# Copy/Pointer + Self-Attention

Wei Xu

(many slides from Greg Durrett)

# Administrivia

- Mid-semester feedback survey
  - Thanks to many of you who have filled it in!
  - If you haven’t yet, today is a good time to do it.
  - We've responded to some comments on Piazza (likely one more update)
- Midterm is released (due Nov 1st)
- Final course project — will discuss more next class.

# This Lecture

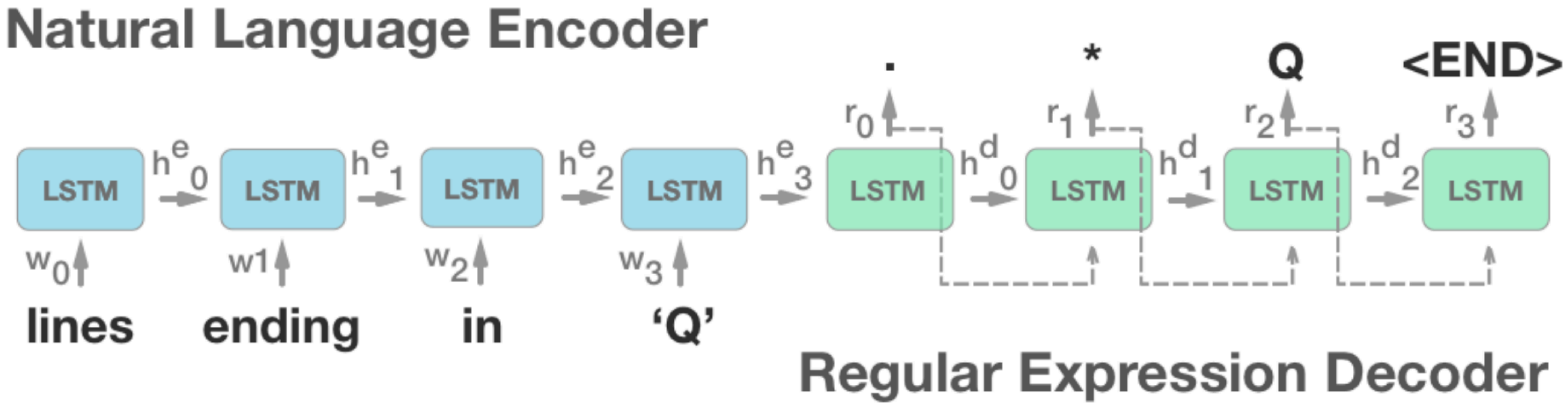
- Applications of Seq2Seq (beyond MT)
- Copy mechanisms for copying words to the output
- Decoding in seq2seq models
- Transformer architecture (if time)

# Other Applications of Seq2Seq

# Regex Prediction

- Seq2seq models can be used for many other tasks!

- Predict regex from text



- Problem: requires a lot of data: 10,000 examples needed to get ~60% accuracy on pretty simple regexes

Locascio et al. (2016)

