Neural Machine Translation Decoding

Philipp Koehn

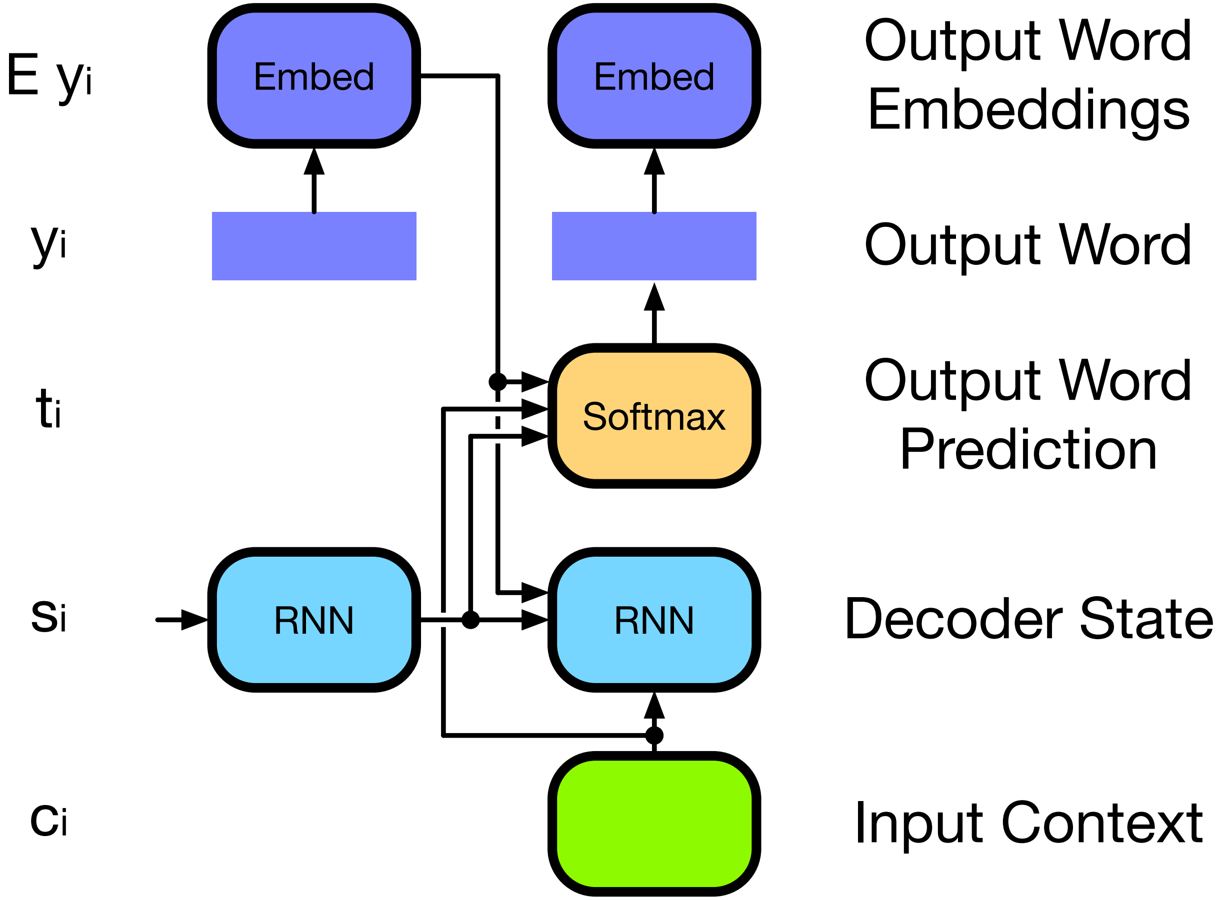
Inference

- Given a trained model

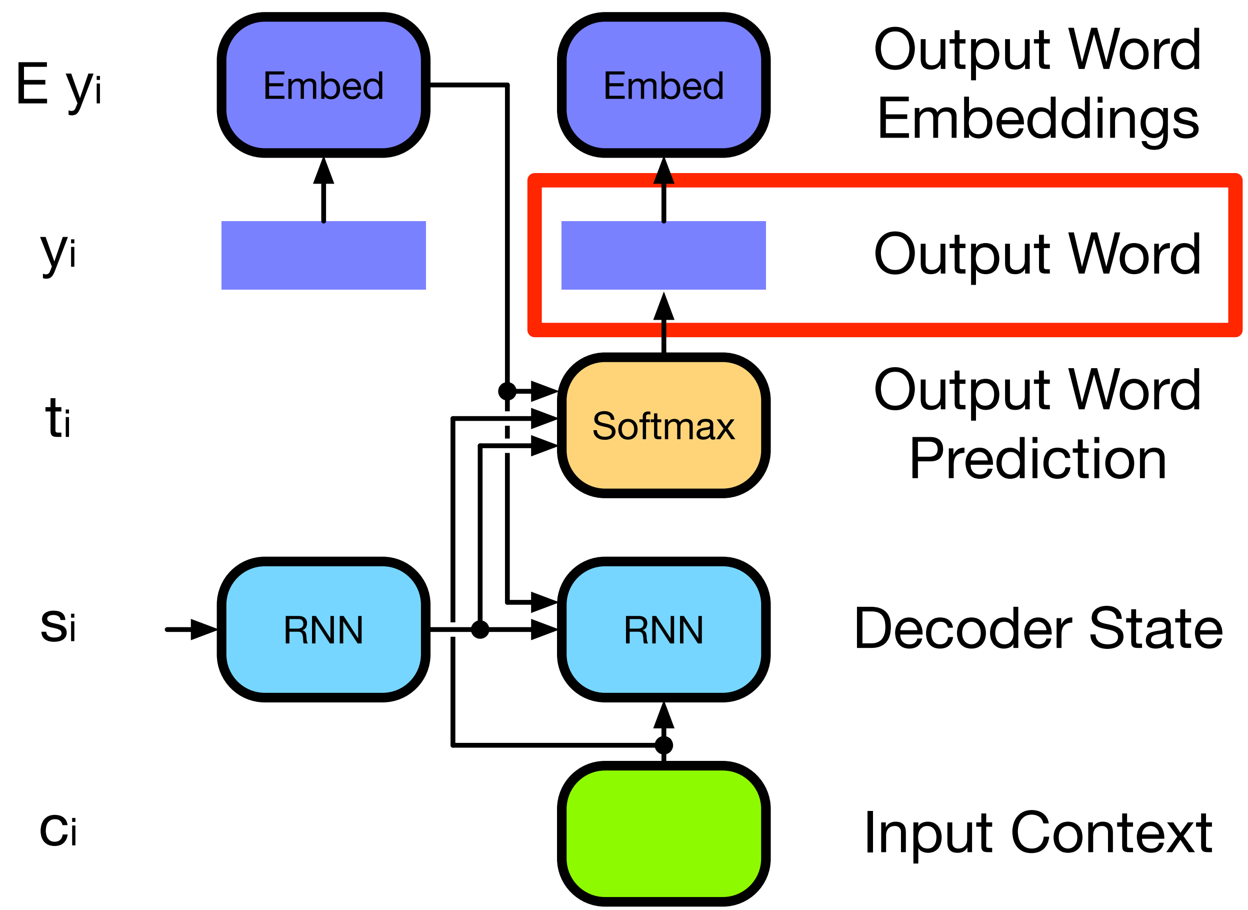
  ... we now want to translate test sentences

