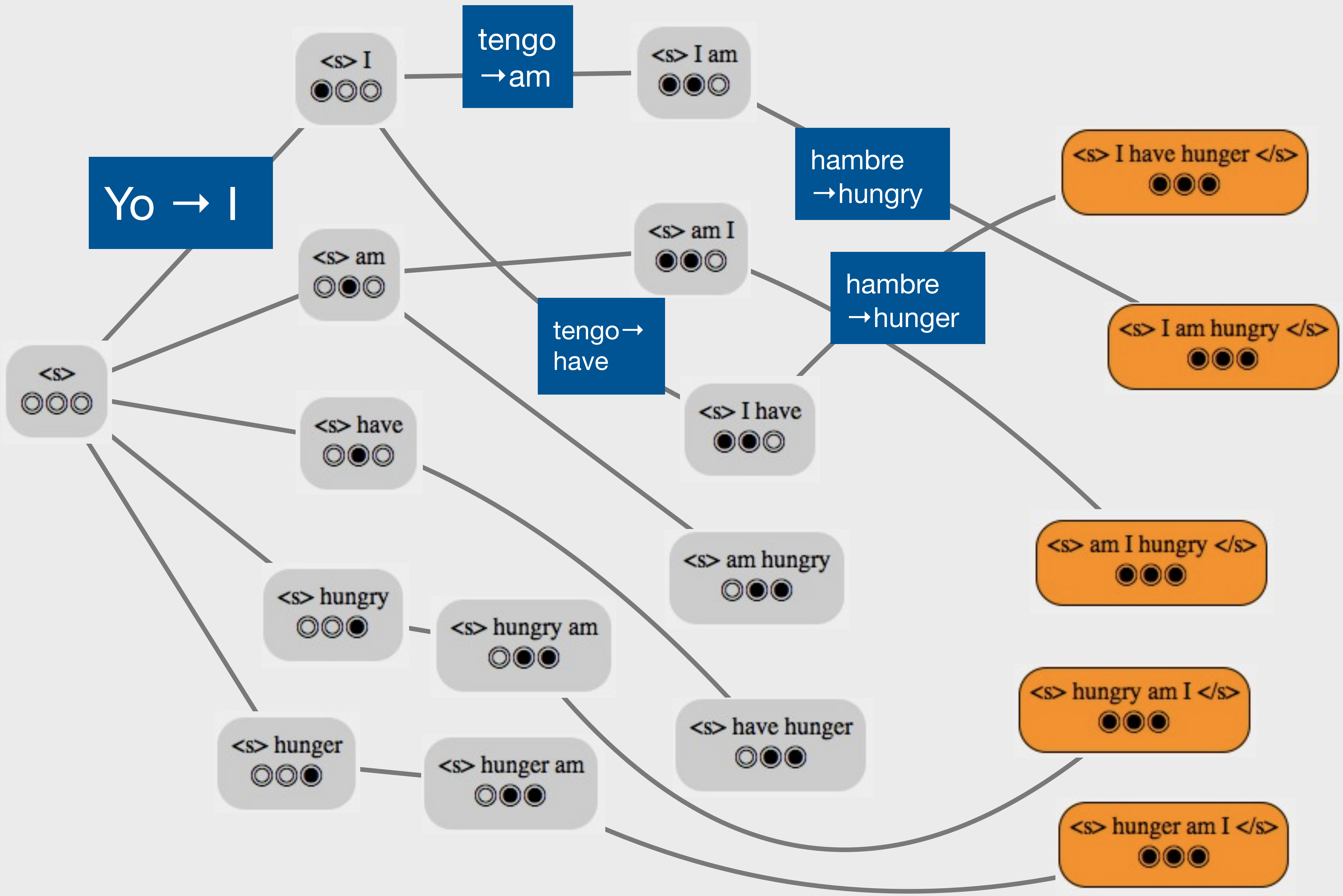
- Start with a list of hypotheses, containing only the empty hypothesis
- For each stack
  - For each hypothesis
    - For each applicable word
      - Extend the hypothesis with the word
      - Place the new hypothesis on the right stack

