CS388: Natural Language Processing

Lecture 15: Seq2seq/ Attention III

Greg Durrett

TEXAS

The University of Texas at Austin

Administrivia

- Mini 2 back
- Final project feedback posted
- Project 2 due in 9 days

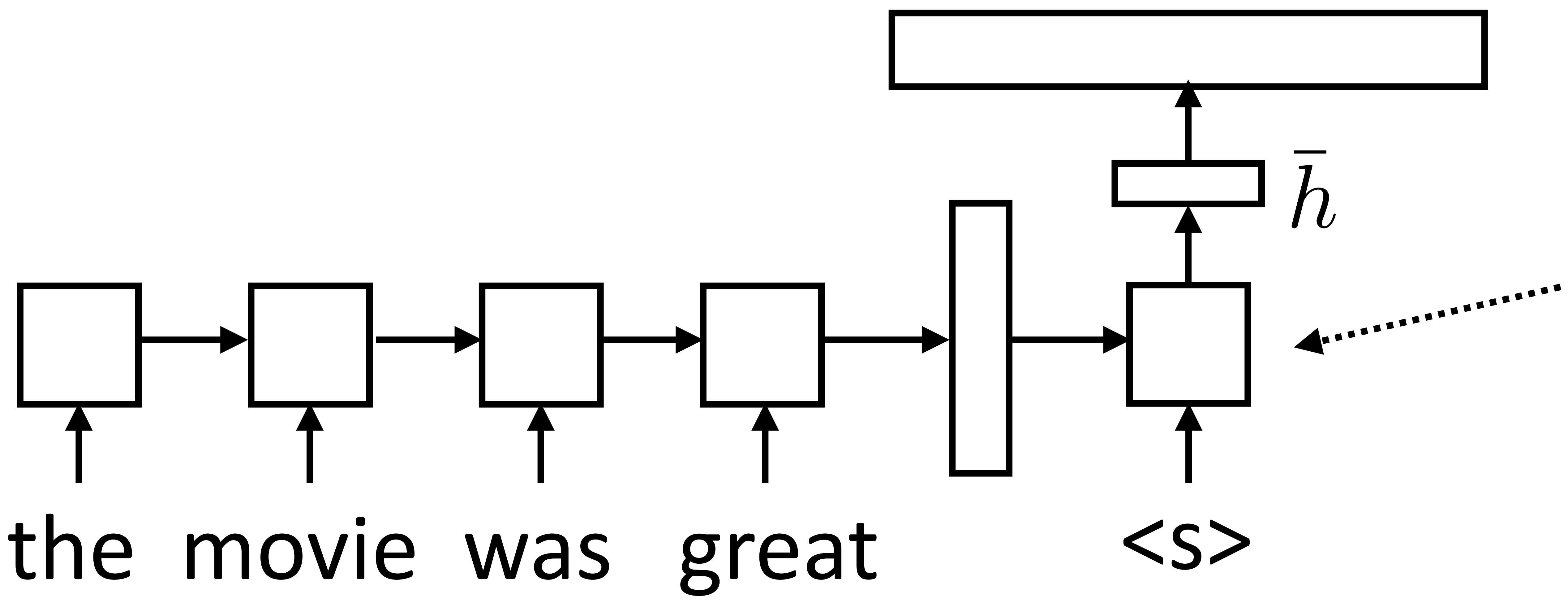
Recall: Seq2seq Model

- Generate next word conditioned on previous word as well as hidden state

- W size is |vocab| x |hidden state|, softmax over entire vocabulary

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W\bar{h})$$

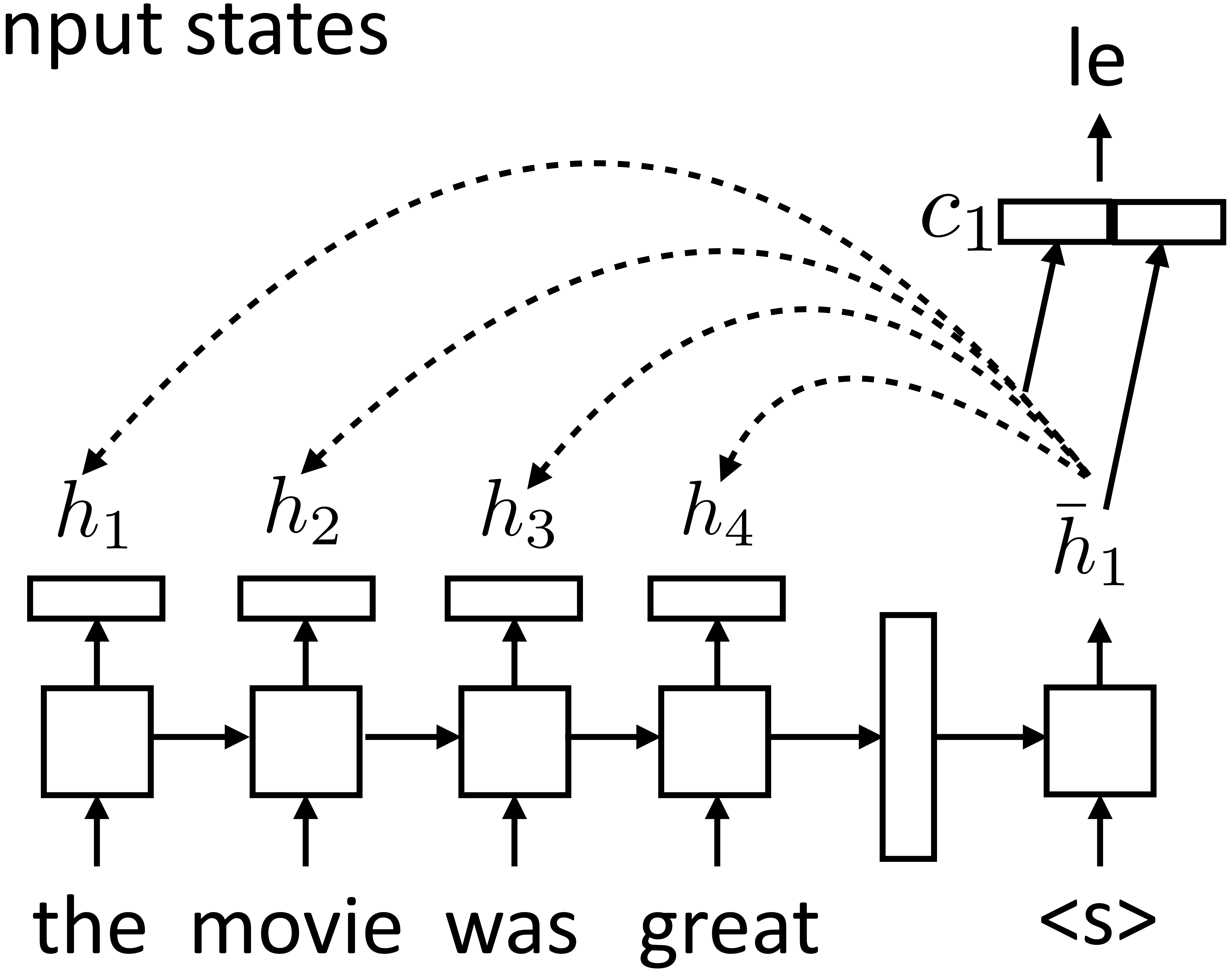
$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1})$$