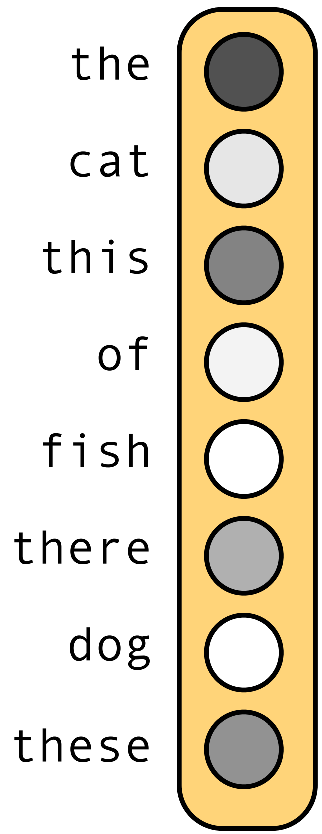
- We only need execute the "forward" step in the computation graph

Word Prediction

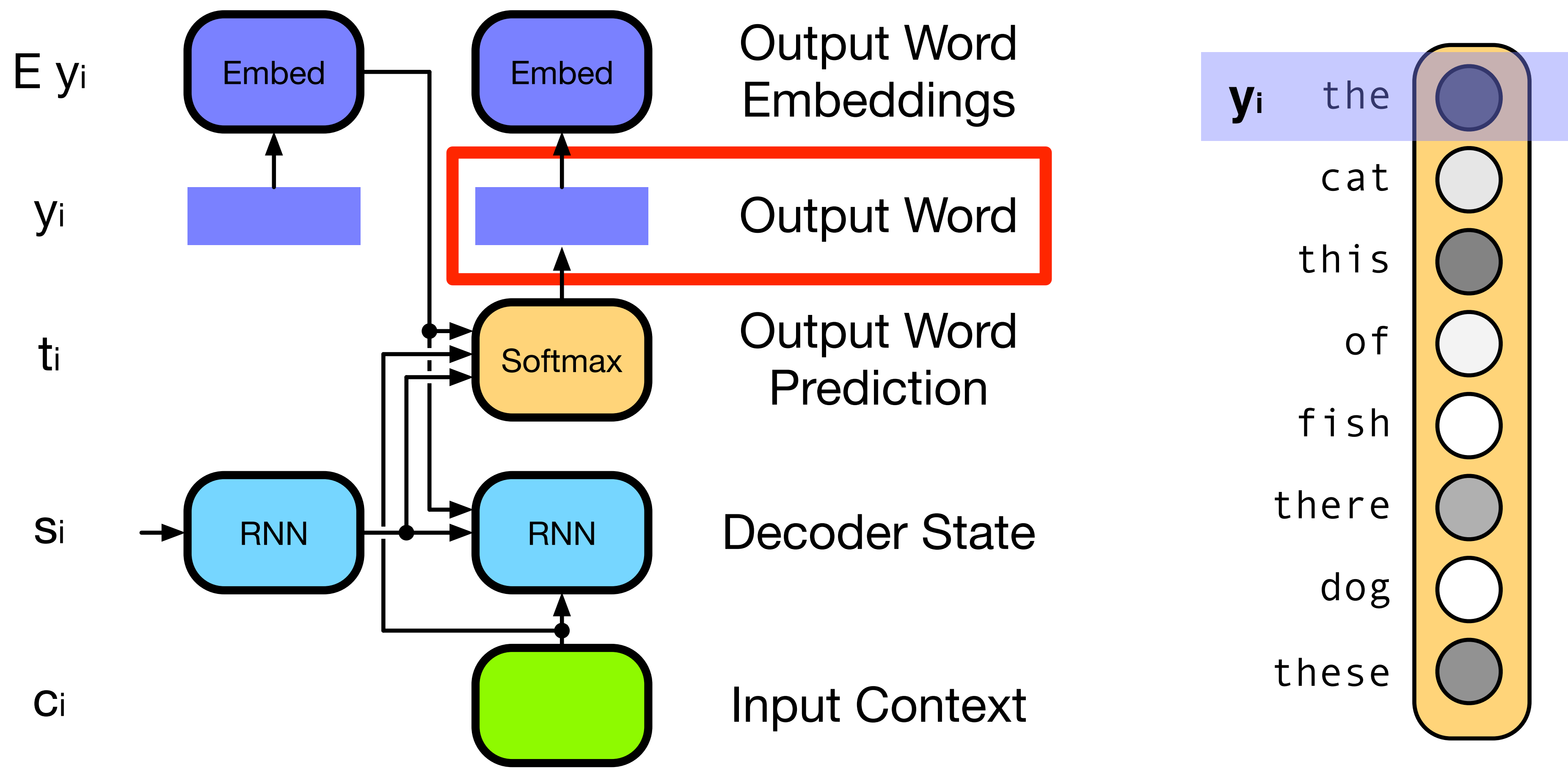


Selected Word

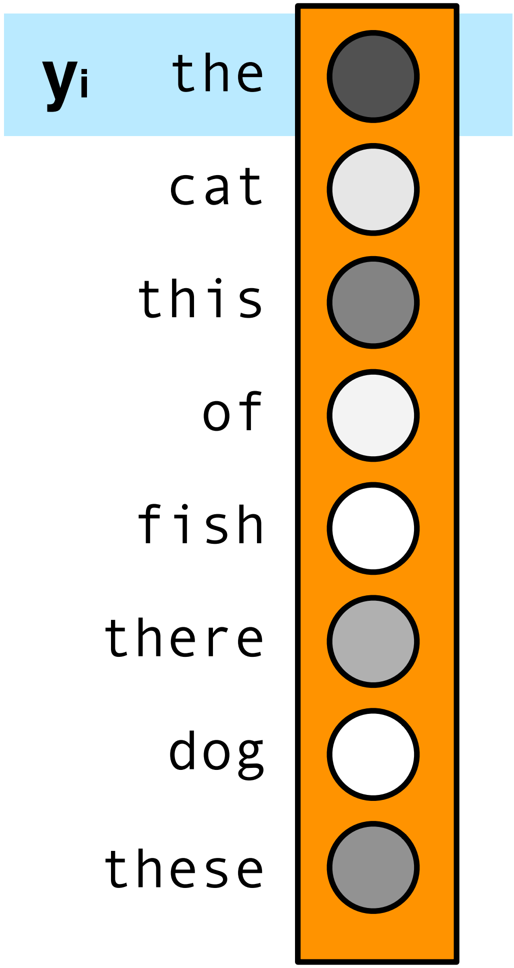




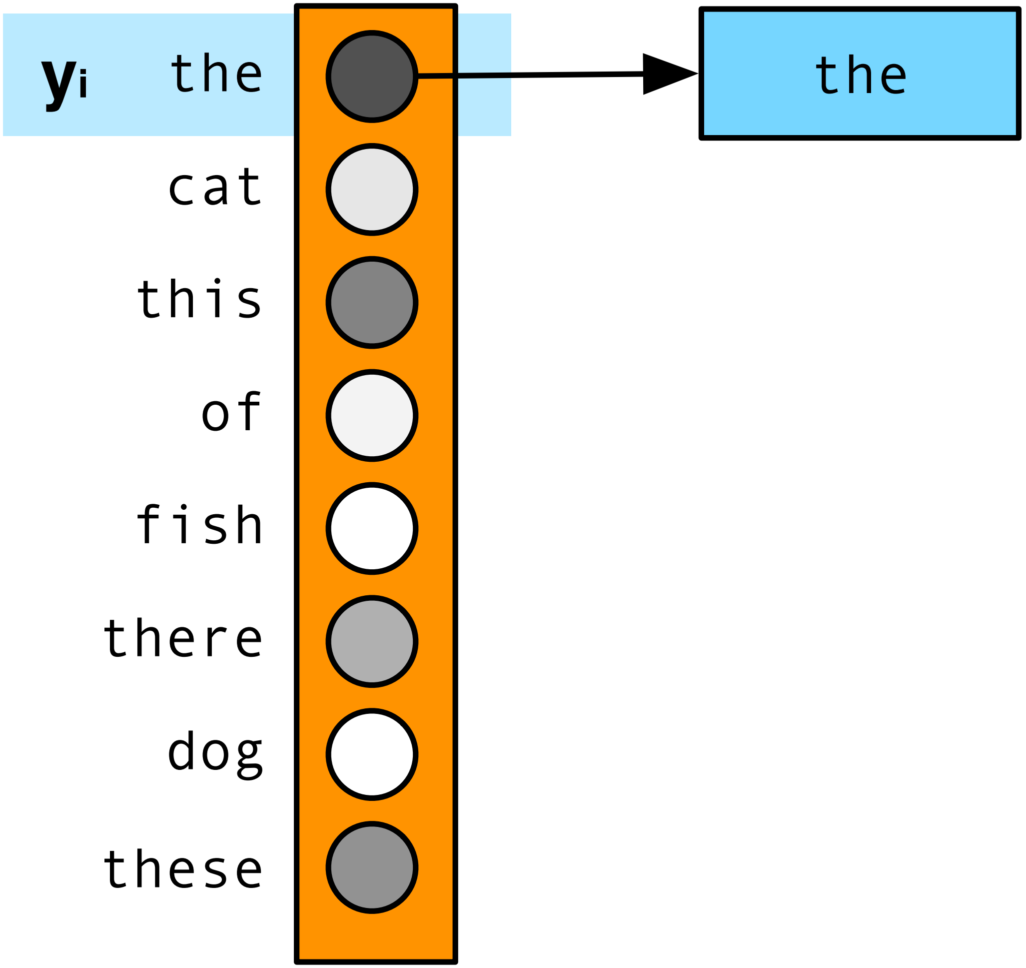
Embedding


E yi
yi
ti
si
ci
Embed
Embed
Softmax
RNN
RNN
Output Word Embeddings
Output Word
Output Word Prediction
Decoder State
Input Context
yi
the
cat
this
of
fish
there
dog
these