FACTORING MODELS

- Stack decoding works by extending hypotheses word by word

- These can be arranged into a *search graph* representing the space we search

FACTORING MODELS



FACTORING MODELS

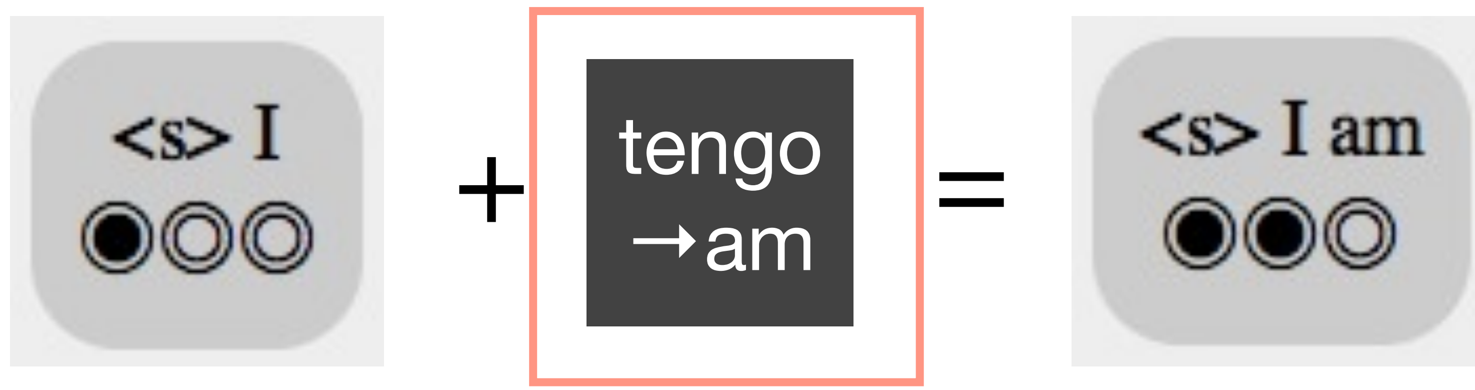
- Stack decoding works by extending hypotheses word by word

- These can be arranged into a *search graph* representing the space we search

- The component models we use need to *factorize* over this graph, and we accumulate the score as we go

FACTORING MODELS

- Example hypothesis creation:



old hypothesis | add word | new hypothesis

- **translation model**: trivial case, since all the words are translated independently

$$\text{hypothesis.score} \mathrel{+}= P_{TM}(\text{am} \mid \text{tengo})$$

- a function of just the word that is added

FACTORING MODELS

- Example hypothesis creation:

old hypothesis | add word | new hypothesis

- **language model**: still easy, since (bigram) language models depend only on the previous word

$$\text{hypothesis.score} \mathrel{+}= P_{LM}(\text{am} \mid \text{I})$$

- a function of the old hyp. and the new word translation

DYNAMIC PROGRAMMING

- We saw Tuesday how huge the search space could get

- Notice anything here?

old hypothesis | add word | new hypothesis

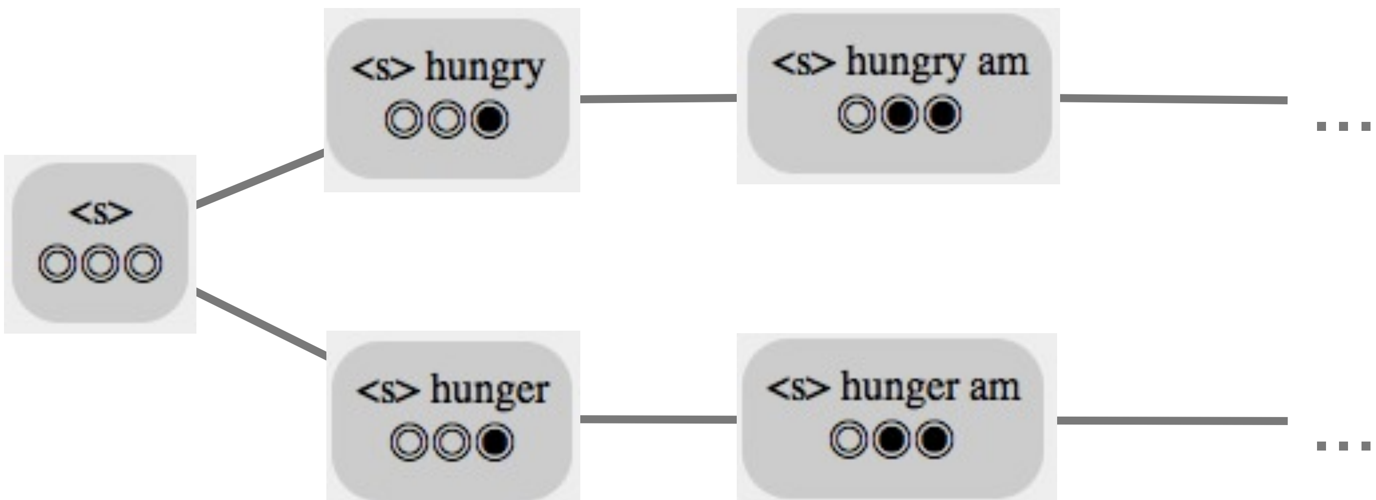
score += $P_{TM}(\text{am} \mid \text{tengo}) + P_{LM}(\text{am} \mid \text{I})$

