Neural Machine Translation Decoding

Philipp Koehn

8 October 2020

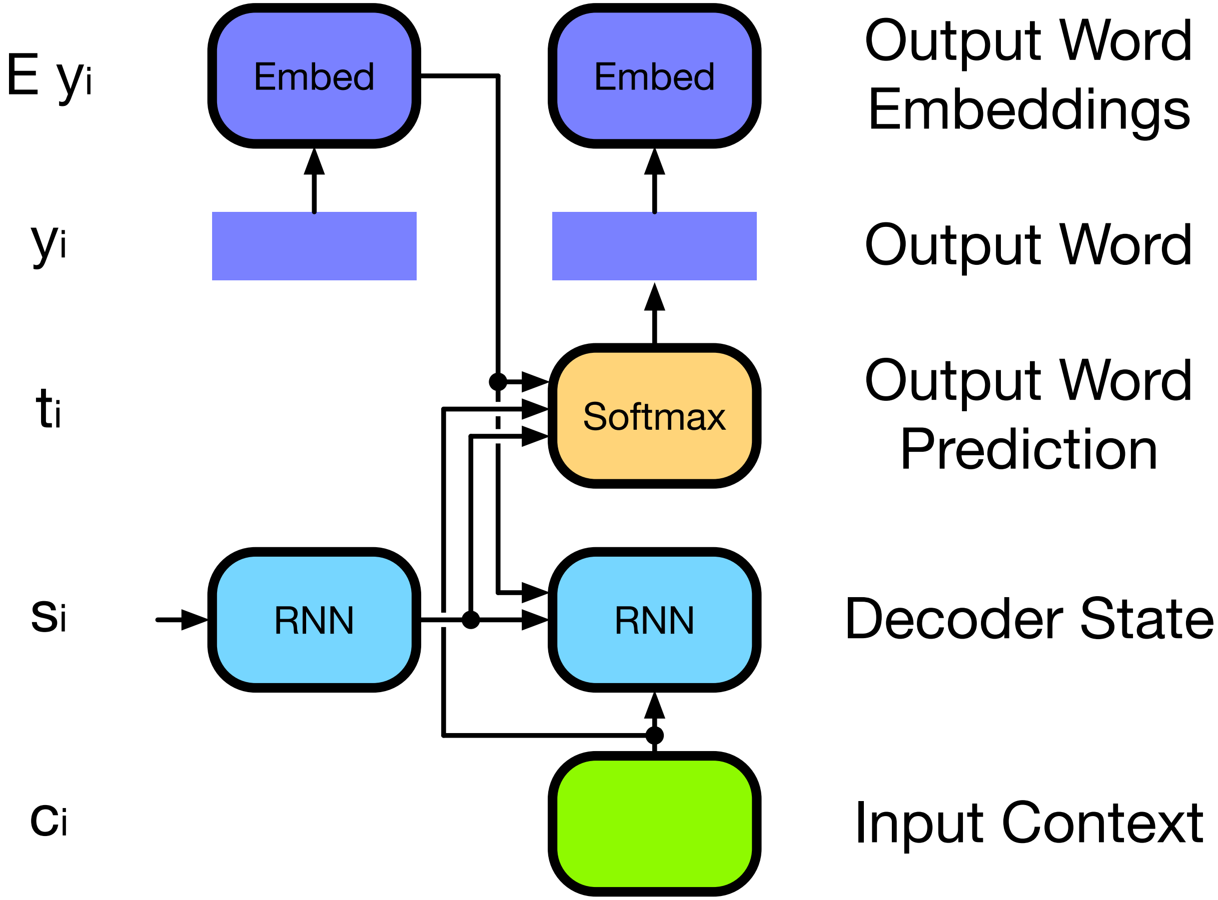
Inference

- Given a trained model

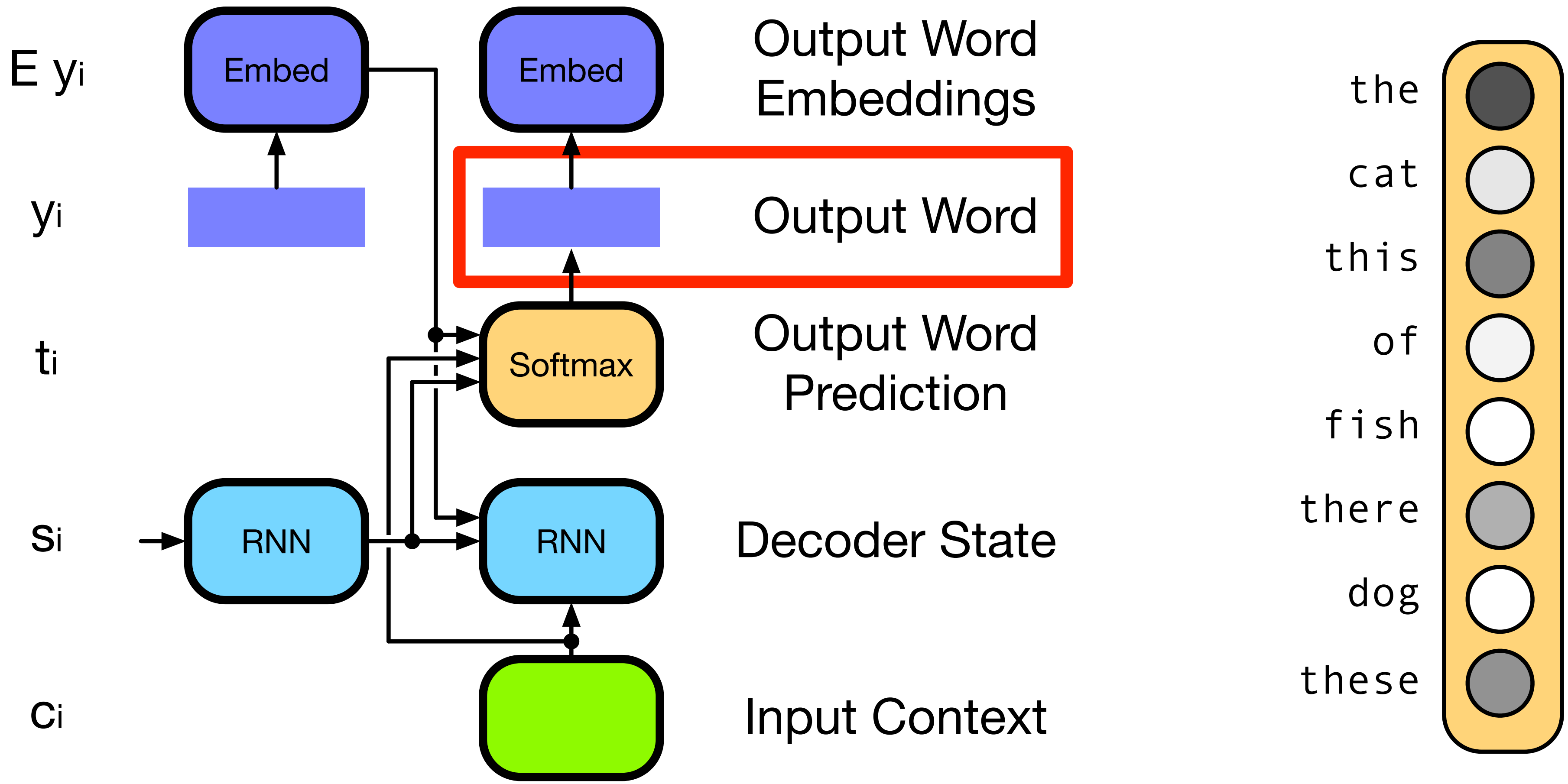
  ... we now want to translate test sentences