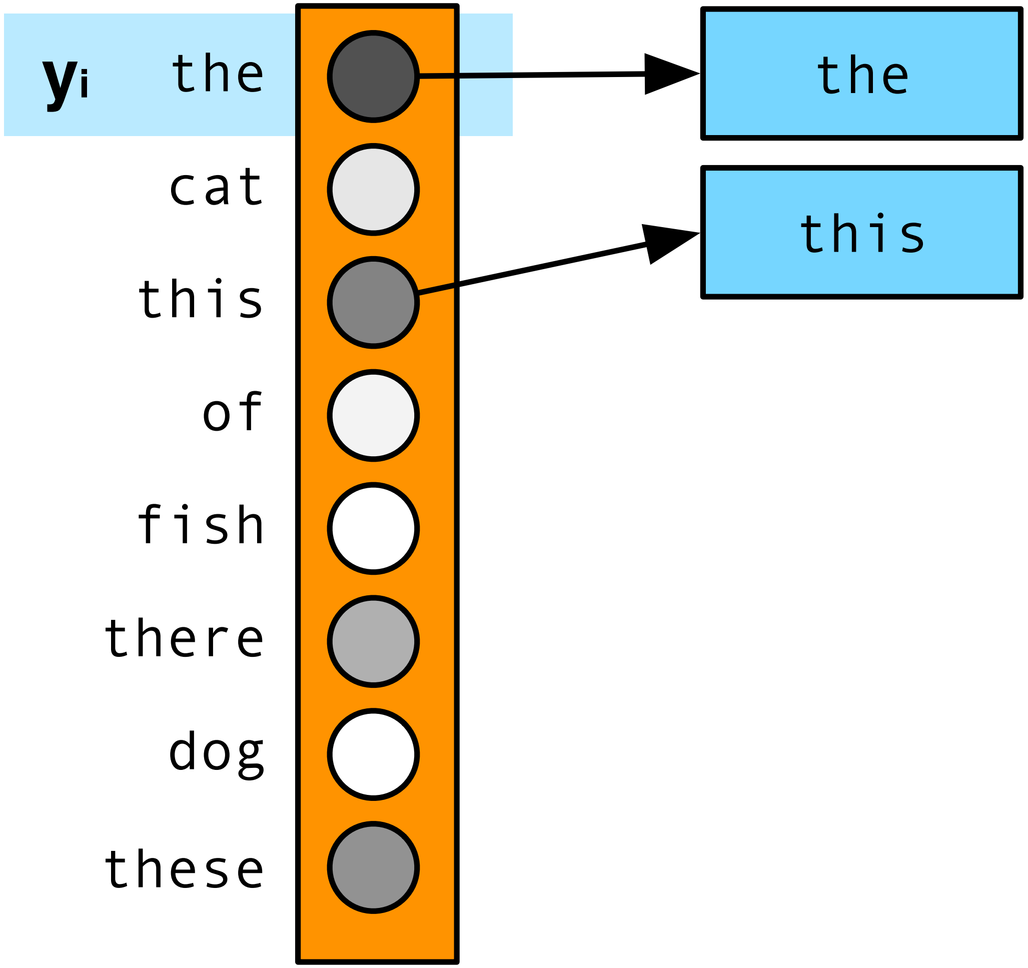
Distribution of Word Predictions



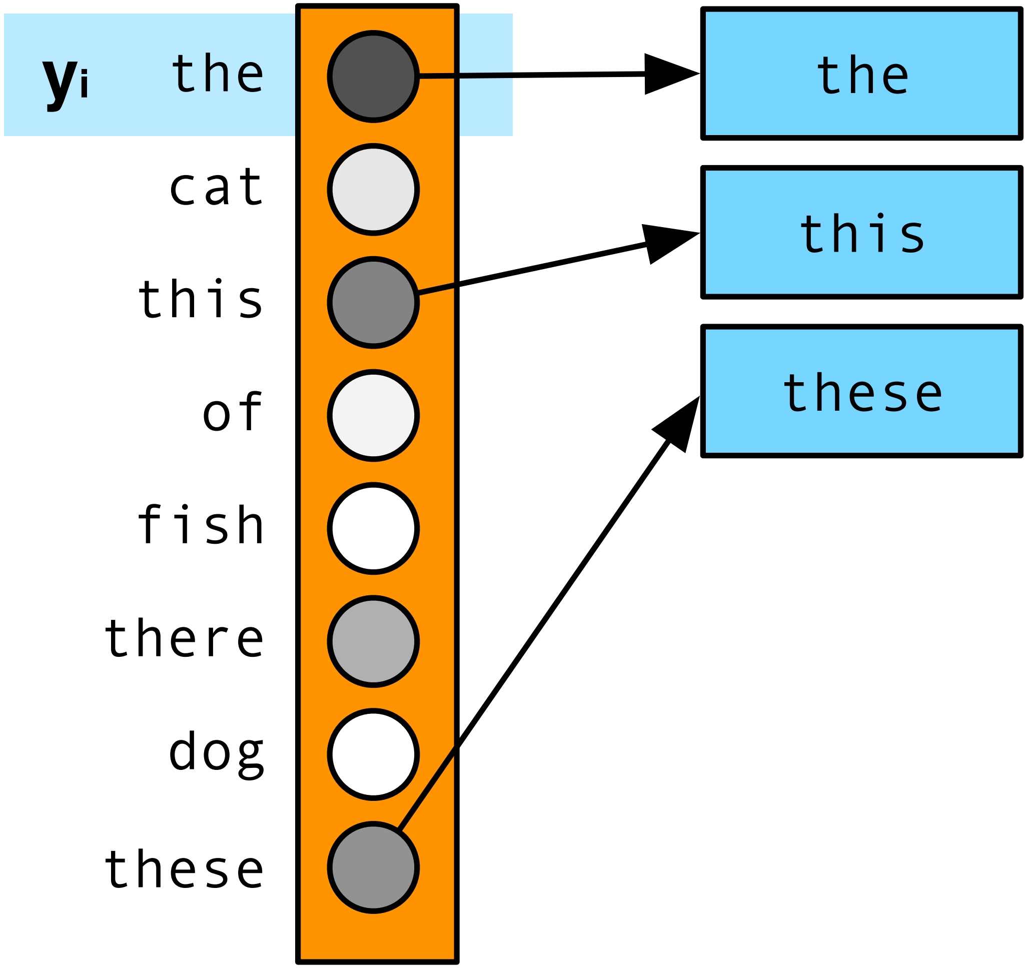
Select Best Word