Decoder has separate parameters from encoder, so this can learn to be a language model (produce a plausible next word given current one)

Recall: Attention

- For each decoder state, compute weighted sum of input states



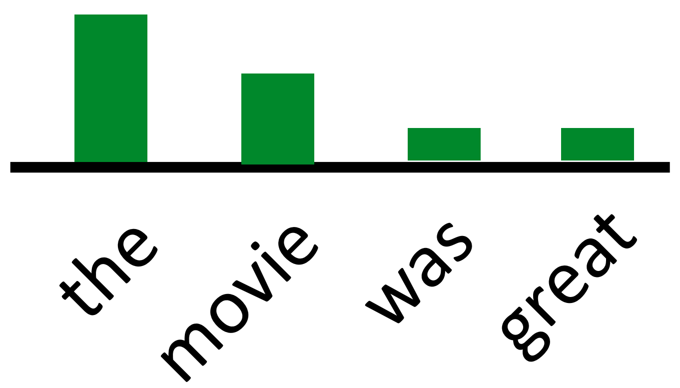
- No attn: $P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W\bar{h}_i)$

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W[c_i; \bar{h}_i])$$

$$c_i = \sum_j \alpha_{ij} h_j$$

- Weighted sum of input hidden states (vector)

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$



$$e_{ij} = f(\bar{h}_i, h_j)$$

- Some function f (TBD)

Recall: Semantic Parsing as Translation

"what states border Texas"

↓

```
lambda x ( state ( x ) and border ( x , e89 ) ) )
```

- Write down a linearized form of the semantic parse, train seq2seq models to directly translate into this representation
- No need to have an explicit grammar, simplifies algorithms
- Might not produce well-formed logical forms, might require lots of data

Jia and Liang (2016)

This Lecture

- Copy mechanisms for copying words to the output

- Decoding in seq2seq models

- Transformer architecture

Copying Input, Pointer Mechanisms

Unknown Words

- Want to be able to copy named entities like Pont-de-Buis

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W[c_i; \bar{h}_i])$$

from attention

from RNN
hidden state

- Problems: target word has to be in the vocabulary, attention + RNN need to generate good embedding to pick it

Jean et al. (2015), Luong et al. (2015)

Learning to Copy

- Suppose we only care about being able to copy words from the input (maybe we're summarizing a document)

the movie was, despite its many flaws, great $\longrightarrow$ *the movie was great*

- Standard models predict from a vocabulary, but here the vocabulary changes with every new input

On Thursday, police arrested two suspects $\longrightarrow$ *police arrested two*

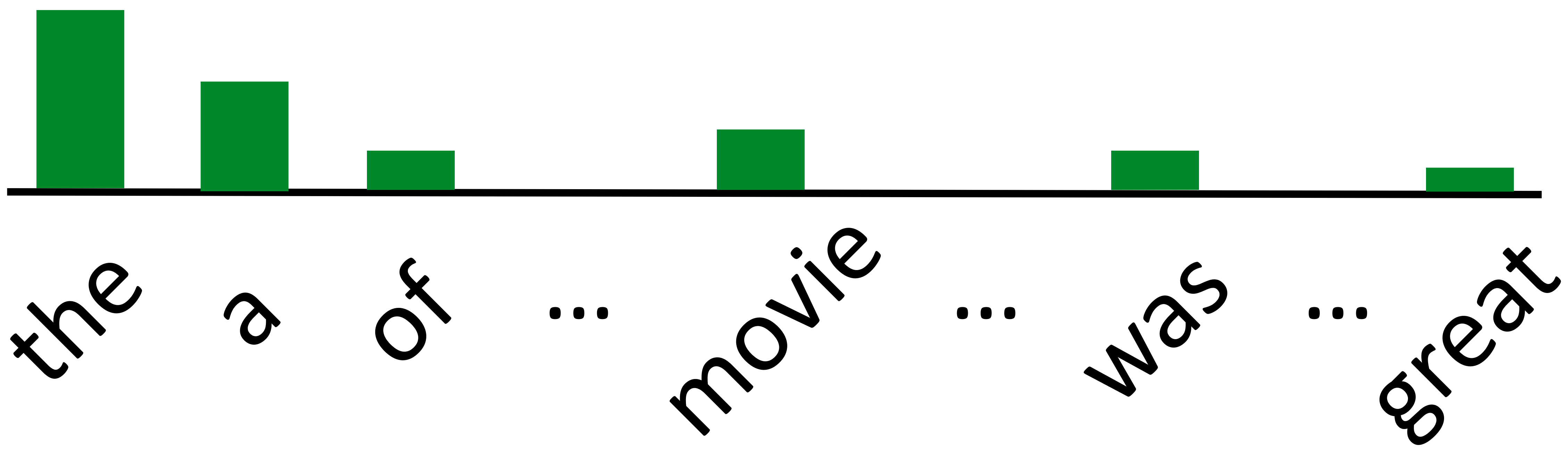
- Predicting from a fixed vocabulary doesn't make sense here

Output Space

- Let $[x_1, \ldots, x_n]$ be the set of words in the input
- Rather than distribution over the vocabulary, predict distribution over the x_i
- **Key observation:** this is exactly the same thing that attention gives us!
- Instead of a traditional softmax layer, **we use attention to predict the output directly**.
- This is called a pointer network (or a copy mechanism)