# Semantic Parsing as Translation

*“what states border Texas”*

↓

```
λ x state( x ) ∧ borders( 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 (2015)

Semantic Parsing/Lambda Calculus: https://www.youtube.com/watch?v=OocGXG-BY6k&t=200s

# SQL Generation

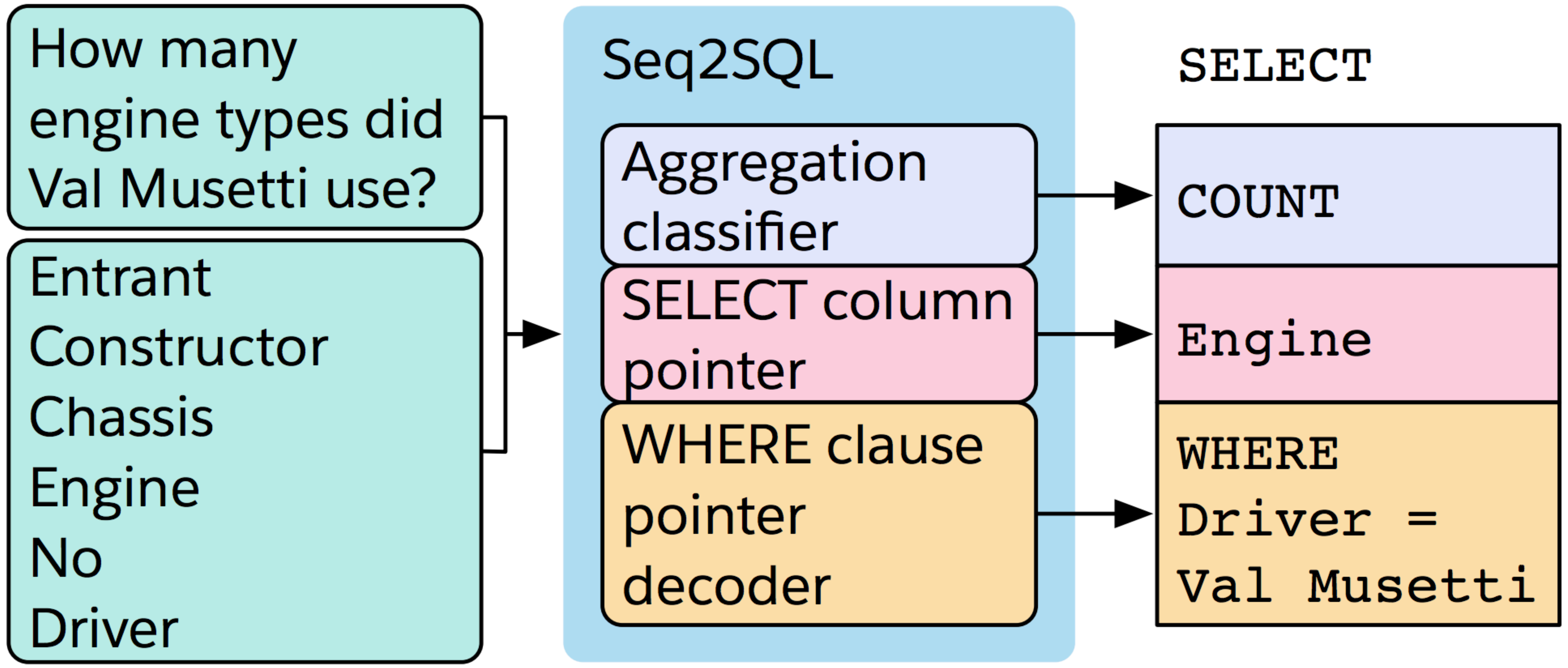
- Convert natural language description into a SQL query against some DB
- How to ensure that well-formed SQL is generated?
  - Three components
- How to capture column names + constants?
  - Pointer mechanisms

Question:

How many CFL teams are from York College?

SQL:

```
SELECT COUNT CFL Team FROM
CFLDraft WHERE College = “York”
```



How many engine types did Val Musetti use?

Entrant
Constructor
Chassis
Engine
No
Driver

Seq2SQL

Aggregation classifier

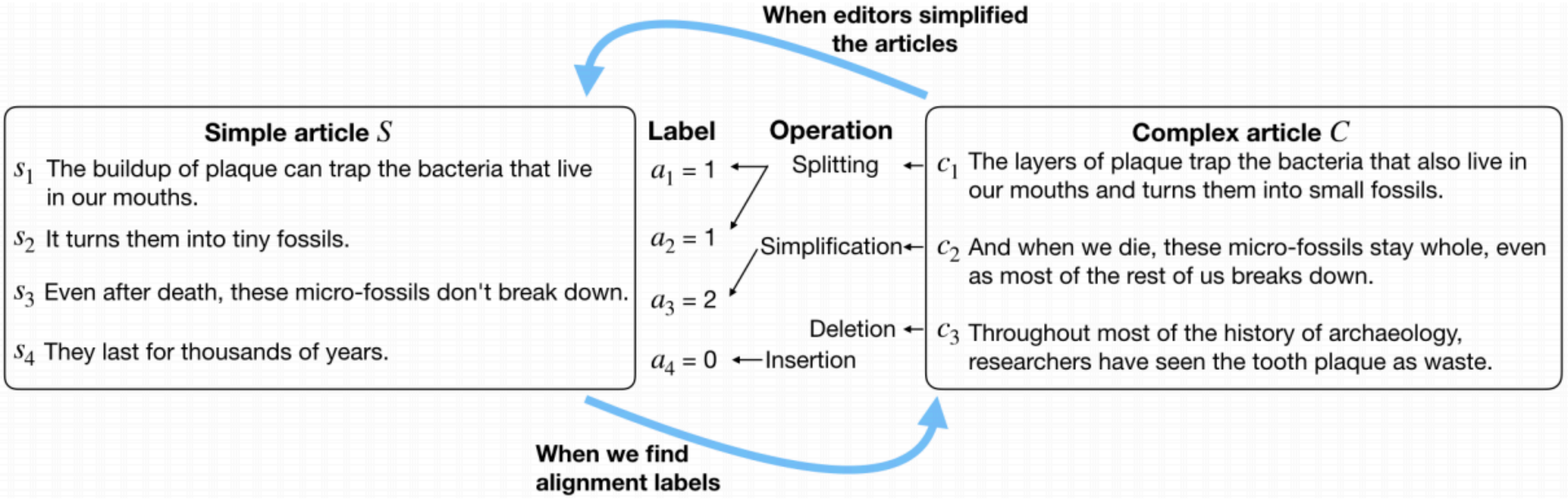
SELECT column pointer

WHERE clause pointer decoder