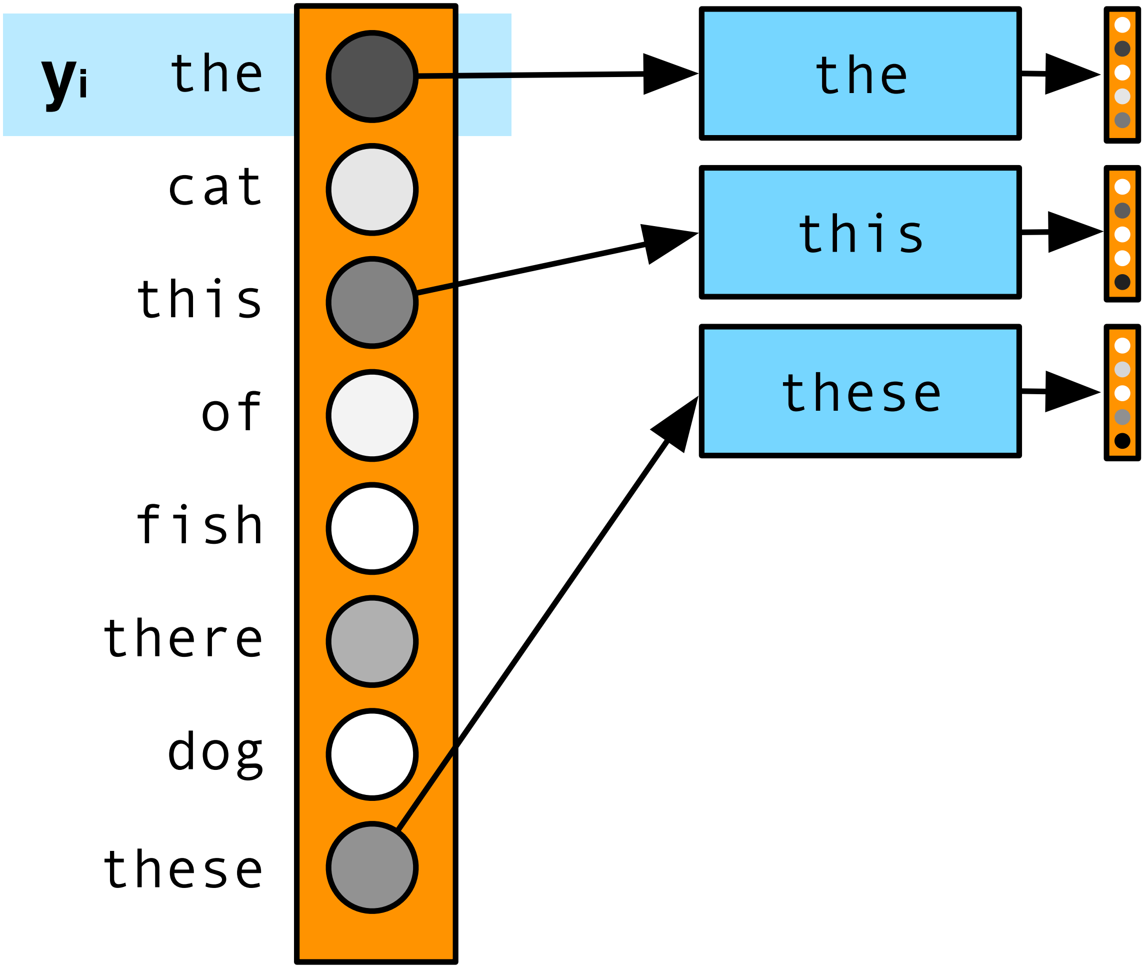
Select Second Best Word



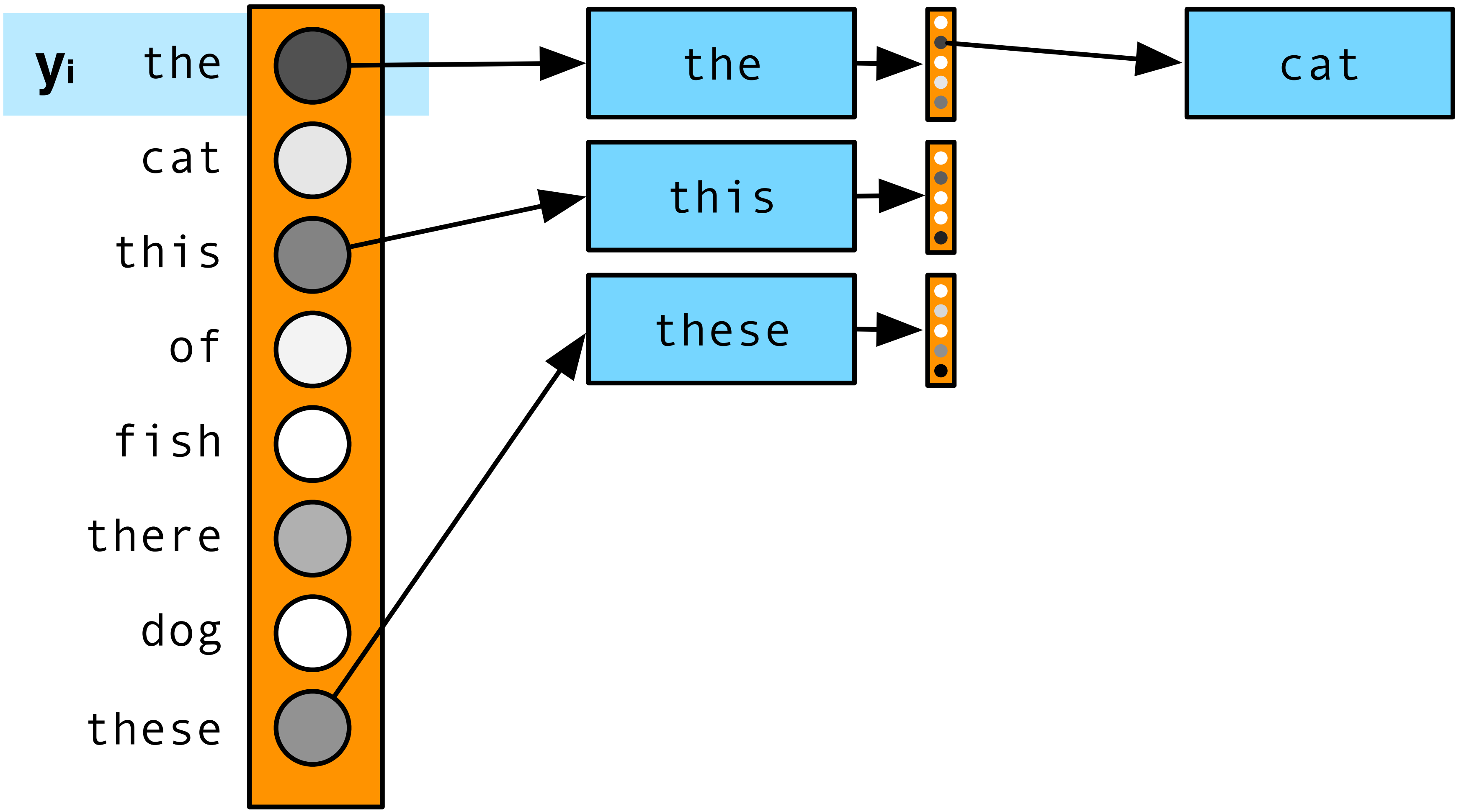