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

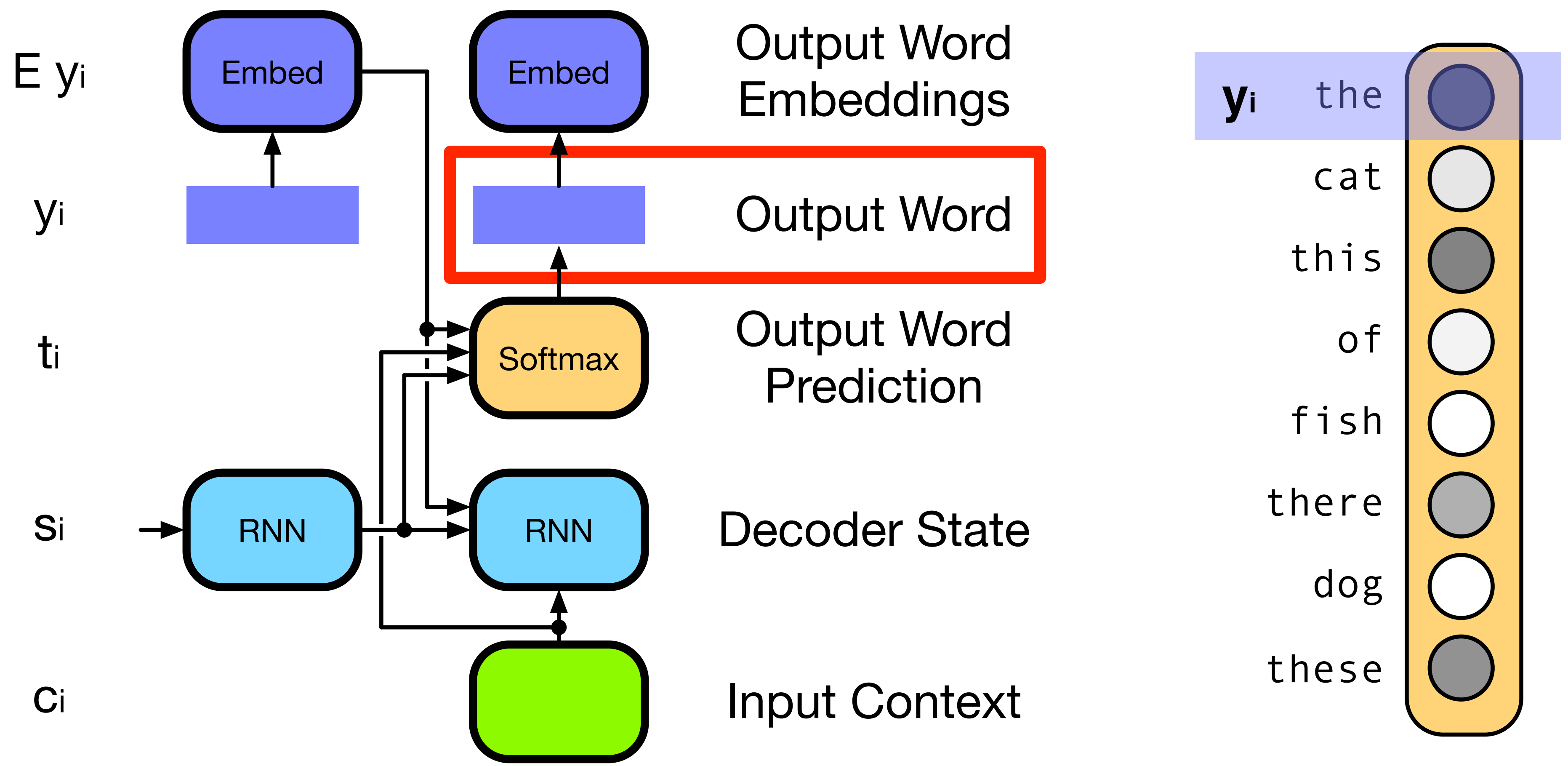
Word Prediction



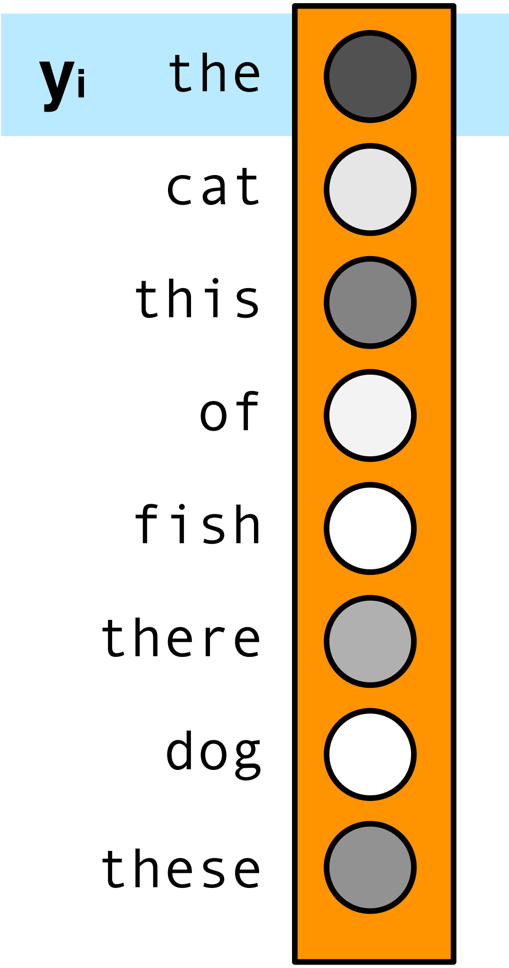
Selected Word



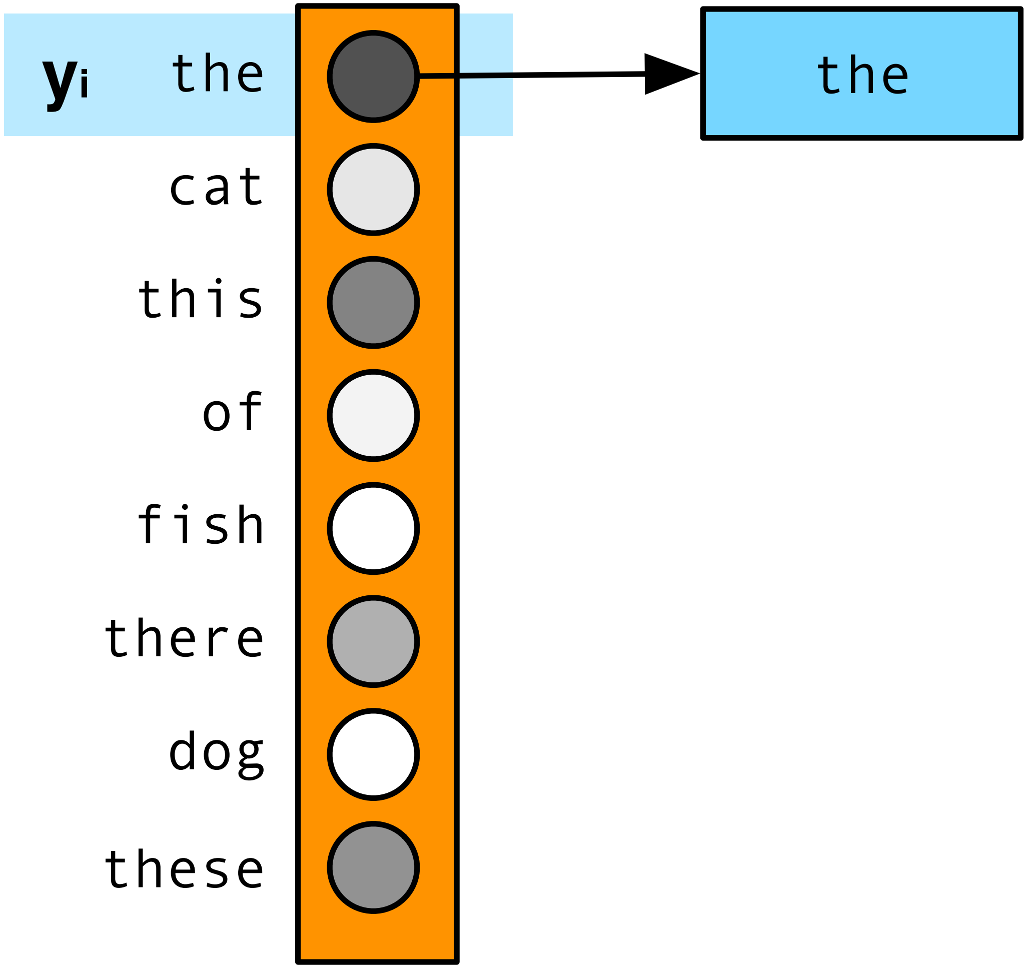
Embedding