```
SELECT
COUNT
Engine
WHERE
Driver =
Val Musetti
```

Zhong et al. (2017)

# Text Simplification (Text-to-Text)



When editors simplified the articles

**Simple article $S$**

$s_1$ The buildup of plaque can trap the bacteria that live in our mouths.

$s_2$ It turns them into tiny fossils.

$s_3$ Even after death, these micro-fossils don't break down.

$s_4$ They last for thousands of years.

**Complex article $C$**

$c_1$ The layers of plaque trap the bacteria that also live in our mouths and turns them into small fossils.

$c_2$ And when we die, these micro-fossils stay whole, even as most of the rest of us breaks down.

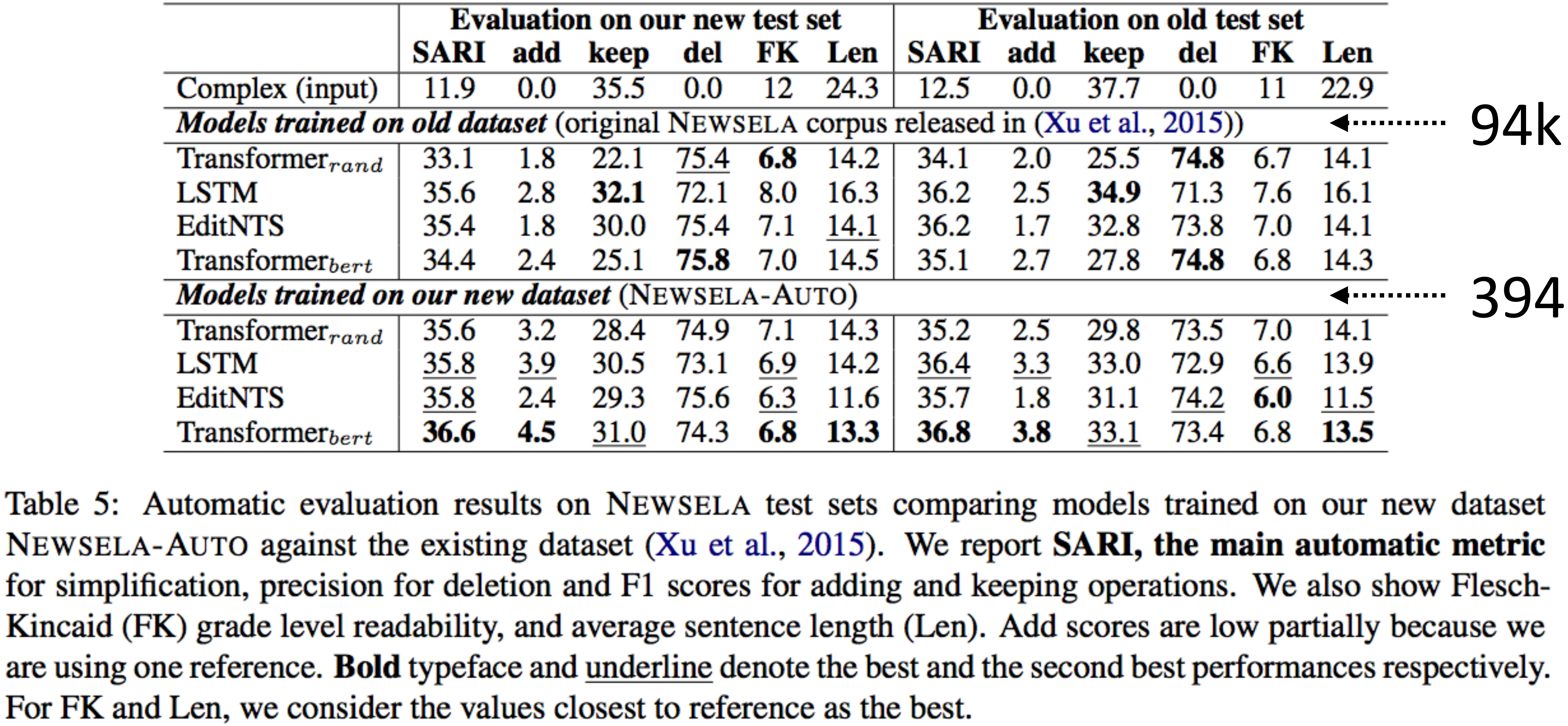
$c_3$ Throughout most of the history of archaeology, researchers have seen the tooth plaque as waste.

When we find alignment labels

Figure 1: An example of sentence alignment between an original news article (right) and its simplified version (left) in Newsela. The label $a_i$ for each simple sentence $s_i$ is the index of complex sentence $c_{a_i}$ it aligned to.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, Wei Xu. "Neural CRF Model for Sentence Alignment in Text Simplification" in ACL (2020)