Select Third Best Word



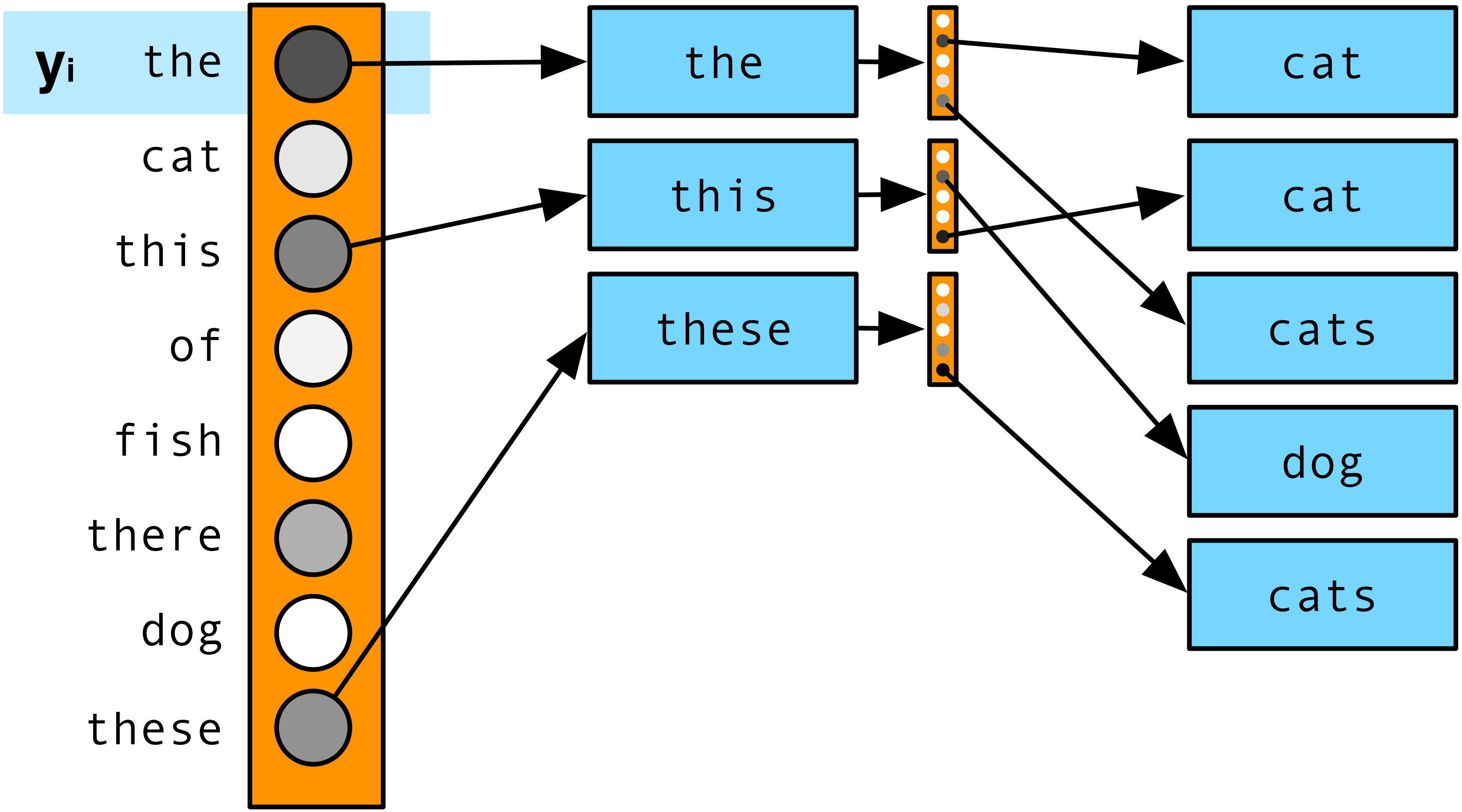
Use Selected Word for Next Predictions



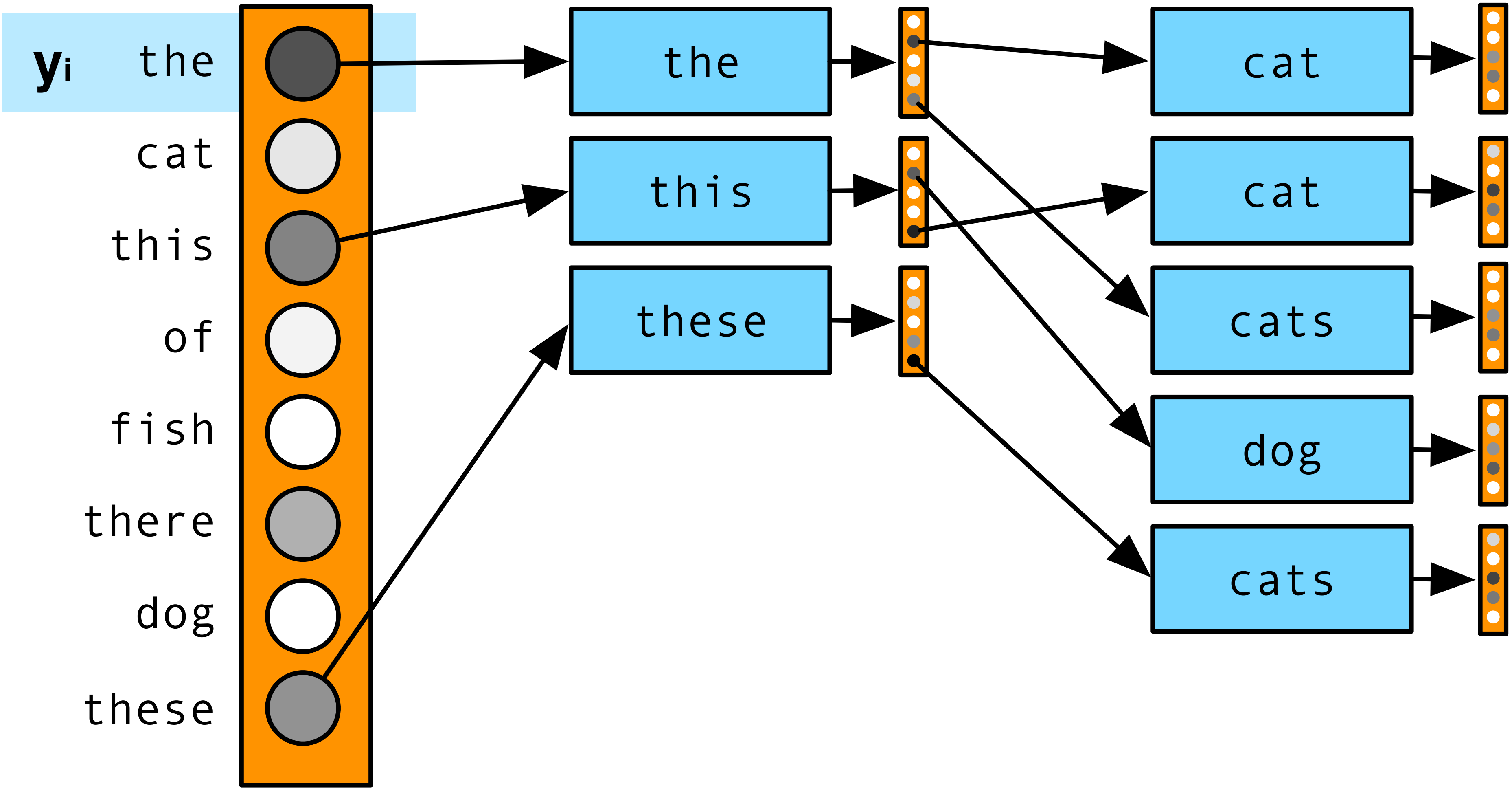