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

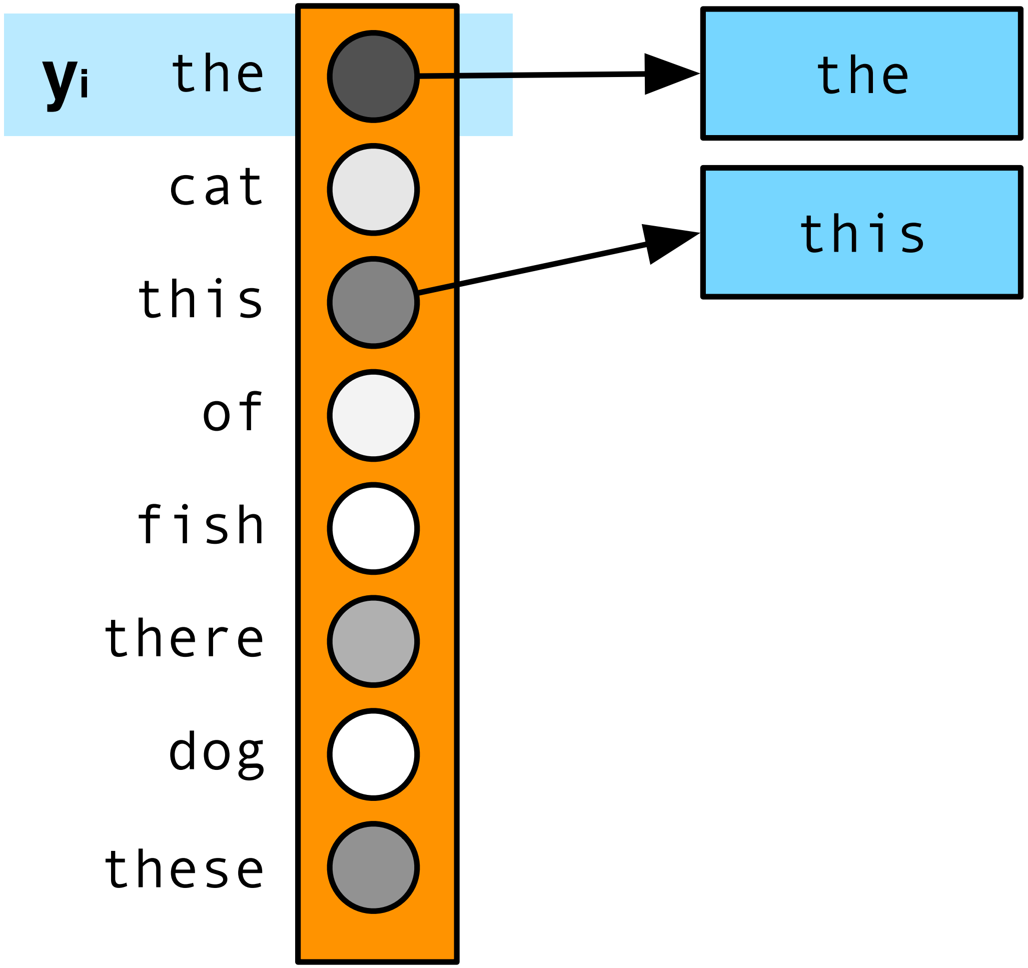
Distribution of Word Predictions



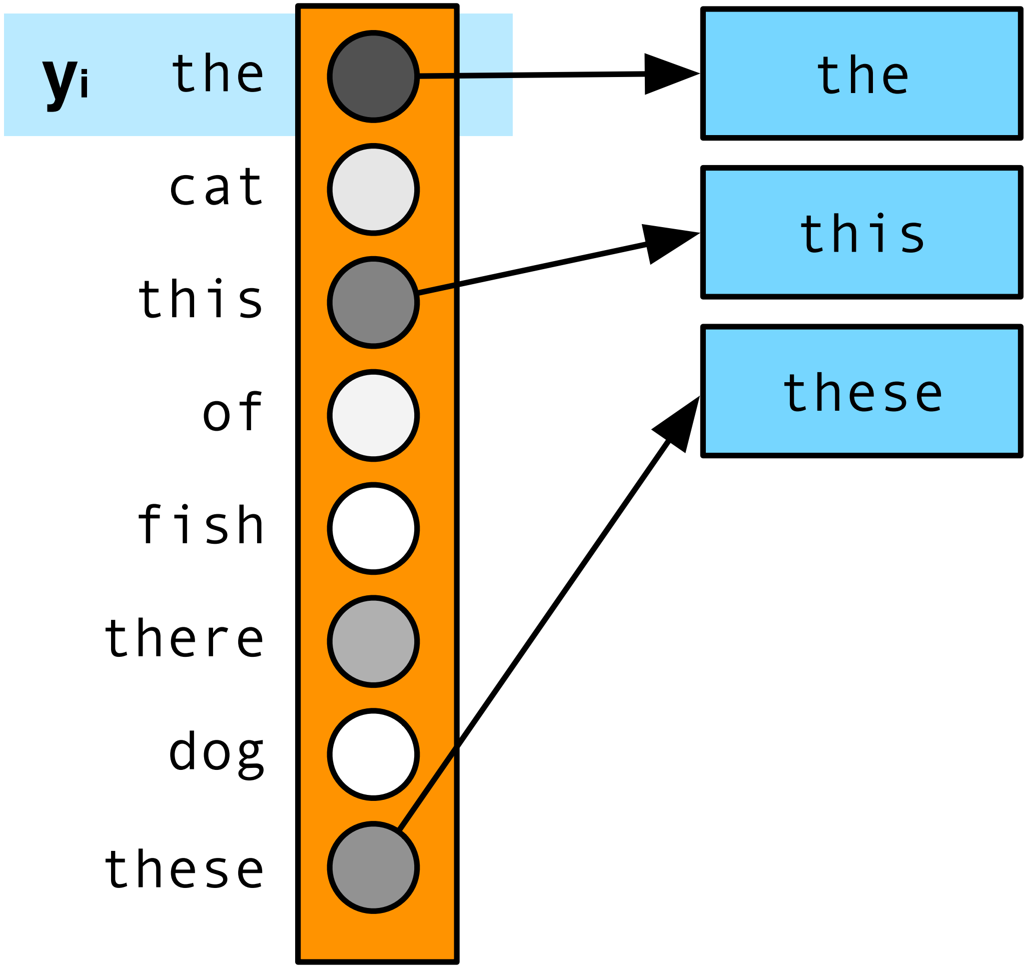
Select Best Word



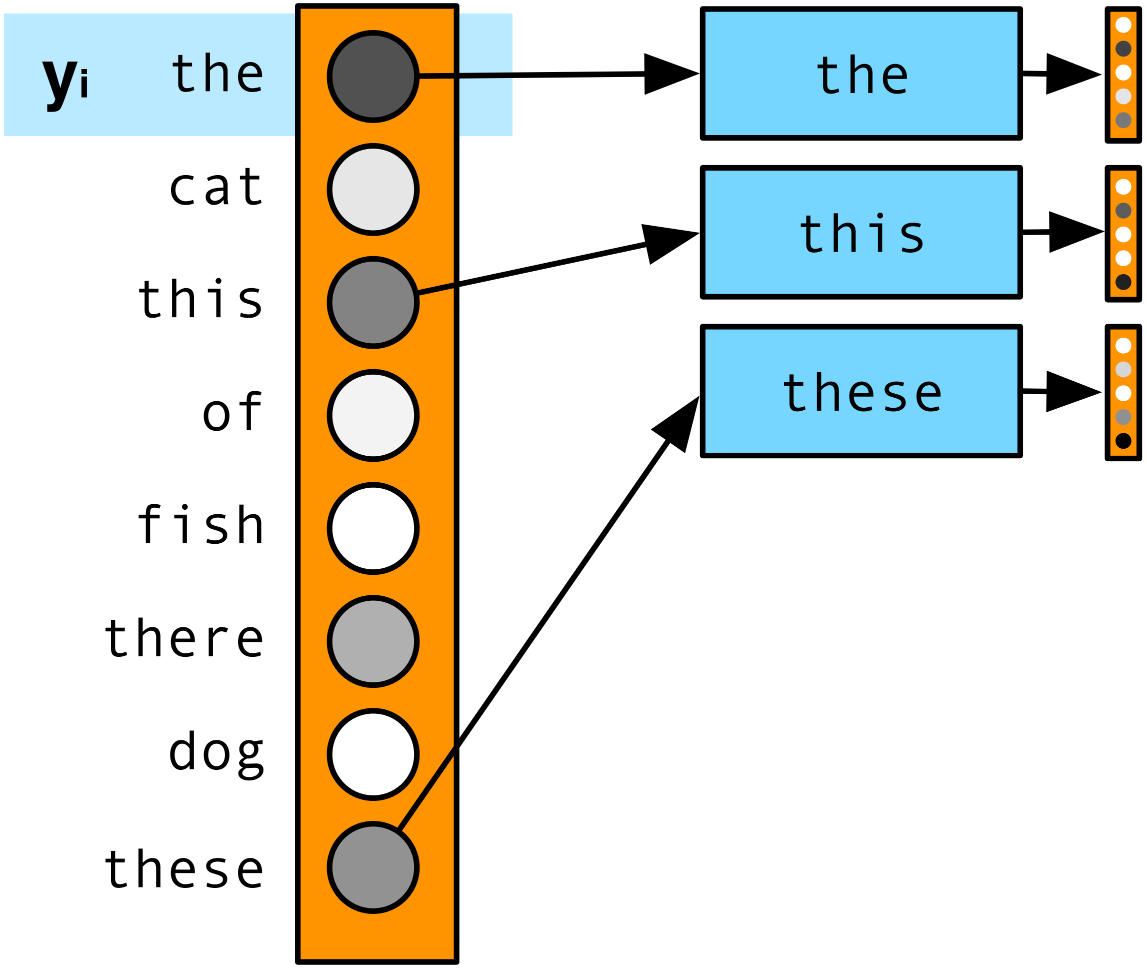
Select Second Best Word