# Text Simplification



| | Evaluation on our new test set | | | | | | Evaluation on old test set | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **SARI** | **add** | **keep** | **del** | **FK** | **Len** | **SARI** | **add** | **keep** | **del** | **FK** | **Len** |
| Complex (input) | 11.9 | 0.0 | 35.5 | 0.0 | 12 | 24.3 | 12.5 | 0.0 | 37.7 | 0.0 | 11 | 22.9 |
| ***Models trained on old dataset*** (original NEWSELA corpus released in (Xu et al., 2015)) | | | | | | | | | | | | |
| Transformer$_{rand}$ | 33.1 | 1.8 | 22.1 | 75.4 | **6.8** | 14.2 | 34.1 | 2.0 | 25.5 | **74.8** | 6.7 | 14.1 |
| LSTM | 35.6 | 2.8 | **32.1** | 72.1 | 8.0 | 16.3 | 36.2 | 2.5 | **34.9** | 71.3 | 7.6 | 16.1 |
| EditNTS | 35.4 | 1.8 | 30.0 | 75.4 | 7.1 | 14.1 | 36.2 | 1.7 | 32.8 | 73.8 | 7.0 | 14.1 |
| Transformer$_{bert}$ | 34.4 | 2.4 | 25.1 | **75.8** | 7.0 | 14.5 | 35.1 | 2.7 | 27.8 | **74.8** | 6.8 | 14.3 |
| ***Models trained on our new dataset*** (NEWSELA-AUTO) | | | | | | | | | | | | |