Pointer Networks

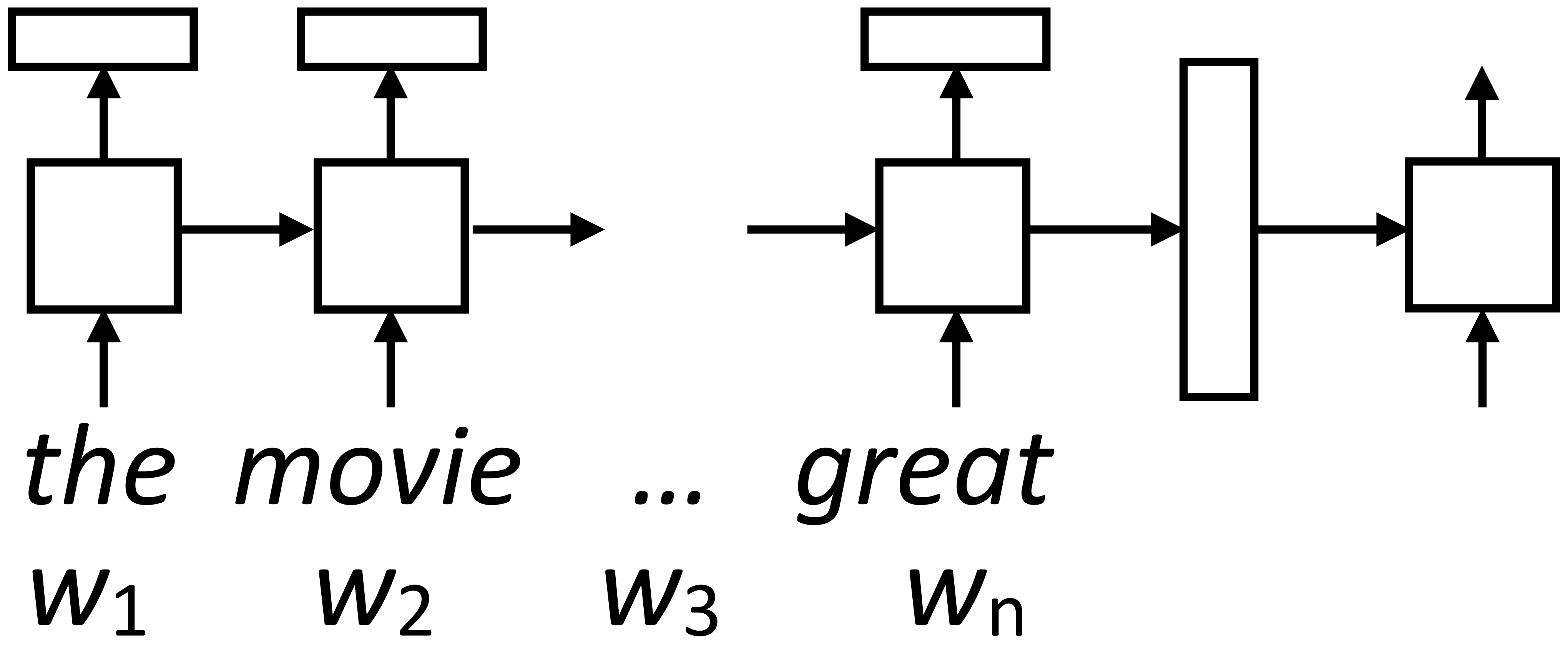
$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W[c_i; \bar{h}_i])$$

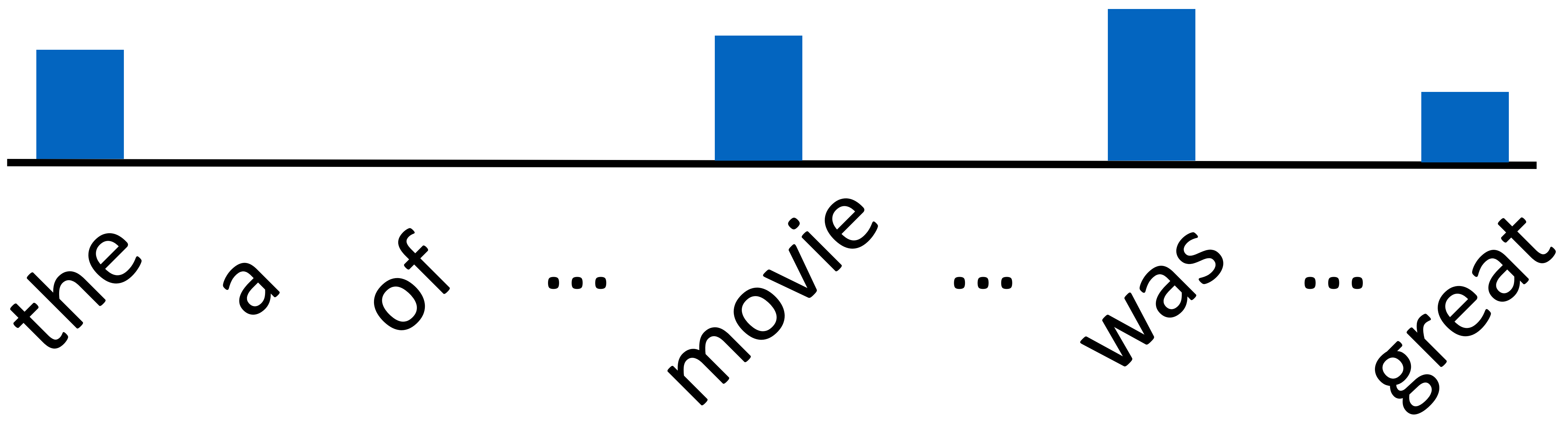


- Standard decoder (P_{vocab}): softmax over vocabulary, all words get >0 prob

- Pointer network: predict from *source words* instead of *target vocab*

$$P_{\text{pointer}}(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) \propto \begin{cases} \exp(h_j^\top V \bar{h}_i) & \text{if } y_i = w_j \\ \text{0 otherwise} & \end{cases}$$