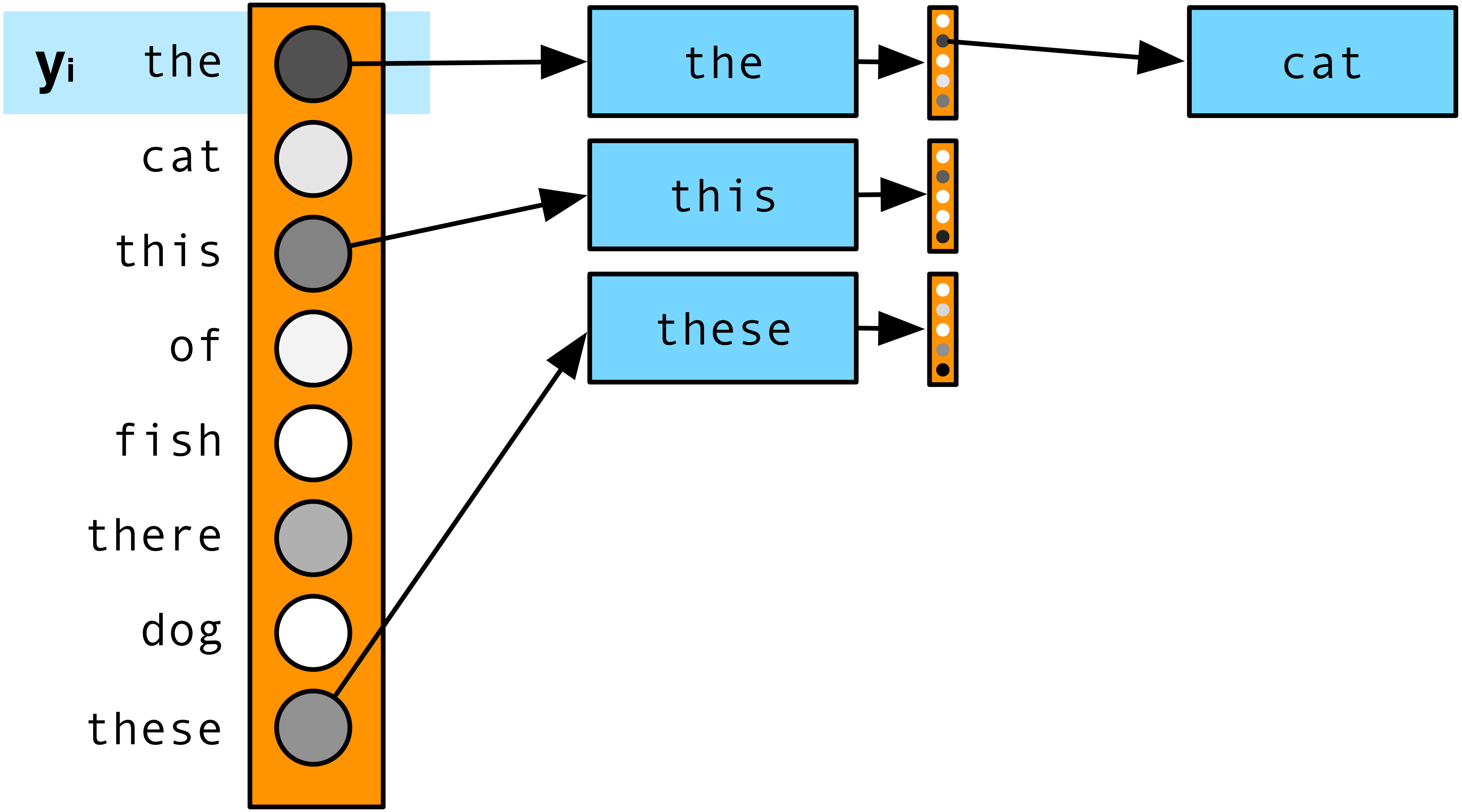
Select Third Best Word



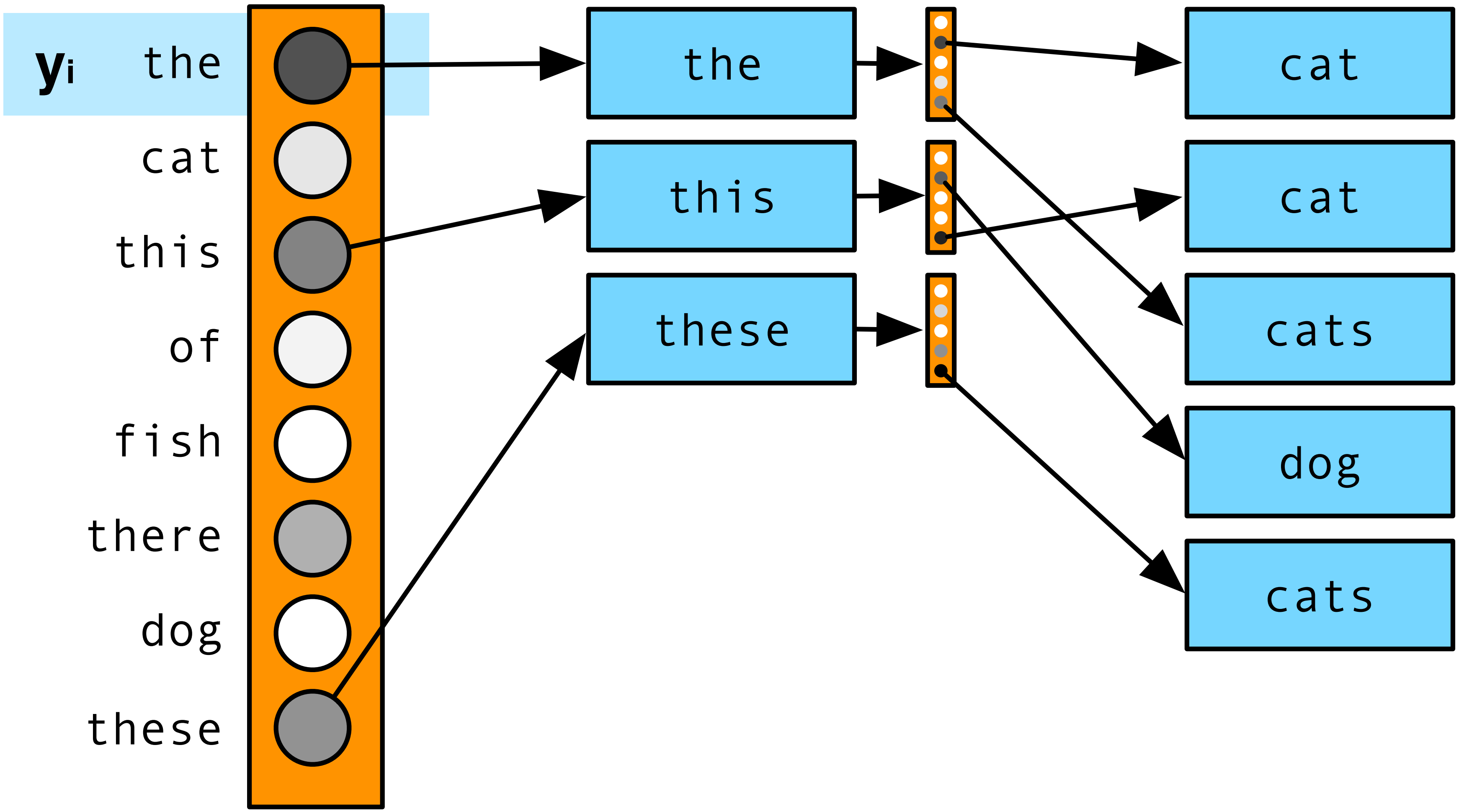
Use Selected Word for Next Predictions



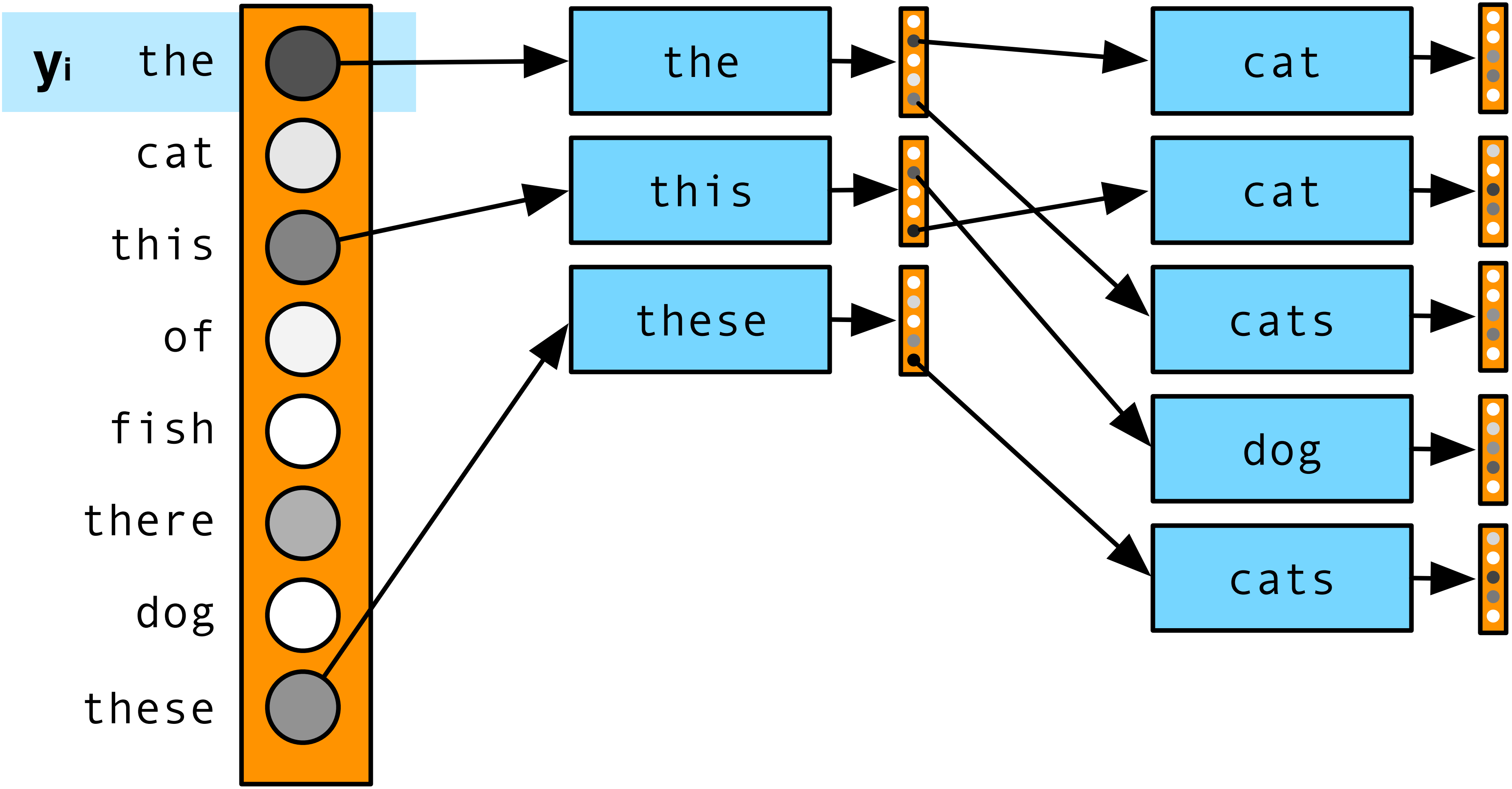
Select Best Continuation



Select Next Best Continuations