| Transformer$_{rand}$ | 35.6 | 3.2 | 28.4 | 74.9 | 7.1 | 14.3 | 35.2 | 2.5 | 29.8 | 73.5 | 7.0 | 14.1 |
| LSTM | 35.8 | 3.9 | 30.5 | 73.1 | 6.9 | 14.2 | 36.4 | 3.3 | 33.0 | 72.9 | 6.6 | 13.9 |
| EditNTS | 35.8 | 2.4 | 29.3 | 75.6 | 6.3 | 11.6 | 35.7 | 1.8 | 31.1 | 74.2 | **6.0** | 11.5 |
| Transformer$_{bert}$ | **36.6** | **4.5** | 31.0 | 74.3 | **6.8** | **13.3** | **36.8** | **3.8** | 33.1 | 73.4 | 6.8 | **13.5** |

Table 5: Automatic evaluation results on NEWSELA test sets comparing models trained on our new dataset NEWSELA-AUTO against the existing dataset (Xu et al., 2015). We report **SARI, the main automatic metric** for simplification, precision for deletion and F1 scores for adding and keeping operations. We also show Flesch-Kincaid (FK) grade level readability, and average sentence length (Len). Add scores are low partially because we are using one reference. **Bold** typeface and underline denote the best and the second best performances respectively. For FK and Len, we consider the values closest to reference as the best.

94k sent. pairs

394k sent. pairs