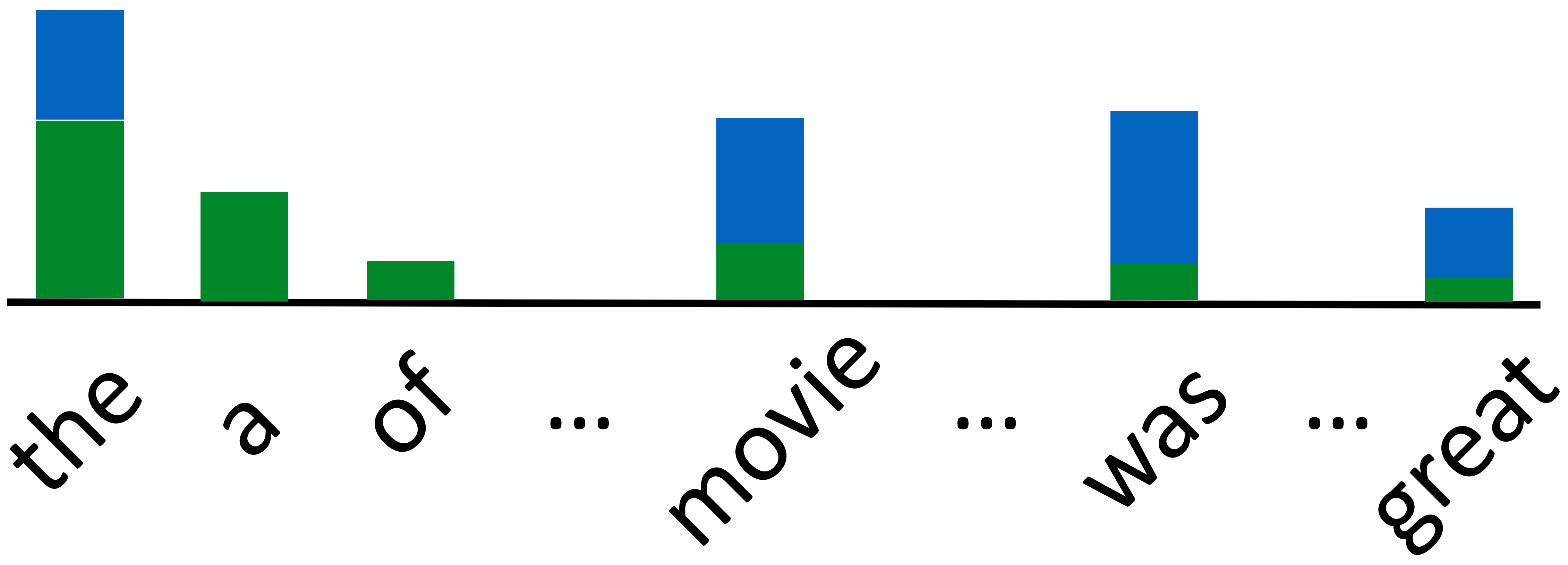


Pointer Generator Mixture Models

- Define the decoder model as a mixture model of the P_{vocab} and P_{pointer} models (previous slide)

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = P(\text{copy})P_{\text{pointer}} + (1 - P(\text{copy}))P_{\text{vocab}}$$

- Predict *P*(copy) based on decoder state, input, etc.
- Marginalize over copy variable during training and inference
- Model will be able to both generate and copy, flexibly adapt between the two



Copying

en: The *ecotax* portico in *Pont-de-Buis* , . . . [truncated] . .

fr: Le *portique* *écotaxe* de *Pont-de-Buis* , . . . [truncated]

nn: Le unk de unk à unk , . . . [truncated] . . . , a été pris

$$\left\{ \begin{array}{l} \text{le} \\ \text{de} \\ \ldots \\ \text{pris} \\ \hline \text{Pont-de-Buis} \\ \text{ecotax} \end{array} \right\}$$

(copied over and not transliterated)

- Some words we may want to copy may not be in the fixed output vocab (*Pont-de-Buis*)

- Solution: expand the vocabulary dynamically. New words can only be predicted by copying (always 0 probability under P_{vocab})

Results

	GEO	ATIS
No Copying	74.6	69.9
With Copying	85.0	76.3

- For semantic parsing, copying tokens from the input (*texas*) can be very useful
- Copying typically helps a bit, but attention captures most of the benefit. However, vocabulary expansion is critical for some tasks (machine translation)

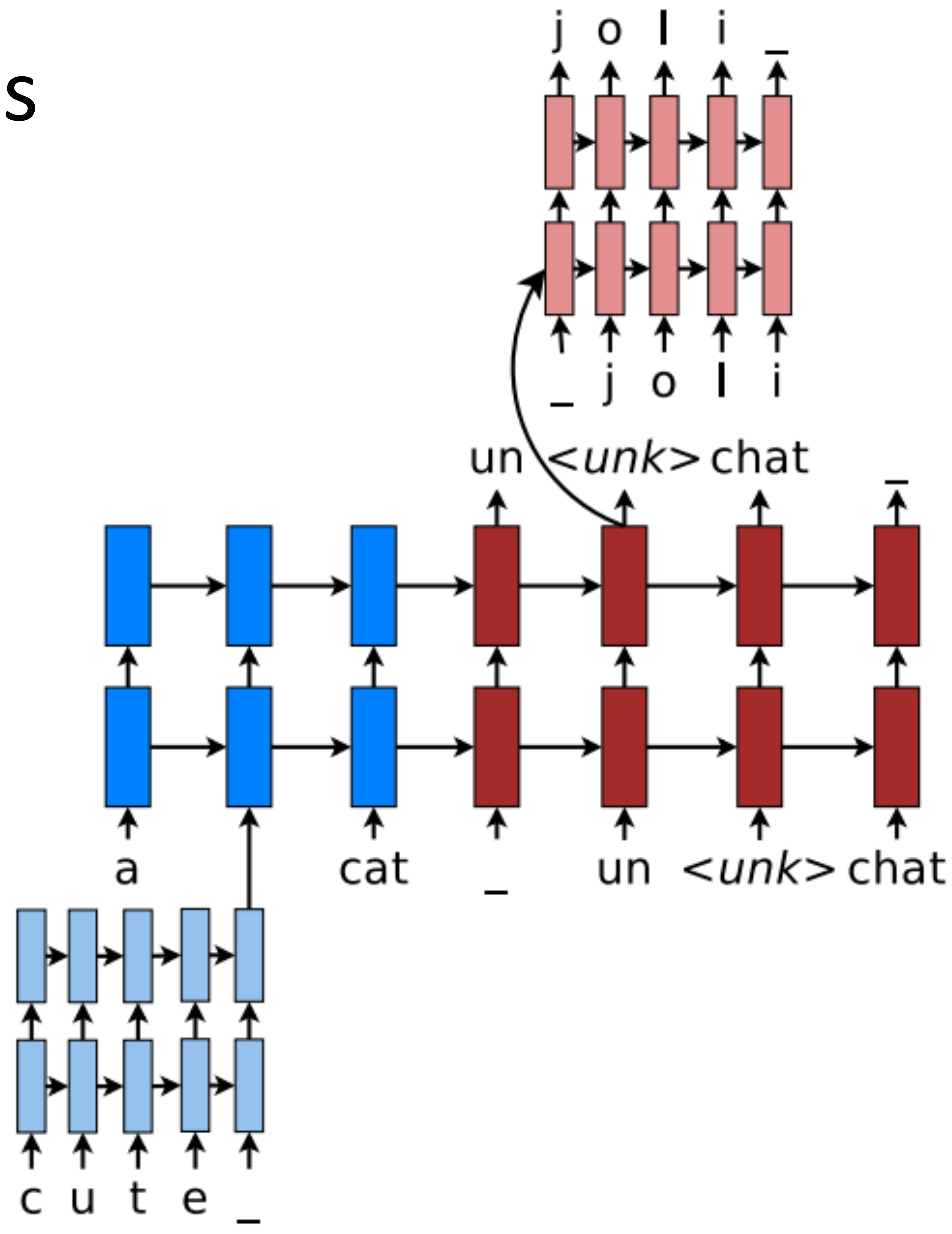
Jia and Liang (2016)