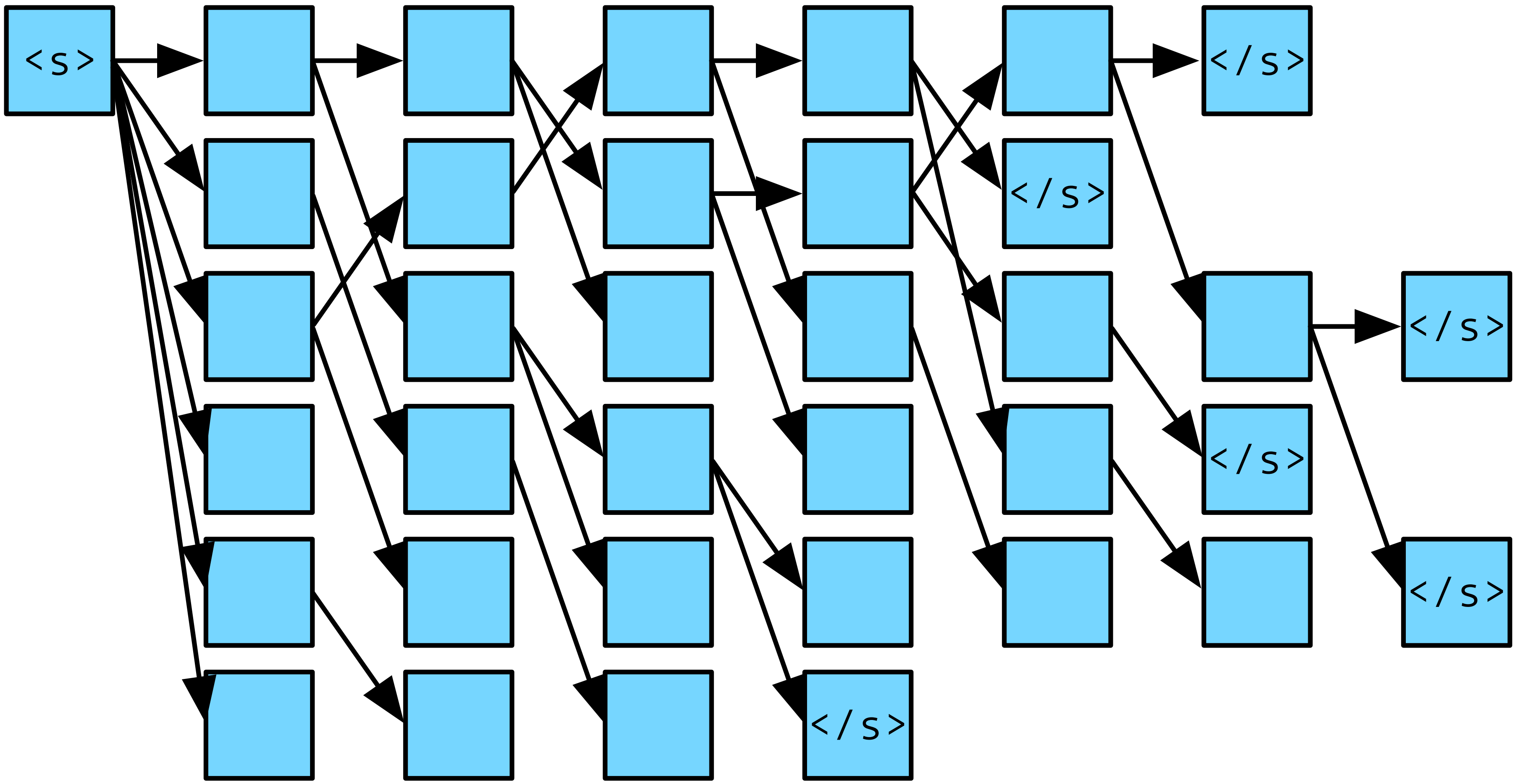


Continue...

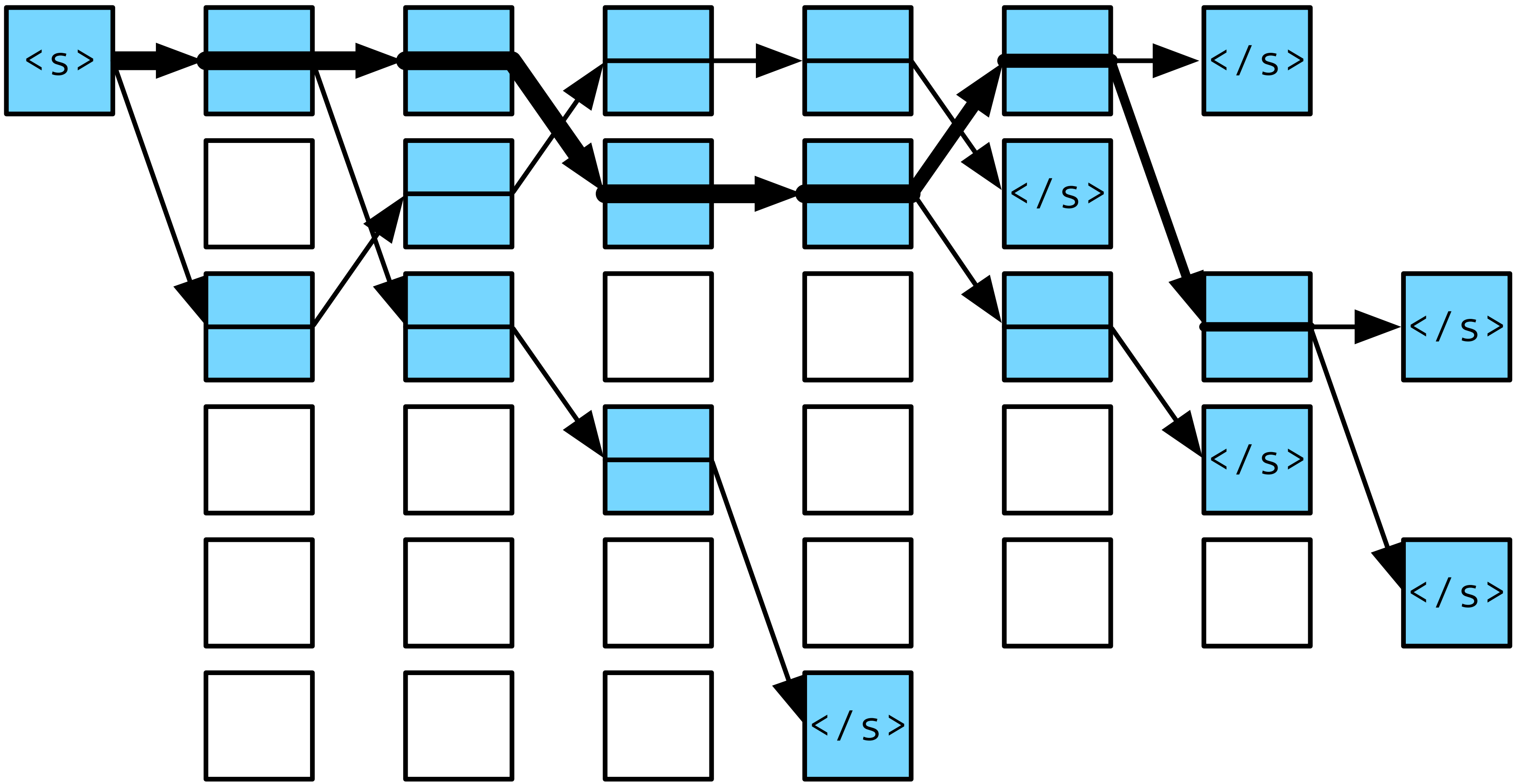


y_i
the
cat
this
of
fish
there
dog
these
the
this
these
cat
cat
cats
dog
cats

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%)	