Select Best Continuation



Select Next Best Continuations

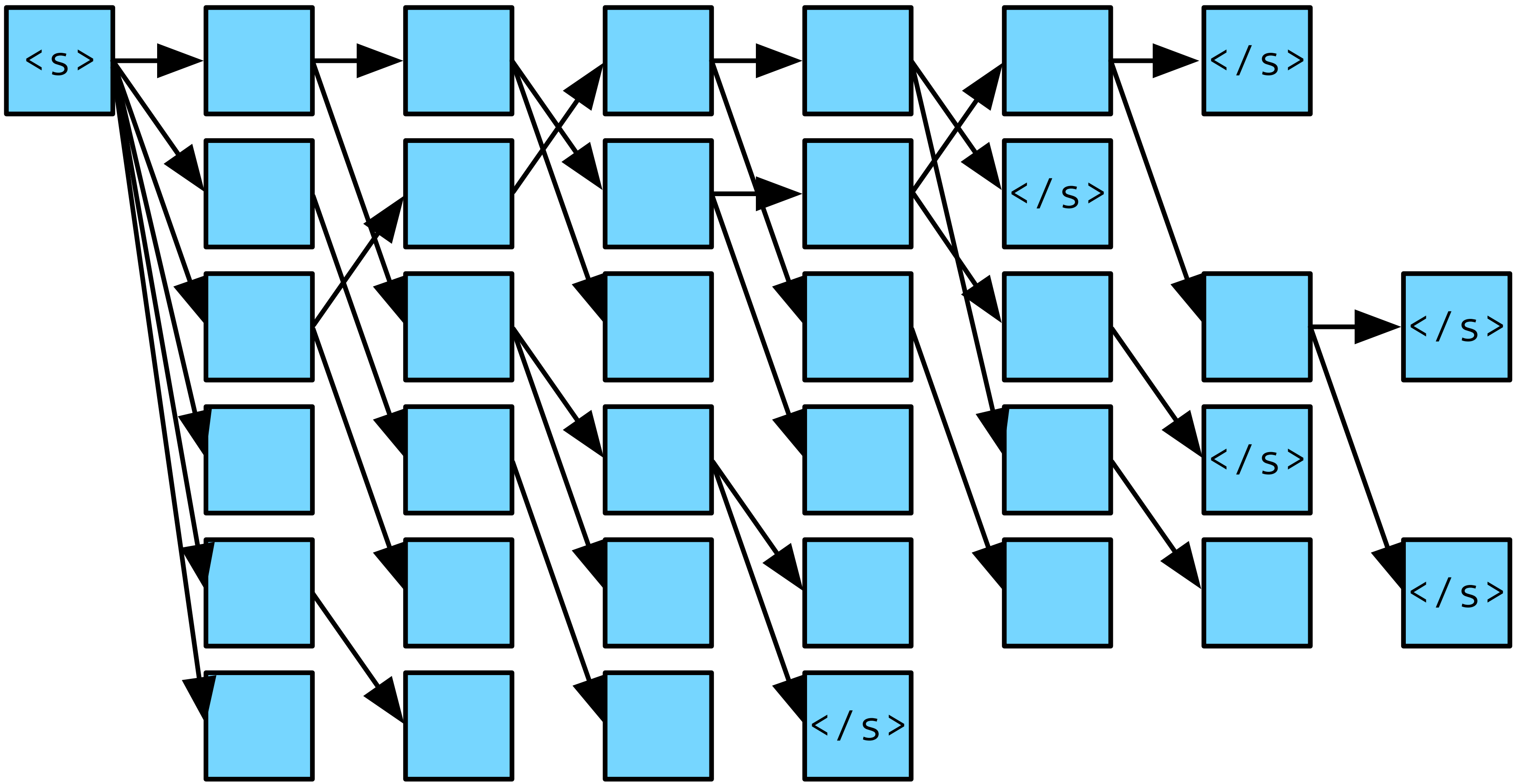


Continue...