- (1) <s> *is never used* in computing the scores AND
  (2) <s> is implicit in the graph structure

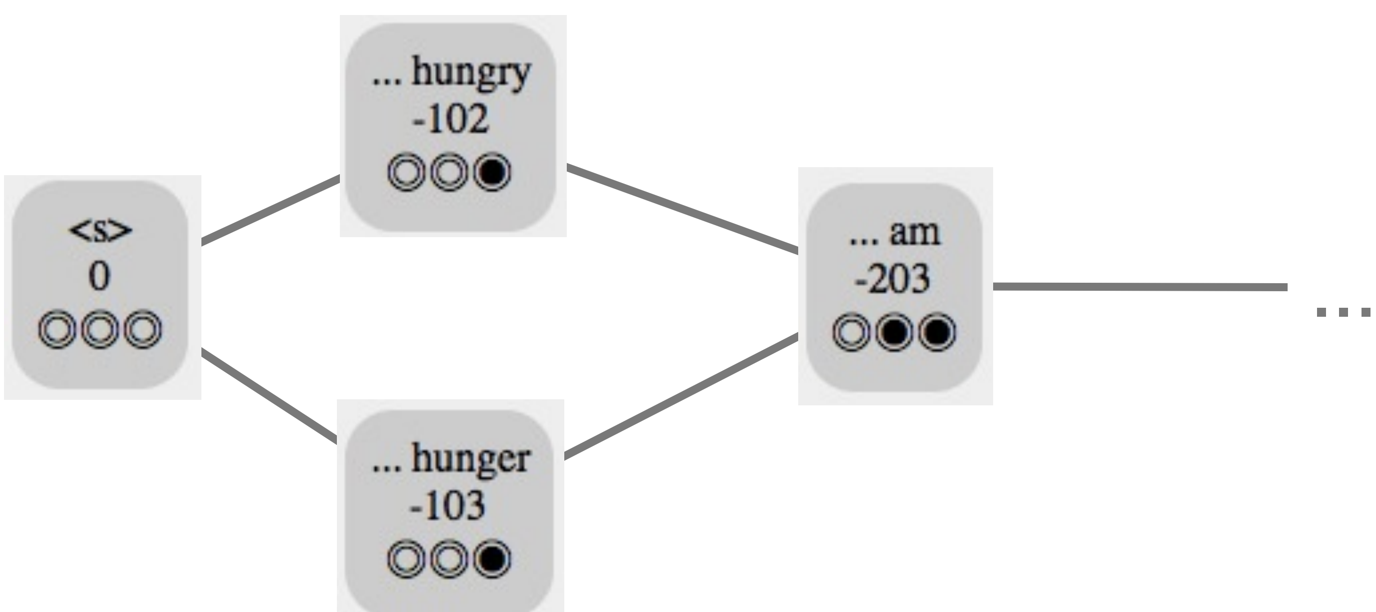
- let's get rid of the extra state!

DYNAMIC PROGRAMMING

- Before



- After



The score of the new hypothesis is the maximum way to compute it

STACK DECODING (WITH DP)