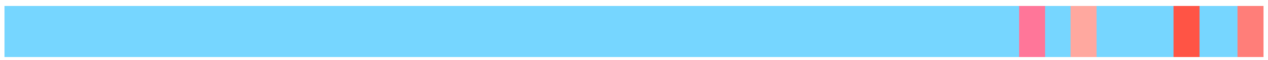
ensembling

Ensembling

- Train multiple models

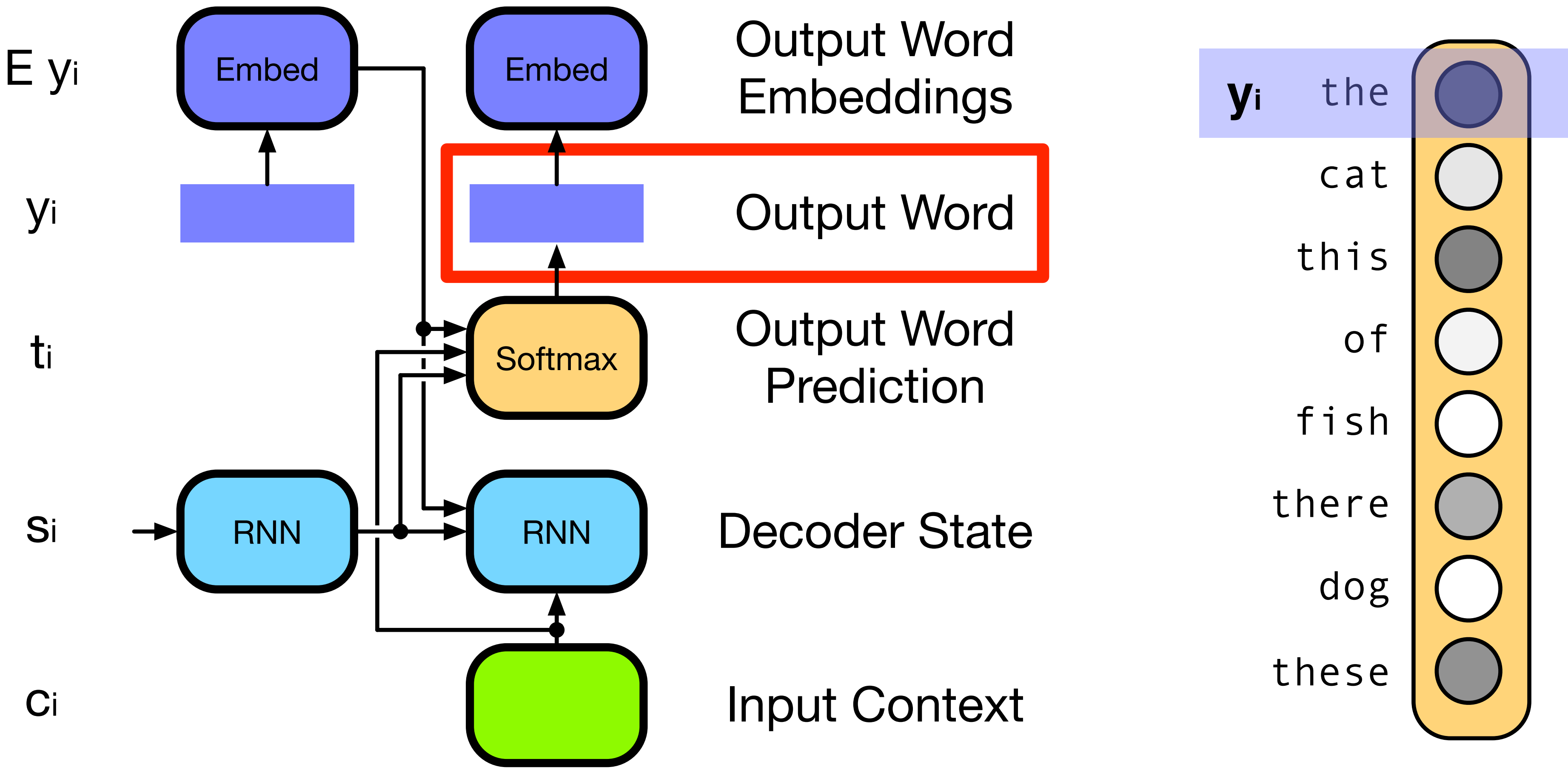
- Say, by different random initializations

- Or, by using model dumps from earlier iterations



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

Decoding with Single Model



Combine Predictions

	Model 1	Model 2	Model 3	Model 4	Model Average
the	.54	.52	.12	.29	.37
cat	.01	.02	.33	.03	.10
this	.11	.01	.06	.14	.08
of	.00	.00	.01	.08	.02
fish	.00	.12	.15	.00	.07
there	.03	.03	.00	.07	.03
dog	.00	.00	.05	.20	.06
these	.05	.09	.09	.00	.00

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)

the $\rightarrow$ cat $\rightarrow$ is $\rightarrow$ in $\rightarrow$ the $\rightarrow$ bag $\rightarrow$.

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

the $\leftarrow$ cat $\leftarrow$ is $\leftarrow$ in $\leftarrow$ the $\leftarrow$ 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

Inverse Decoding

- Recall: Bayes rule

$$p(y|x) = \frac{1}{p(x)}\, p(x|y)\, p(y)$$

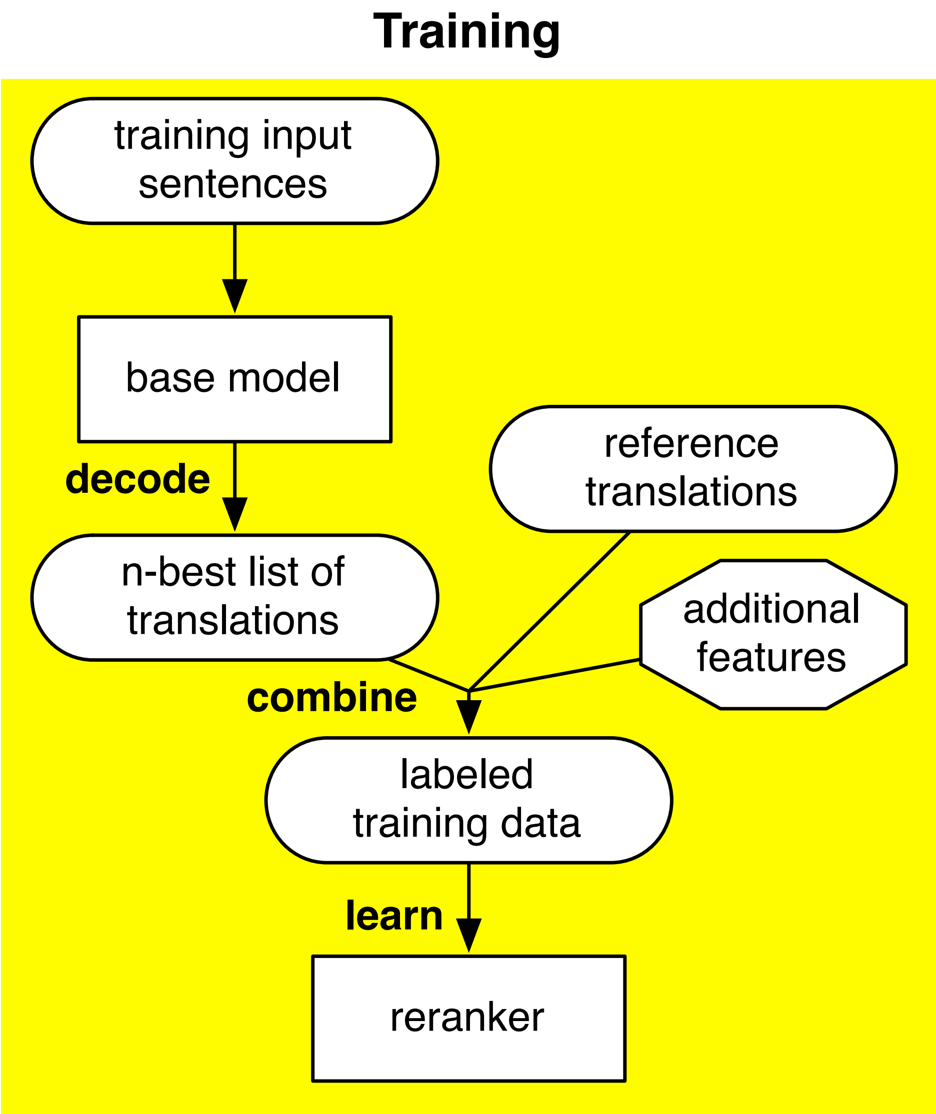
- Language model $p(y)$
  - trained on monolingual target side data
  - can already be added to ensemble decoding