Rare Words: Character Models

- If we predict an unk token, generate the results from a character LSTM

- Can potentially transliterate new concepts, but architecture is more complicated and slower to train

- We will talk about alternatives to this when we talk about machine translation

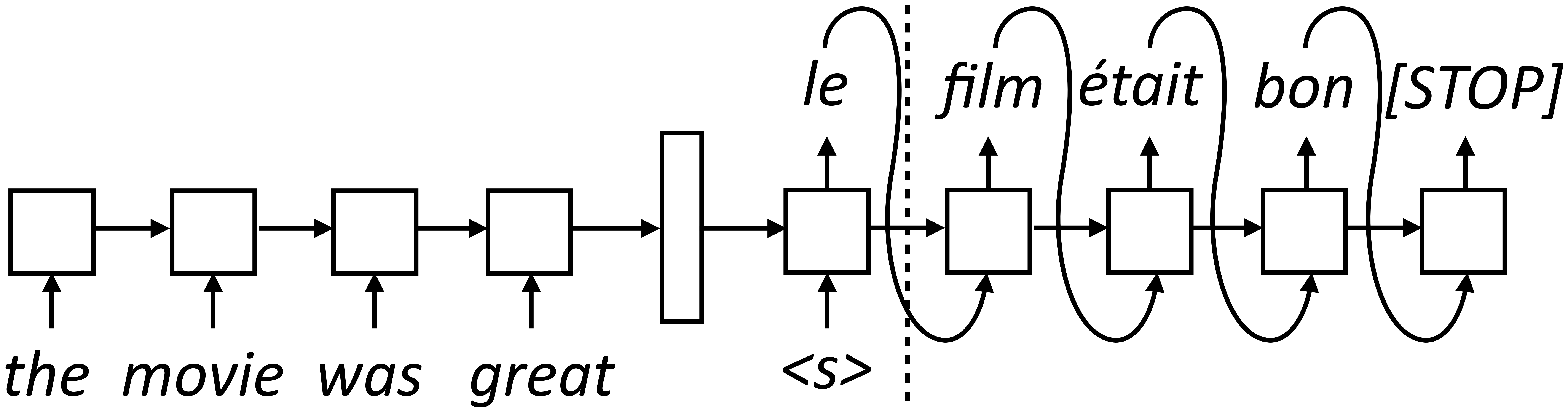


Luong et al. (2016)

Decoding Strategies

Greedy Decoding

- Generate next word conditioned on previous word as well as hidden state



- During inference: need to compute the argmax over the word predictions and then feed that to the next RNN state. This is **greedy decoding**

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W\bar{h})$$

(or attention/copying/etc.)

$$y_{\mathrm{pred}} = \mathrm{argmax}_y P(y|\mathbf{x}, y_1, \ldots, y_{i-1})$$

Problems with Greedy Decoding

- Only returns one solution, and it may not be optimal
- Can address this with **beam search**, which usually works better…but even beam search may not find the correct answer! (max probability sequence)

Model	Beam-10	
	BLEU	#Search err.
LSTM*	28.6	58.4%
SliceNet*	28.8	46.0%
Transformer-Base	30.3	57.7%
Transformer-Big*	31.7	32.1%

Stahlberg and Byrne (2019)

“Problems” with Beam Decoding

- For machine translation, the highest probability sequence is often the empty string! (>50% of the time)

Search	BLEU	Ratio	#Search errors	#Empty
Greedy	29.3	1.02	73.6%	0.0%
Beam-10	30.3	1.00	57.7%	0.0%
Exact	2.1	0.06	0.0%	51.8%

- Beam search results in *fortuitous search errors* that avoid these bad solutions

Stahlberg and Byrne (2019)

Sampling

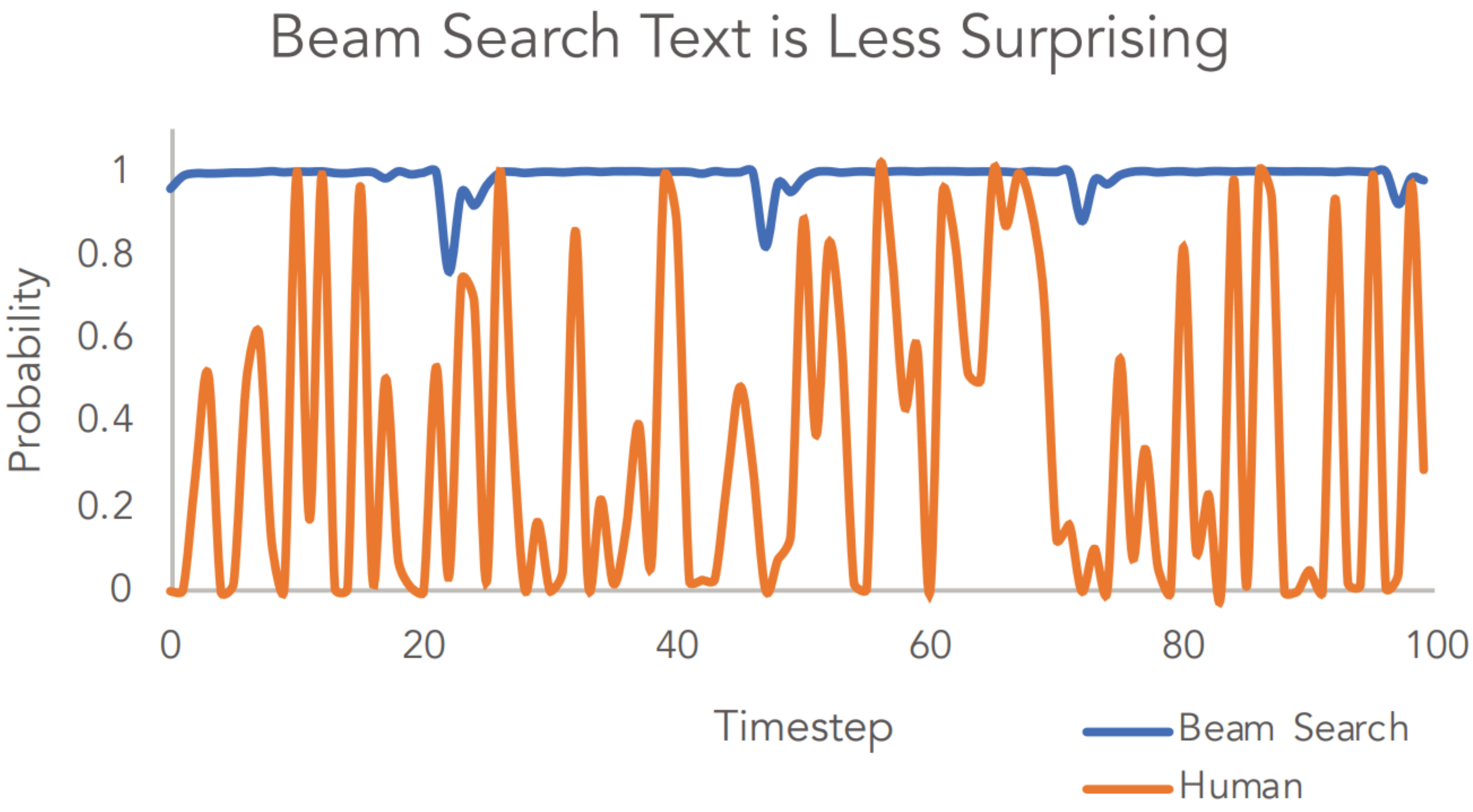
- Beam search may give many similar sequences, and these actually may be *too close* to the optimal. Can sample instead:

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W\bar{h})$$

$$y_{\mathrm{sampled}} \sim P(y|\mathbf{x}, y_1, \ldots, y_{i-1})$$

- Text *degeneration*: greedy solution can be uninteresting / vacuous for various reasons. Sampling can help.

Beam Search vs. Sampling



Holtzman et al. (2019)

Beam Search vs. Sampling

- These are samples from an unconditioned language model (not seq2seq model)

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

- Sampling is better but sometimes draws too far from the tail of the distribution

Holtzman et al. (2019)

Decoding Strategies

- Greedy
- Beam search
- Sampling
- Nucleus or top-k sampling:
  - Nucleus: take the top p% (95%) of the distribution, sample from within that
  - Top-k: take the top k most likely words (k=5), sample from those

Generation Tasks

WebText

An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

Beam Search, *b*=16

The number of stranded whales has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year. The number of whales stranded on the West Australian coast has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year.

Pure Sampling

The Australian Food Safety Authority has warned Australia's beaches may be revitalised this year because healthy seabirds and seals have been on the move. More than 50,000 seabirds, sea mammals and seahorses have been swept into the sea by the Holden CS118 and Adelaide Airport CS300 from 2013. A major white-bat and umidauda migration across Australia is under way in Australia for the first time, with numbers reaching an estimated 50,000.

Nucleus, *p*=0.95

There has been an unprecedented number of calves caught in the nets of whaling stations that operate in WA. Pilot whales continue to migrate to feeding grounds to feed their calves. They are now vulnerable due to the decline of wild populations; they are restricted to one breeding site each year. Image copyright Yoon Bo Kim But, with sharp decline in wild populations the size of the Petrels are shrinking and dwindling population means there will only be room for a few new fowl.

Generation Tasks

- There are a range of seq2seq modeling tasks we will address
- For more constrained problems: greedy/beam decoding are usually best
- For less constrained problems: nucleus sampling introduces favorable variation in the output

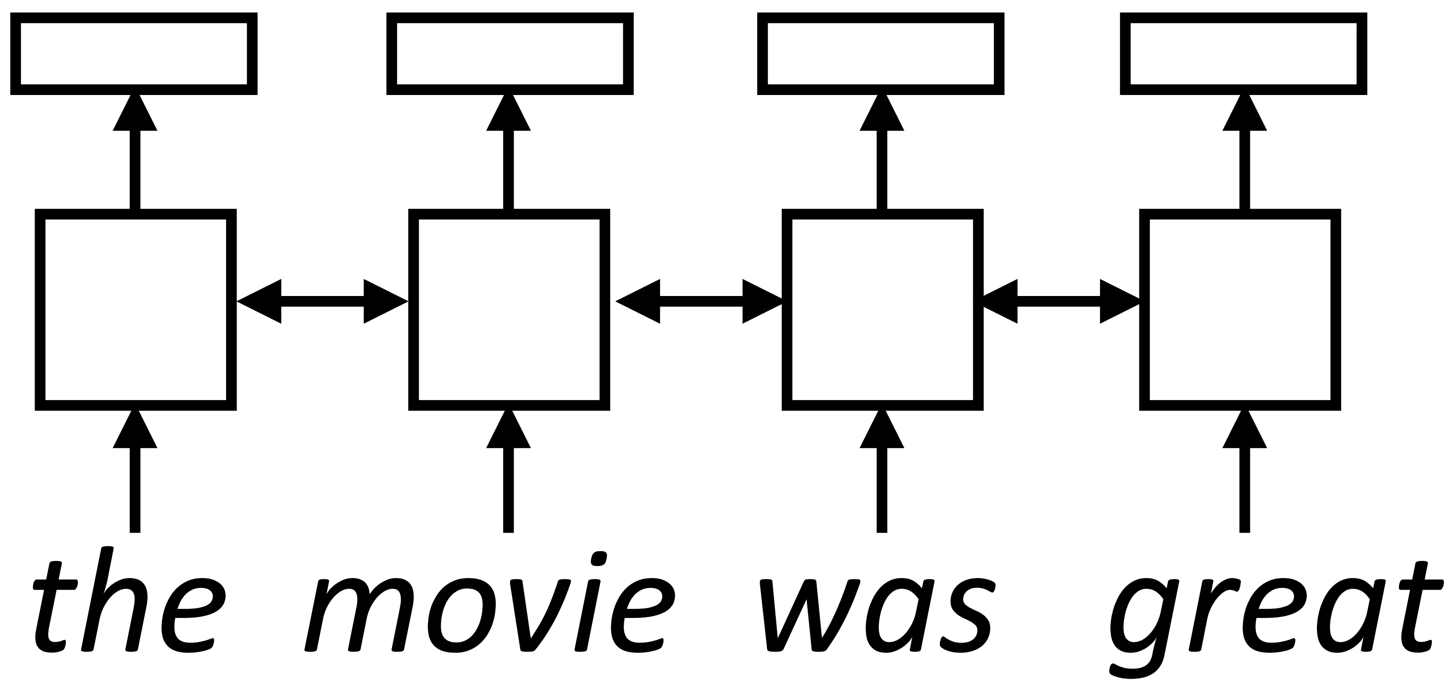
Less constrained ⟷ More constrained

Unconditioned sampling/ "story generation" — Dialogue — Summarization — Data-to-text — Translation — Text-to-code