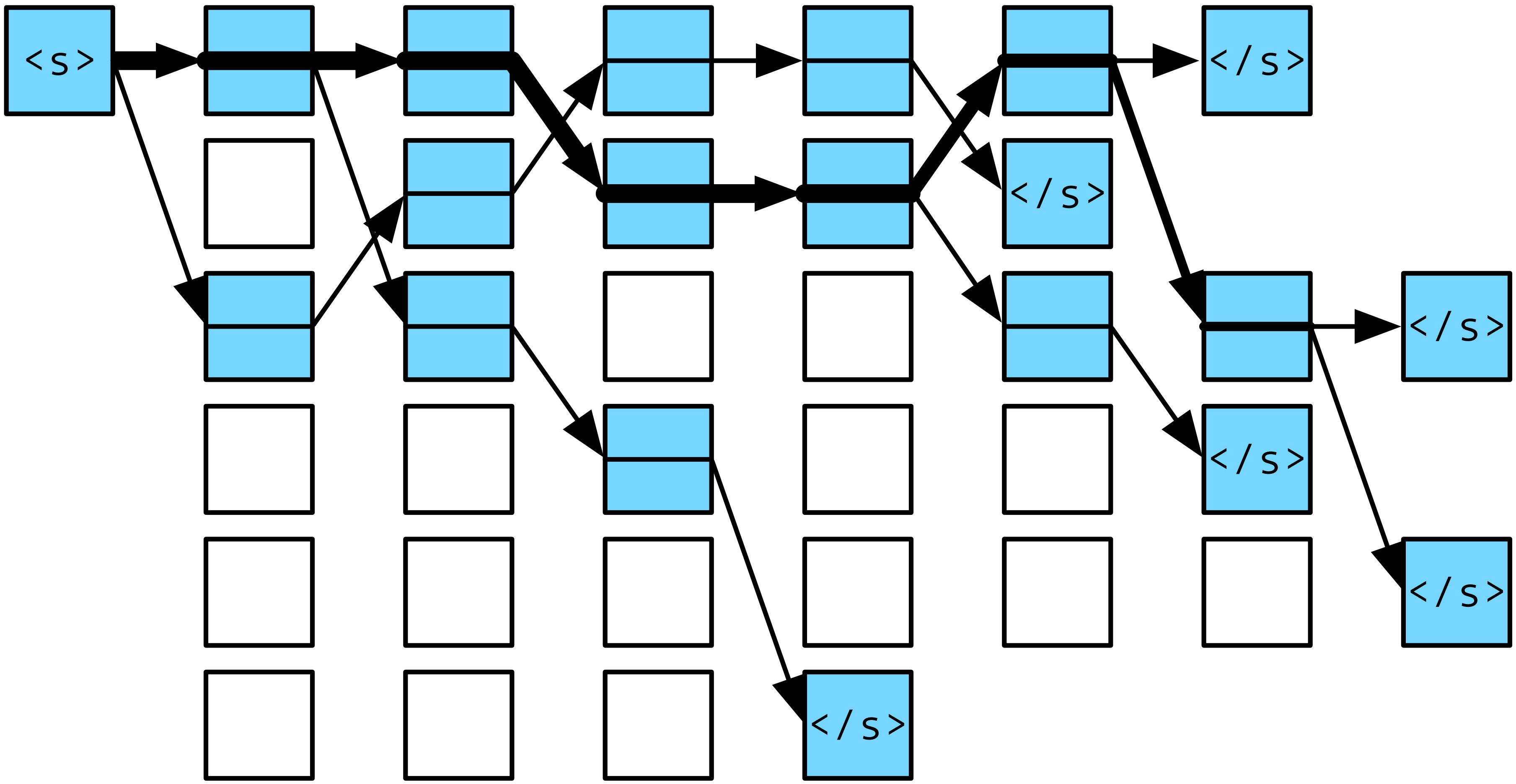


y_i
the
cat
this
of
fish
there
dog
these
the
this
these
cat
cat
cats
dog
cats

Beam Search



Best Paths



Beam Search Details

- Normalize score by length
- No recombination (paths cannot be merged)

Output Word Predictions

Input Sentence: *ich glaube aber auch , er ist clever genug um seine Aussagen vage genug zu halten , so dass sie auf verschiedene Art und Weise interpretiert werden können .*

Best		Alternatives
but	(42.1%)	*however* (25.3%), *I* (20.4%), *yet* (1.9%), *and* (0.8%), *nor* (0.8%), ...
I	(80.4%)	*also* (6.0%), *,* (4.7%), *it* (1.2%), *in* (0.7%), *nor* (0.5%), *he* (0.4%), ...
also	(85.2%)	*think* (4.2%), *do* (3.1%), *believe* (2.9%), *,* (0.8%), *too* (0.5%), ...
believe	(68.4%)	*think* (28.6%), *feel* (1.6%), *do* (0.8%), ...
he	(90.4%)	*that* (6.7%), *it* (2.2%), *him* (0.2%), ...
is	(74.7%)	*'s* (24.4%), *has* (0.3%), *was* (0.1%), ...
clever	(99.1%)	*smart* (0.6%), ...
enough	(99.9%)	
to	(95.5%)	*about* (1.2%), *for* (1.1%), *in* (1.0%), *of* (0.3%), *around* (0.1%), ...
keep	(69.8%)	*maintain* (4.5%), *hold* (4.4%), *be* (4.2%), *have* (1.1%), *make* (1.0%), ...
his	(86.2%)	*its* (2.1%), *statements* (1.5%), *what* (1.0%), *out* (0.6%), *the* (0.6%), ...
statements	(91.9%)	*testimony* (1.5%), *messages* (0.7%), *comments* (0.6%), ...
vague	(96.2%)	*v@@* (1.2%), *in* (0.6%), *ambiguous* (0.3%), ...
enough	(98.9%)	*and* (0.2%), ...
so	(51.1%)	*,* (44.3%), *to* (1.2%), *in* (0.6%), *and* (0.5%), *just* (0.2%), *that* (0.2%), ...
they	(55.2%)	*that* (35.3%), *it* (2.5%), *can* (1.6%), *you* (0.8%), *we* (0.4%), *to* (0.3%), ...
can	(93.2%)	*may* (2.7%), *could* (1.6%), *are* (0.8%), *will* (0.6%), *might* (0.5%), ...
be	(98.4%)	*have* (0.3%), *interpret* (0.2%), *get* (0.2%), ...
interpreted	(99.1%)	*interpre@@* (0.1%), *constru@@* (0.1%), ...
in	(96.5%)	*on* (0.9%), *differently* (0.5%), *as* (0.3%), *to* (0.2%), *for* (0.2%), *by* (0.1%), ...
different	(41.5%)	*a* (25.2%), *various* (22.7%), *several* (3.6%), *ways* (2.4%), *some* (1.7%), ...
ways	(99.3%)	*way* (0.2%), *manner* (0.2%), ...
.	(99.2%)	</s> (0.2%), *,* (0.1%), ...
</s>	(100.0%)	