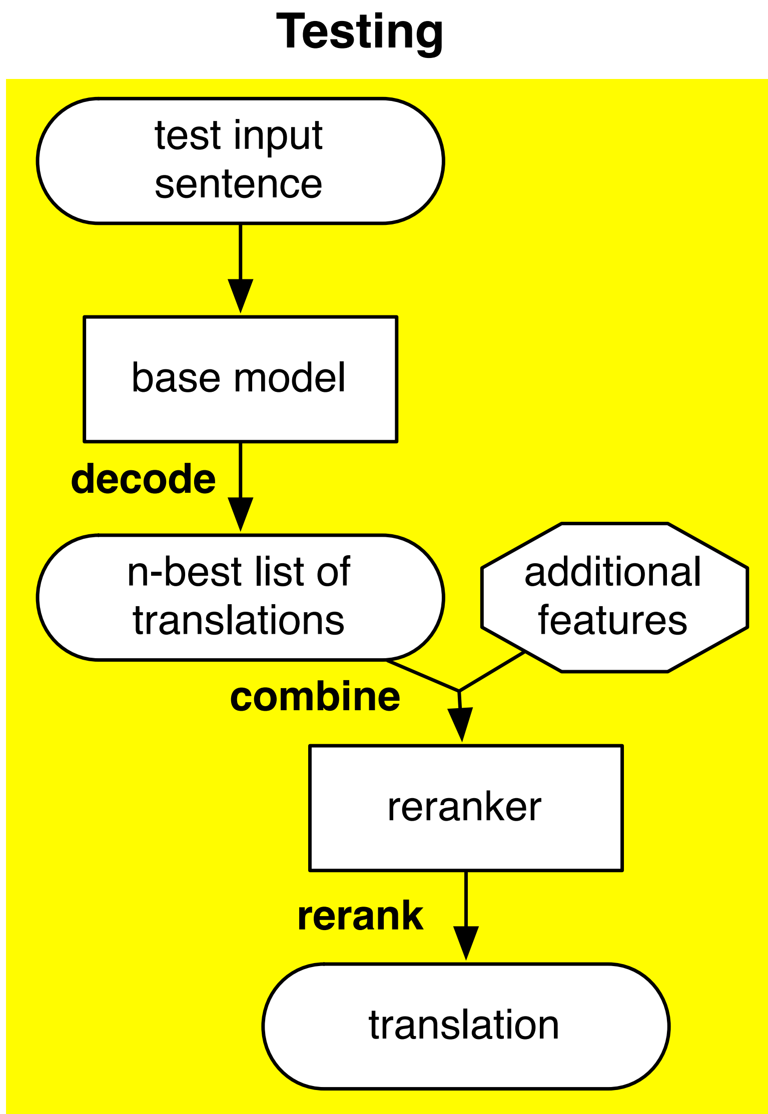
- Inverse translation model $p(x|y)$
  - train a system in the reverse language direction
  - used in reranking

Reranking

- Several models provide a score each
  - regular model
  - inverse model
  - right-to-left model
  - language model
- These scores could be just added up
- Typically better: weighting the score to optimize translation quality

Training Reranker





Learning Reranking Weights

- Minimum error rate training (MERT)
  - optimize one weight at a time, leave others constant
  - check how different values change n-best lists
  - only a some threshold values change ranking
    $\rightarrow$ can be done exhaustively

- Pairwise Ranked Optimization (PRO)
  - for each sentence in tuning set
  - for each pair of translations in n-best list
  - check which one is a better translation, leaving everything else fixed
  - create a training example
    (difference in feature values $\rightarrow$ { better, worse })
  - train linear classifier that learns weights for each feature

- This has not been explored much in neural machine translation

Lack of Diversity

Translations of the German sentence
Er wollte nie an irgendeiner Art von Auseinandersetzung teilnehmen.

He never wanted to participate in any kind of confrontation.
He never wanted to take part in any kind of confrontation.
He never wanted to participate in any kind of argument.
He never wanted to take part in any kind of argument.
He never wanted to participate in any sort of confrontation.
He never wanted to take part in any sort of confrontation.
He never wanted to participate in any sort of argument.
He never wanted to take part in any sort of argument.
He never wanted to participate in any kind of controversy.
He never wanted to take part in any kind of controversy.
He never intended to participate in any kind of confrontation.
He never intended to take part in any kind of confrontation.
He never wanted to take part in some sort of confrontation.
He never wanted to take part in any sort of controversy.

Increasing Diversity

- Monte Carlo decoding
  - no beam search, i.e., beam size 1
  - when selecting words to extend the beam ...
  - ... do not select the top choice
  - ... do select word randomly based on their probability
  - 10% chance to choose a word with 10% probability

- Diversity bias term
  - extension of regular beam search
  - add a cost for extending a hypothesis based on rank of word choice
    * most probable word: no cost
    * second most probable word: cost c
    * third most probable word: cost $2c$

$\Rightarrow$ prefer to extend many different hypotheses

constraint decoding

Specifying Decoding Constraints

- Overriding the decisions of the decoder
- Why?

  $\Rightarrow$ translations have followed strict terminology

  $\Rightarrow$ rule-based translation of dates, quantities, etc.

  $\Rightarrow$ interactive translation prediction

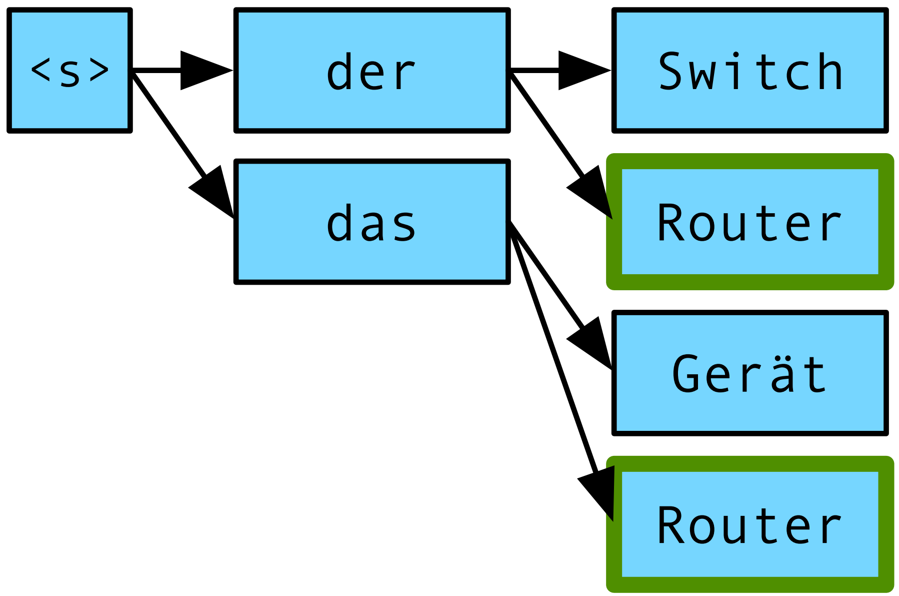
XML Schema

The <x translation="Router"> *router* </x> *is* <wall/>
a model <zone> *Psy X500 Pro* </zone> *.*

- The XML tags specify to the decoder that
  - the word *router* to be translated as *Router*
  - *The router is*, to be translated before the rest (<wall/>)
  - brand name *Psy X500 Pro* to be translated as a unit (<zone>, </zone>)