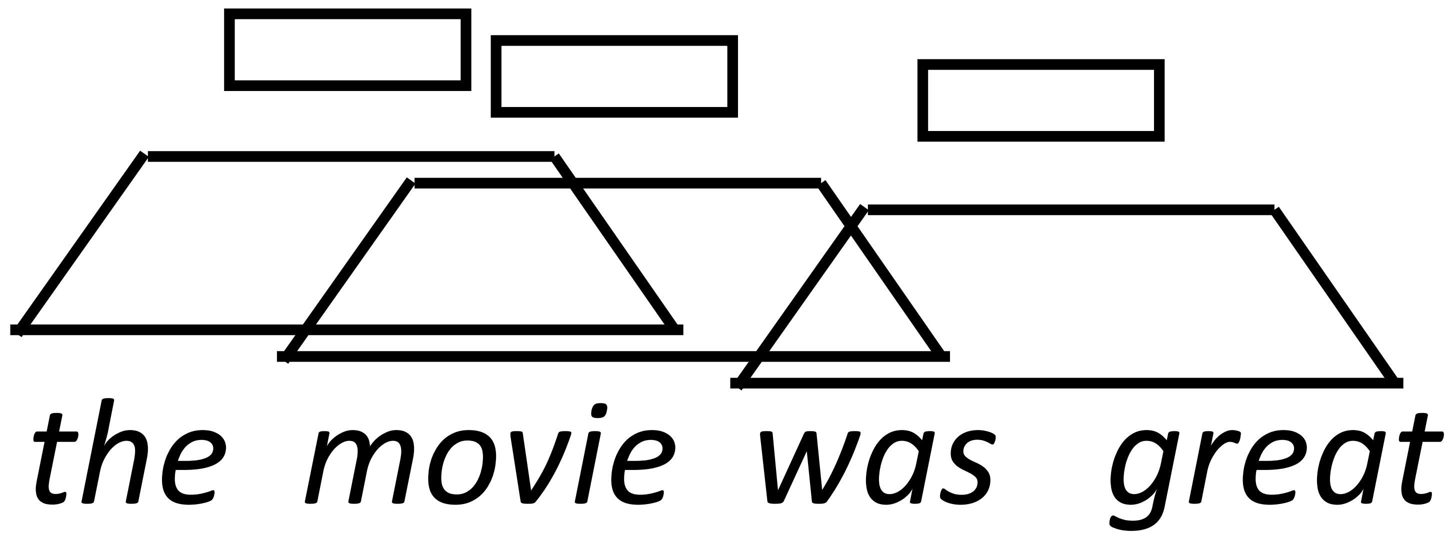
Transformers

Sentence Encoders

- LSTM abstraction: maps each vector in a sentence to a new, context-aware vector



- CNNs do something similar with filters



- Attention can give us a third way to do this

Vaswani et al. (2017)

Self-Attention

- Assume we're using GloVe — what do we want our neural network to do?

The ballerina is very excited that she will dance in the show.

- What words need to be contextualized here?
  - Pronouns need to look at antecedents
  - Ambiguous words should look at context
  - Words should look at syntactic parents/children
- Problem: LSTMs and CNNs don't do this

Vaswani et al. (2017)

Self-Attention

- Want:

The ballerina is very excited that she will dance in the show.

- LSTMs/CNNs: tend to look at local context

The ballerina is very excited that she will dance in the show.

- To appropriately contextualize embeddings, we need to pass information over long distances dynamically for each word

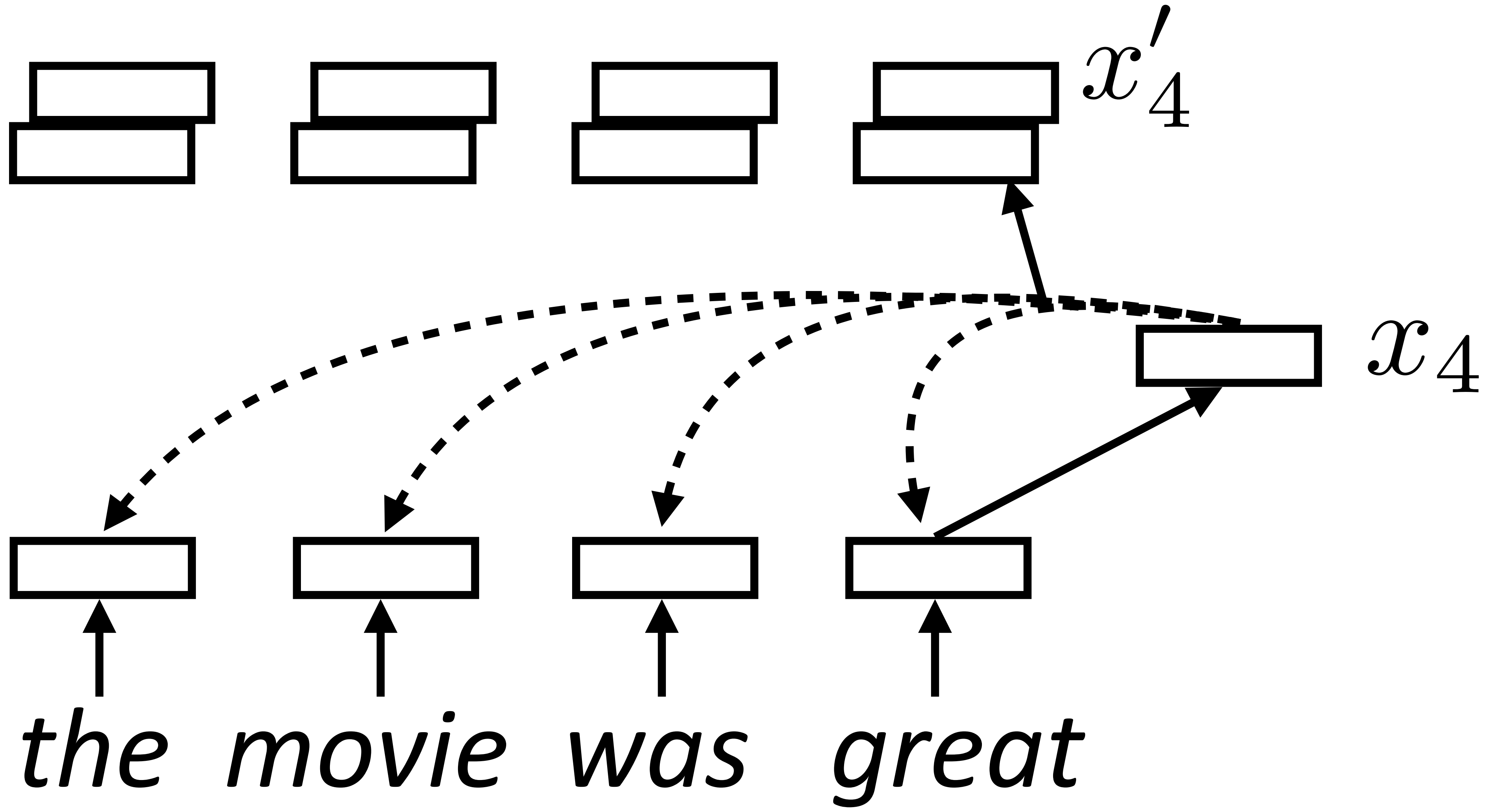
Vaswani et al. (2017)

Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \text{softmax}(x_i^\top x_j) \quad \text{scalar}$$

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector = sum of scalar * vector}$$



- Multiple "heads" analogous to different convolutional filters. Use parameters W_k and V_k to get different attention values + transform vectors

$$\alpha_{k,i,j} = \text{softmax}(x_i^\top W_k x_j) \qquad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j$$

Vaswani et al. (2017)

What can self-attention do?

The ballerina is very excited that she will dance in the show.

0	0.5	0	0	0.1	0.1	0	0.1	0.2	0	0	0

0	0.1	0	0	0	0	0	0	0.5	0	0.4	0

- Attend nearby + to semantically related terms
- This is a demonstration, we will revisit what these models actually learn when we discuss BERT
- Why multiple heads? Softmaxes end up being peaked, single distribution cannot easily put weight on multiple things

Vaswani et al. (2017)

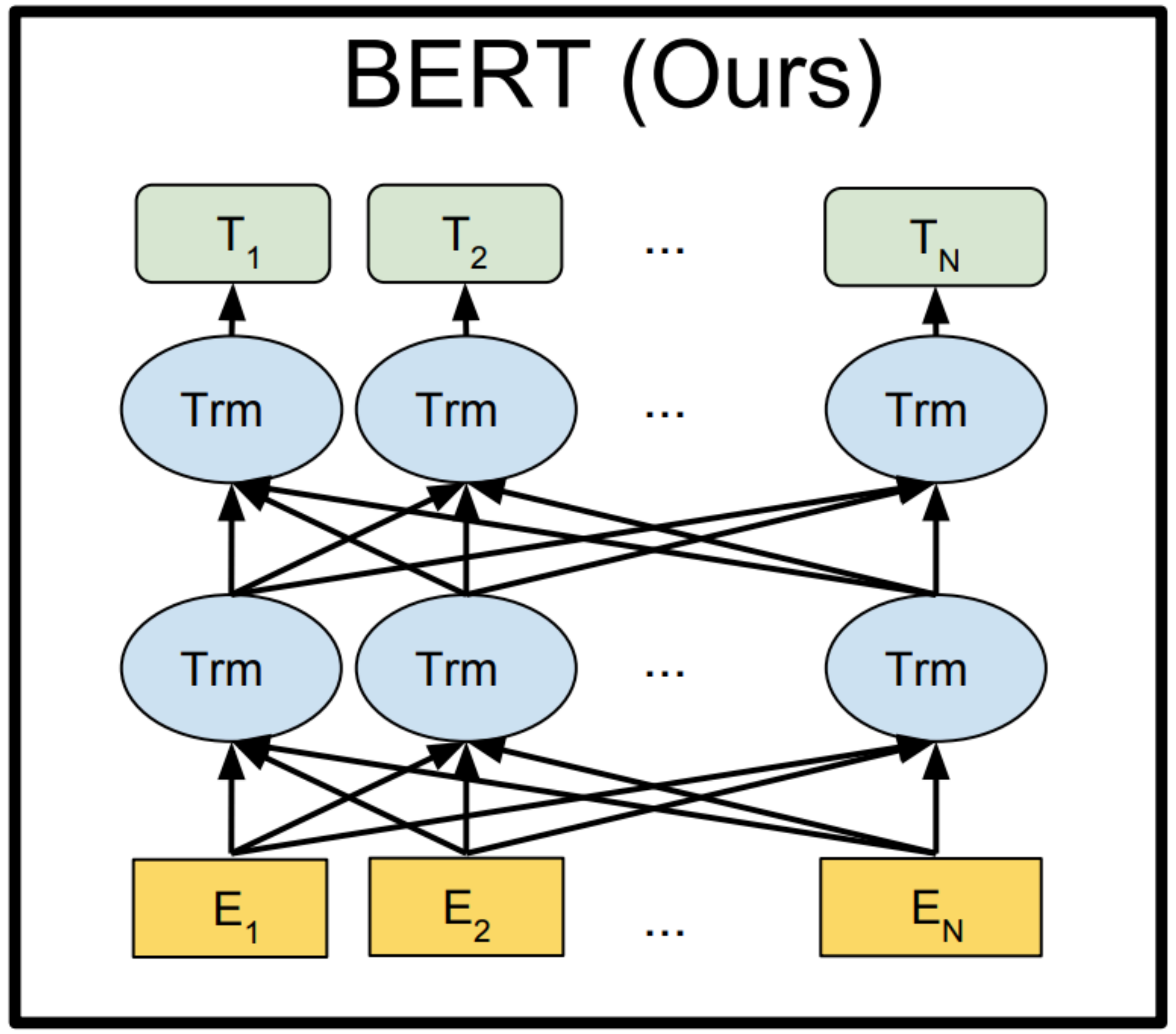
Transformer Uses

- Supervised: transformer can replace LSTM as encoder, decoder, or both; will revisit this when we discuss MT

- Unsupervised: transformers work better than LSTM for unsupervised pre-training of embeddings: predict word given context words

- BERT (Bidirectional Encoder Representations from Transformers): pretraining transformer language models similar to ELMo

- Stronger than similar methods, SOTA on ~11 tasks (including NER — 92.8 F1)



Takeaways

- Attention is very helpful for seq2seq models, and explicit copying can extend this even further
- Transformers are strong models we'll come back to later
- Up next: translation (to finish out seq2seq models)
- Then: pre-trained models and applications