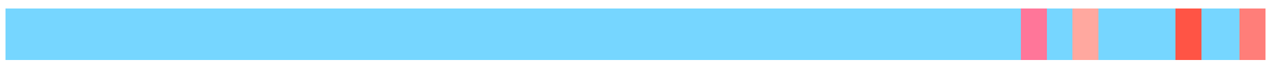
ensembling

Ensembling

- Train multiple models

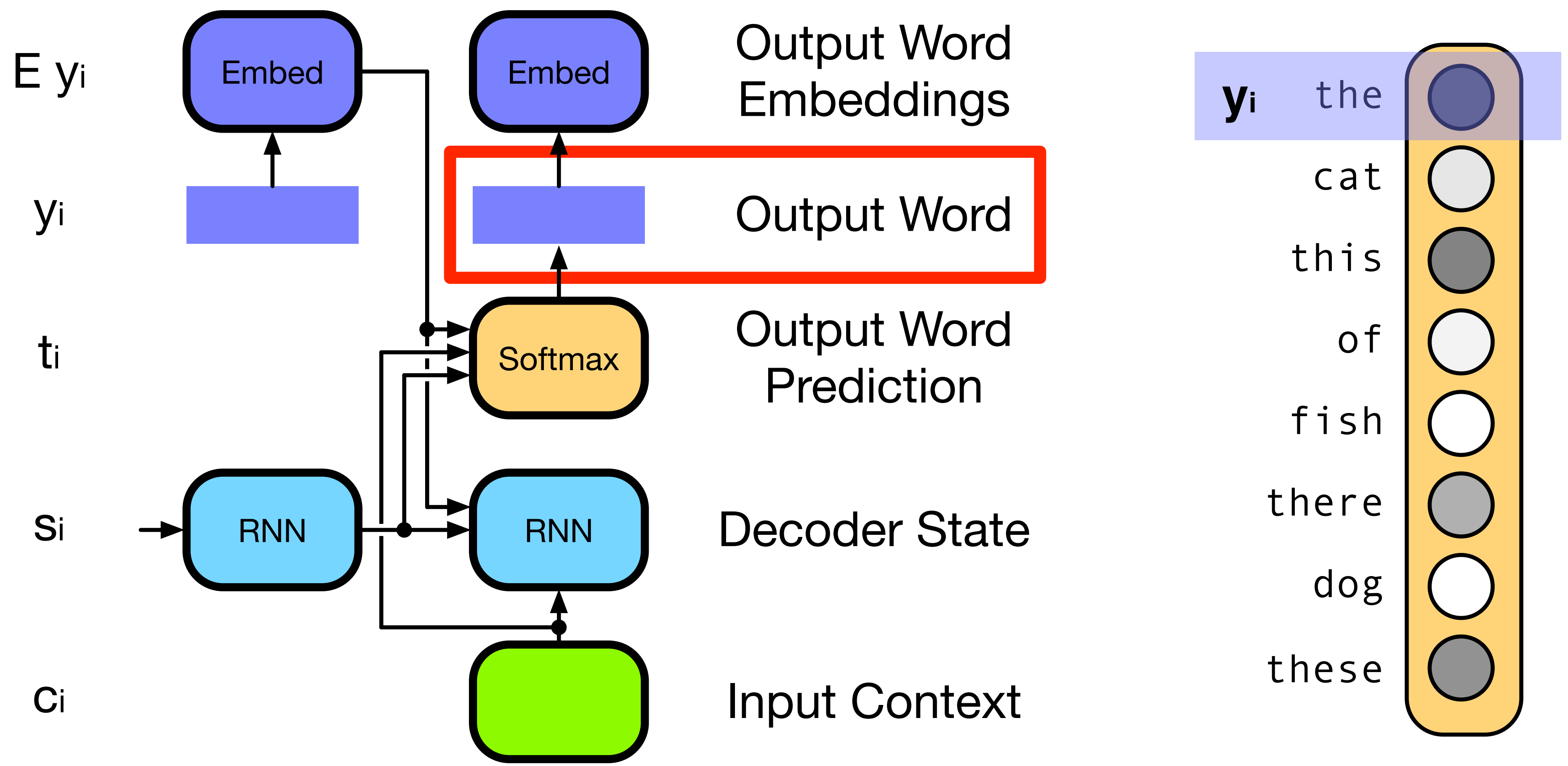
- Say, by different random initializations

- Or, by using model dumps from earlier iterations

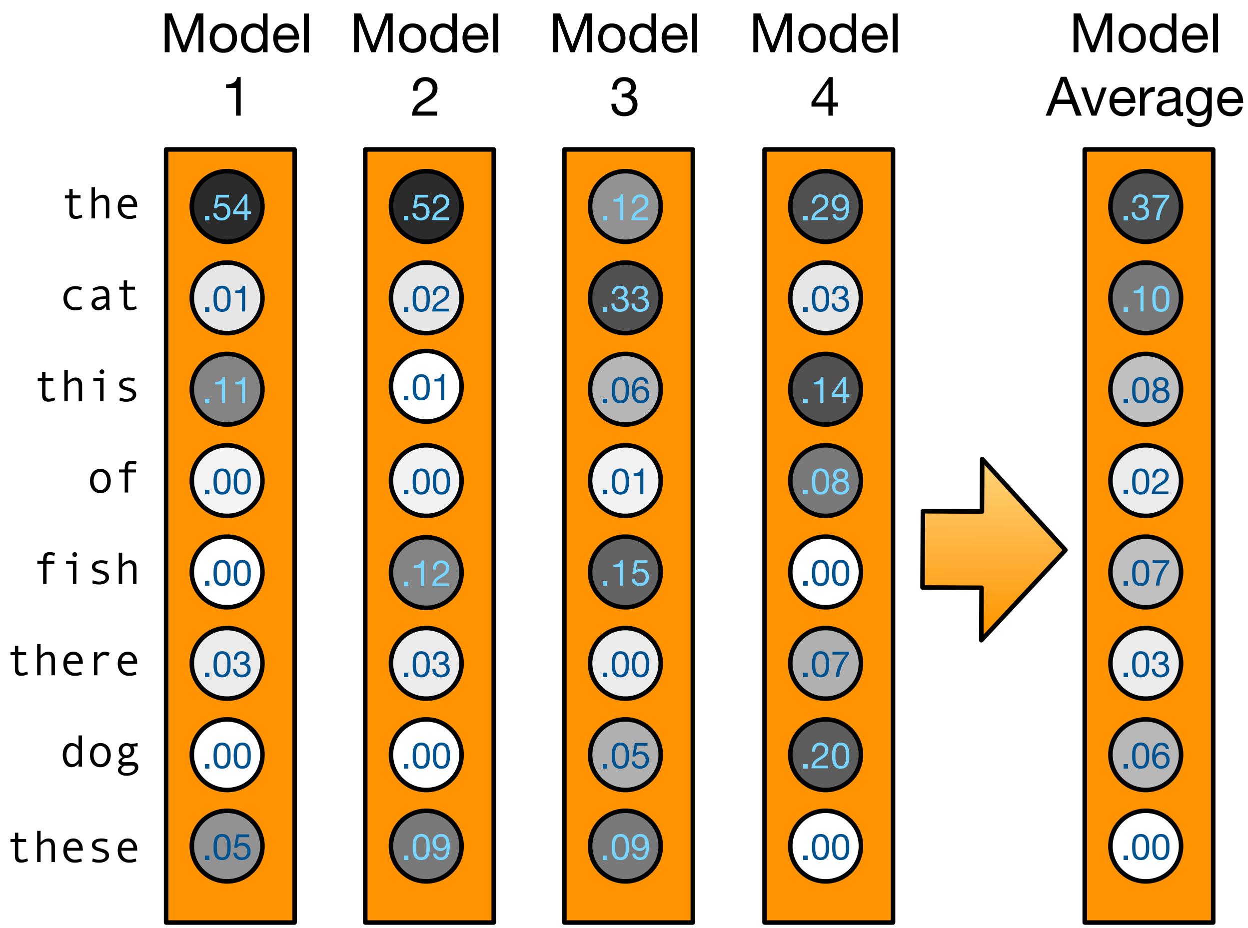


(most recent, or interim models with highest validation score)

Decoding with Single Model



Combine Predictions



Ensembling

- Surprisingly reliable method in machine learning

- Long history, many variants:
  bagging, ensemble, model averaging, system combination, ...

- Works because errors are random, but correct decisions unique

reranking

Right-to-Left Inference

- Neural machine translation generates words right to left (L2R)

$$\text{the} \rightarrow \text{cat} \rightarrow \text{is} \rightarrow \text{in} \rightarrow \text{the} \rightarrow \text{bag} \rightarrow .$$

- But it could also generate them right to left (R2L)

$$\text{the} \leftarrow \text{cat} \leftarrow \text{is} \leftarrow \text{in} \leftarrow \text{the} \leftarrow \text{bag} \leftarrow .$$

Obligatory notice: Some languages (Arabic, Hebrew, ...) have writing systems that are right-to-left, so the use of "right-to-left" is not precise here.

Right-to-Left Reranking

- Train both L2R and R2L model
- Score sentences with both
  $\Rightarrow$ use both left and right context during translation
- Only possible once full sentence produced $\rightarrow$ re-ranking
  1. generate n-best list with L2R model
  2. score candidates in n-best list with R2L model
  3. chose translation with best average score