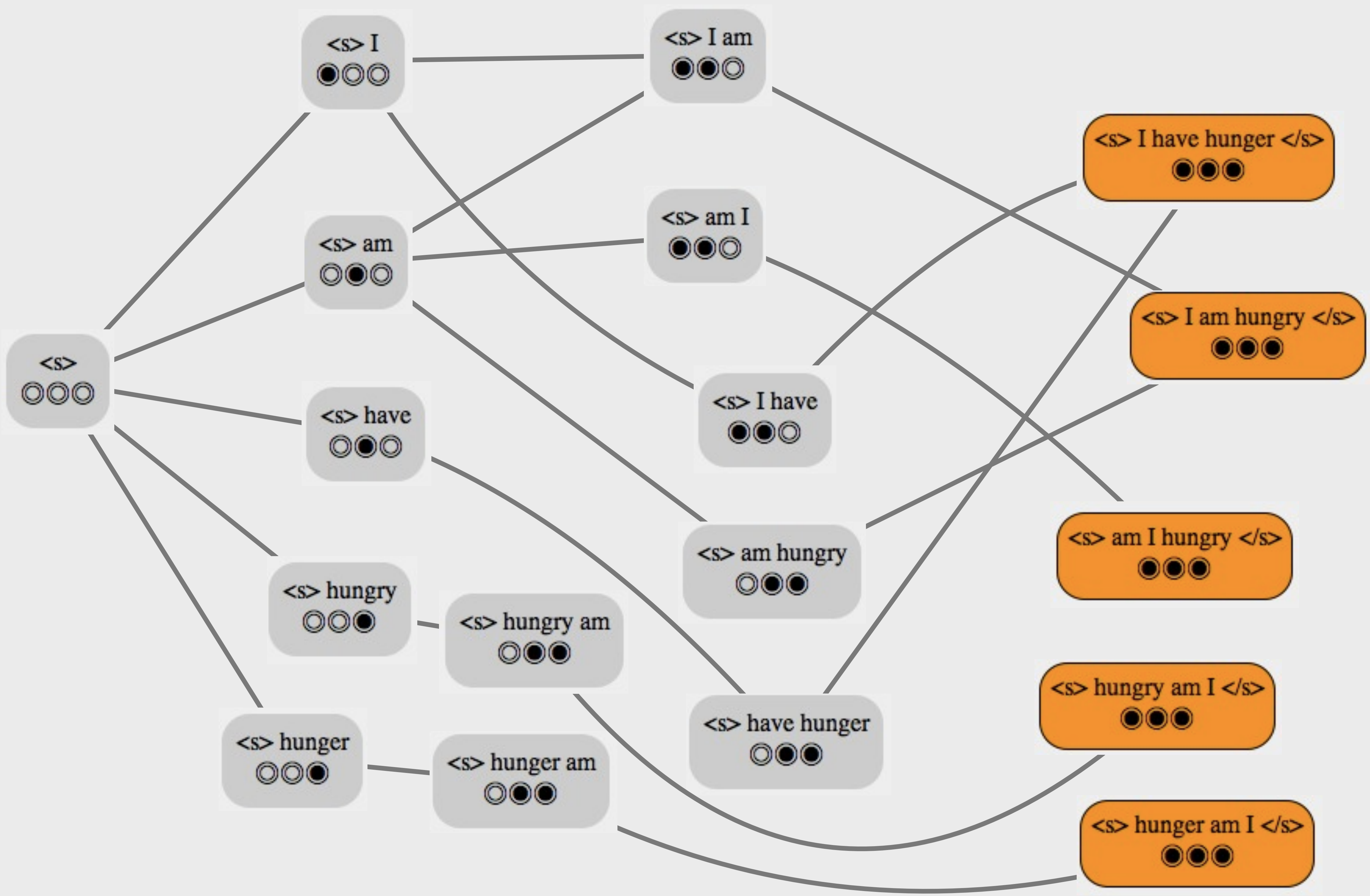
- Start with a list of hypotheses, containing only the empty hypothesis
- For each stack
  - For each hypothesis
    - For each applicable word
      - Extend the hypothesis with the word
      - Place

IF either (1) no equivalent hypothesis exists or (2) this hypothesis has a higher score.

MORE GENERALLY

- What is an "equivalent hypothesis"?
- Hypotheses that match on the minimum necessary state:
  - last word (for language model computation)
  - the score (of the best way to get here)
  - the coverage vector (so we know which words we haven't translated)

OLD GRAPH (BEFORE DP)



PRUNING

- Even with DP, there are still too many hypotheses

- So we prune:

    - histogram pruning: keep only *k* items on each stack

    - threshold pruning: don't keep items that have a score beyond some distance from the most probable item in the stack

Stack decoding (with pruning)

- Start with a list of hypotheses, containing only the empty hypothesis
- For each stack
  - For each hypothesis
    - For each applicable word
      - Extend the hypothesis with the word
      - If it's the best, place the new hypothesis on the right stack (possible replacing an old one)
      - **Prune**

PITFALLS

- *Search errors*
  - *def:* not finding the model's highest-scoring translation
  - this happens when the shortcuts we took excluded good hypotheses
- *Model errors*
  - *def:* the model's best hypothesis isn't a good one
  - depends on some metric (e.g., human judgment)

Activity

http://cs.jhu.edu/~post/mt-class/stack-decoder/

Instructions (10 minutes)
In groups or alone, find the highest-scoring translation under our model under different stack size and reordering settings.

Are there any search or model errors?

IMPORTANT CONCEPTS

- generalized weighted feature function formulation
- decoding as graph search
- factorized models for scoring edges
- **dynamic programming**
- pruning (histogram, beam/threshold)

NOT DISCUSSED (BUT IMPORTANT)

- Outside (future) cost estimates and A* search

- Computational complexity