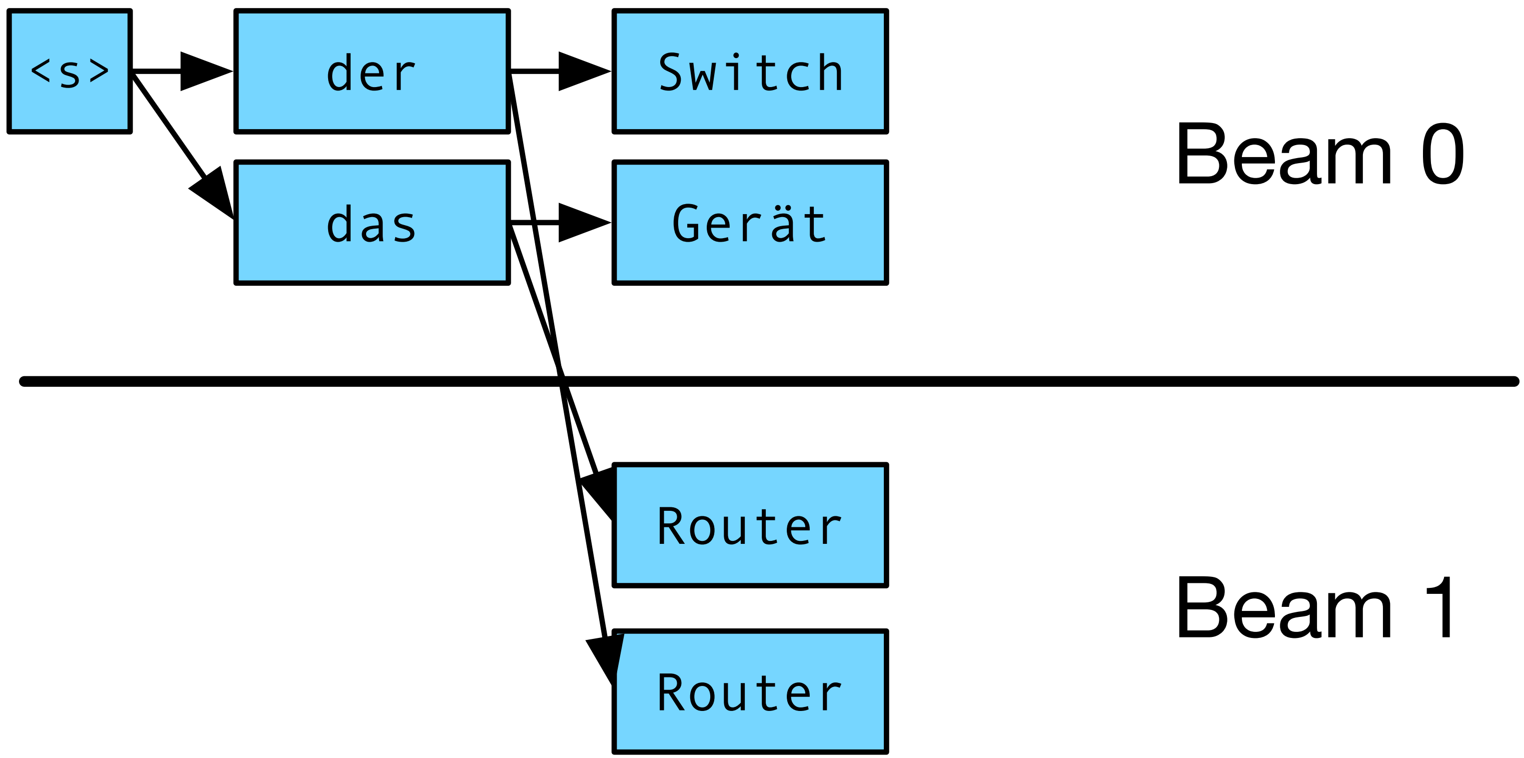
Decoding

The <x translation="Router"> *router* </x> *is a model Psy X500 Pro .*

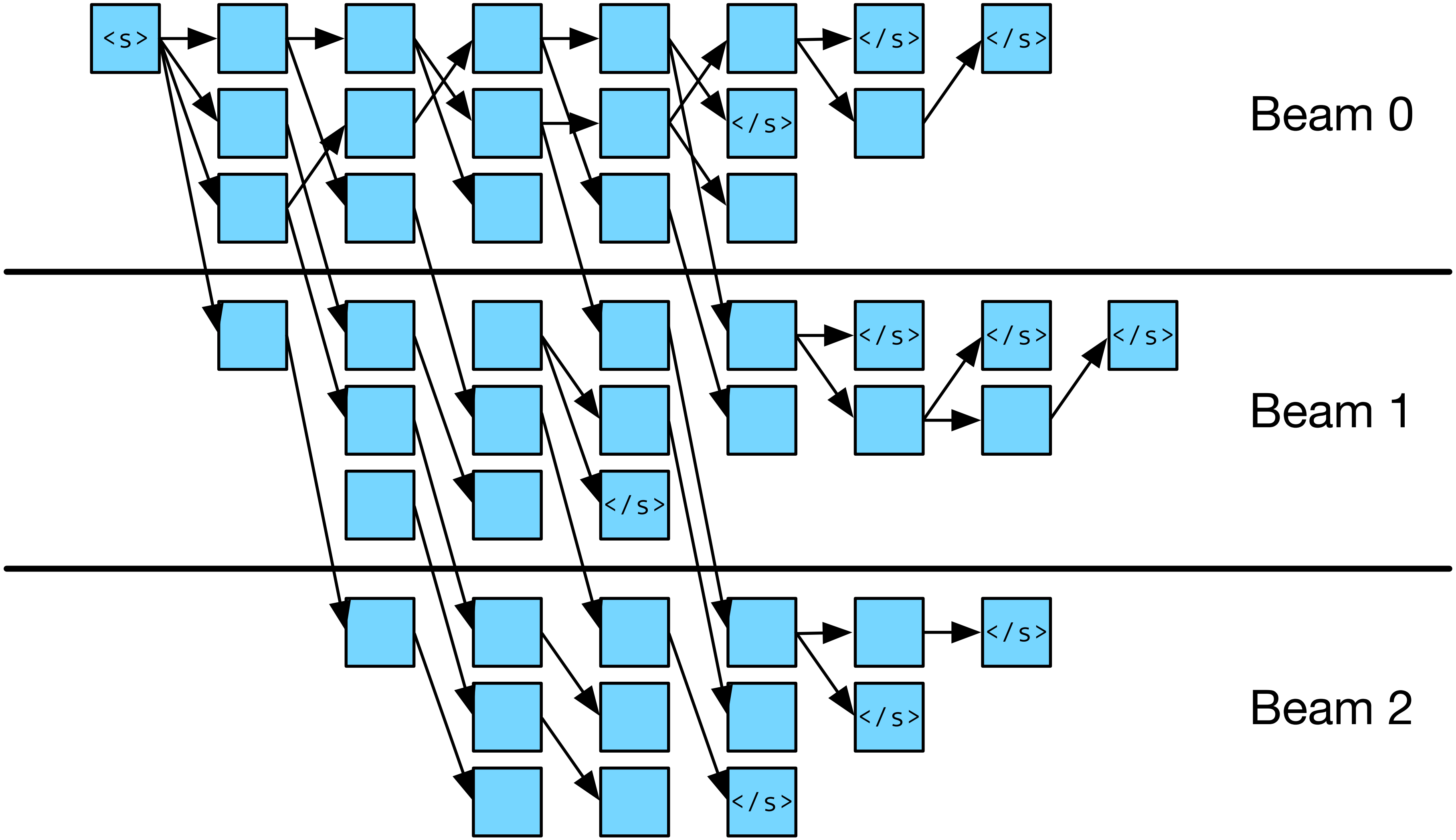


- Satisfying constraints typically costly (overriding model-best choices)
- Solution: separate beam, based on how many constraints satisfied

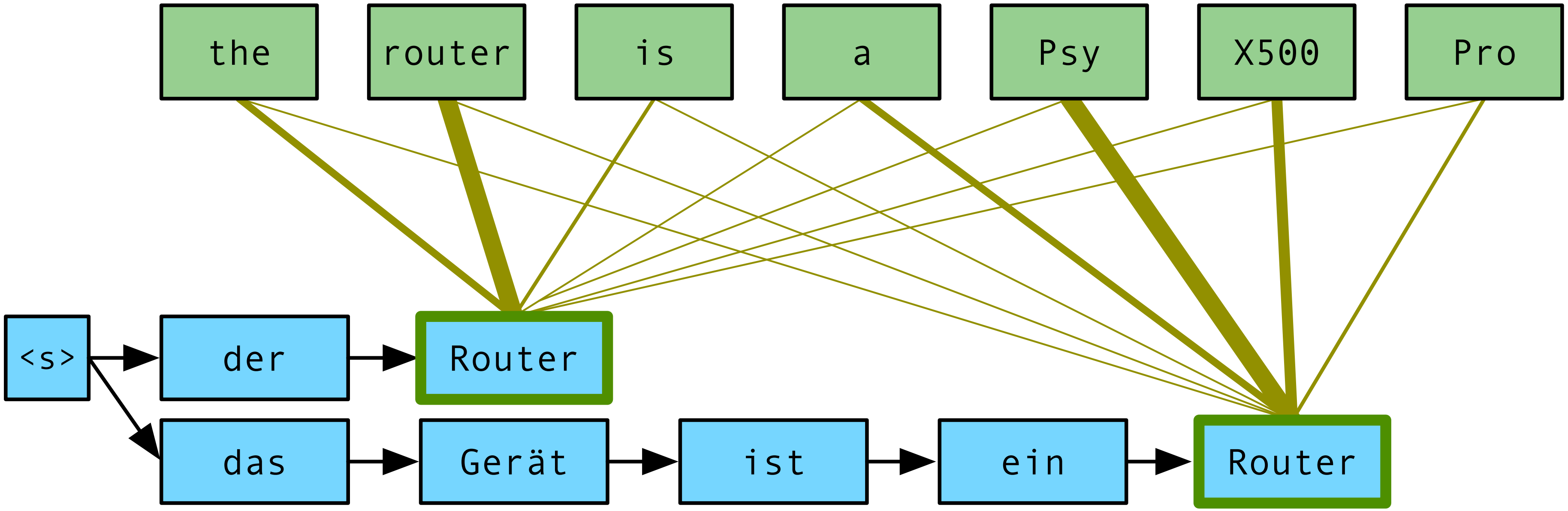
Grid Search



Grid Search



Considering Alignment



- Two hypothesis that fulfill the constraint
  - first one has relevant input words in attention focus
  - second one does not have relevant input words in attention focus

Considering Alignment

- When satisfying a constraint...
  - minimum amount of attention needs to be paid to source
  - use alignment scores as additional cost
- When not satisfying a constraint...
  - block out attention to words not covered by constraint