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, Wei Xu. "Neural CRF Model for Sentence Alignment in Text Simplification" in ACL (2020)

# Copy / Pointer Networks

# Unknown Words

*en*: The *ecotax* portico in *Pont-de-Buis* , . . . [truncated] . . . , was taken down on Thursday morning

*fr*: Le *portique* *écotaxe* de *Pont-de-Buis* , . . . [truncated] . . . , a été *démonté* jeudi matin

*nn*: Le unk de unk à unk , . . . [truncated] . . . , a été pris le jeudi matin

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

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

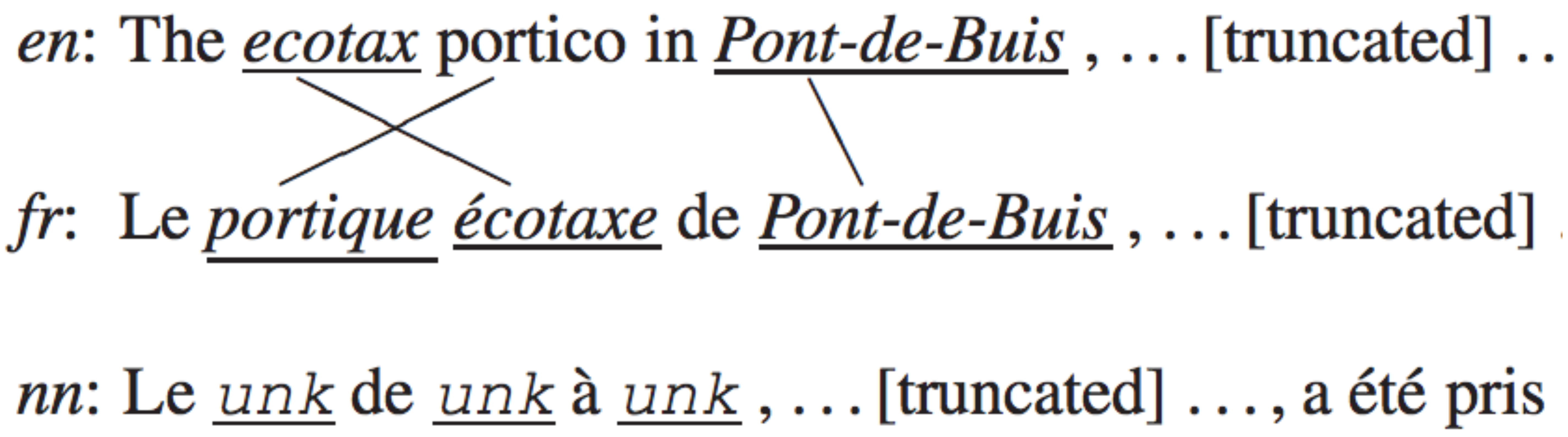
from attention

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)

# 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

Le
de
…
matin

---

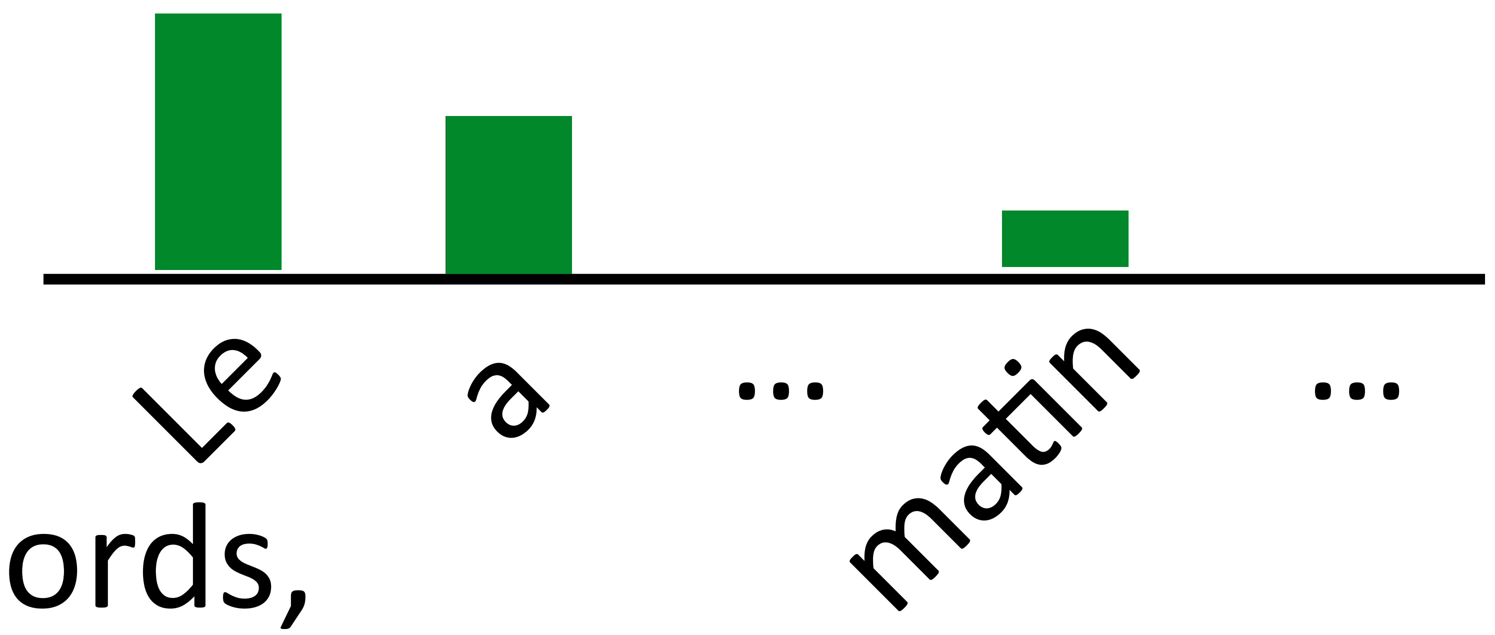
Pont-de-Buis
ecotax

- Some words we want to copy may not be in the fixed output vocab (*Pont-de-Buis*)
- Solution: Vocabulary contains "normal" vocab as well as words in input.

# Pointer Networks

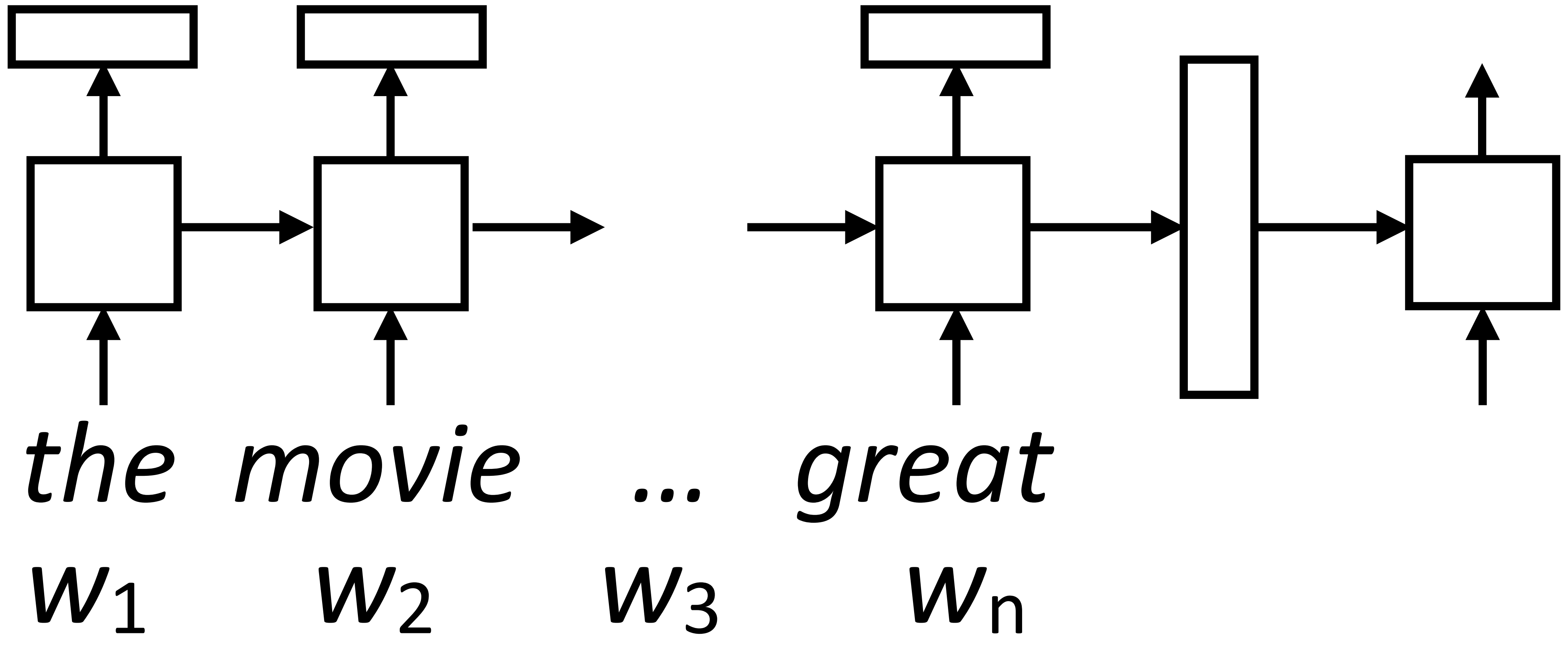
- Standard decoder ($P_{\text{vocab}}$): softmax over vocabulary

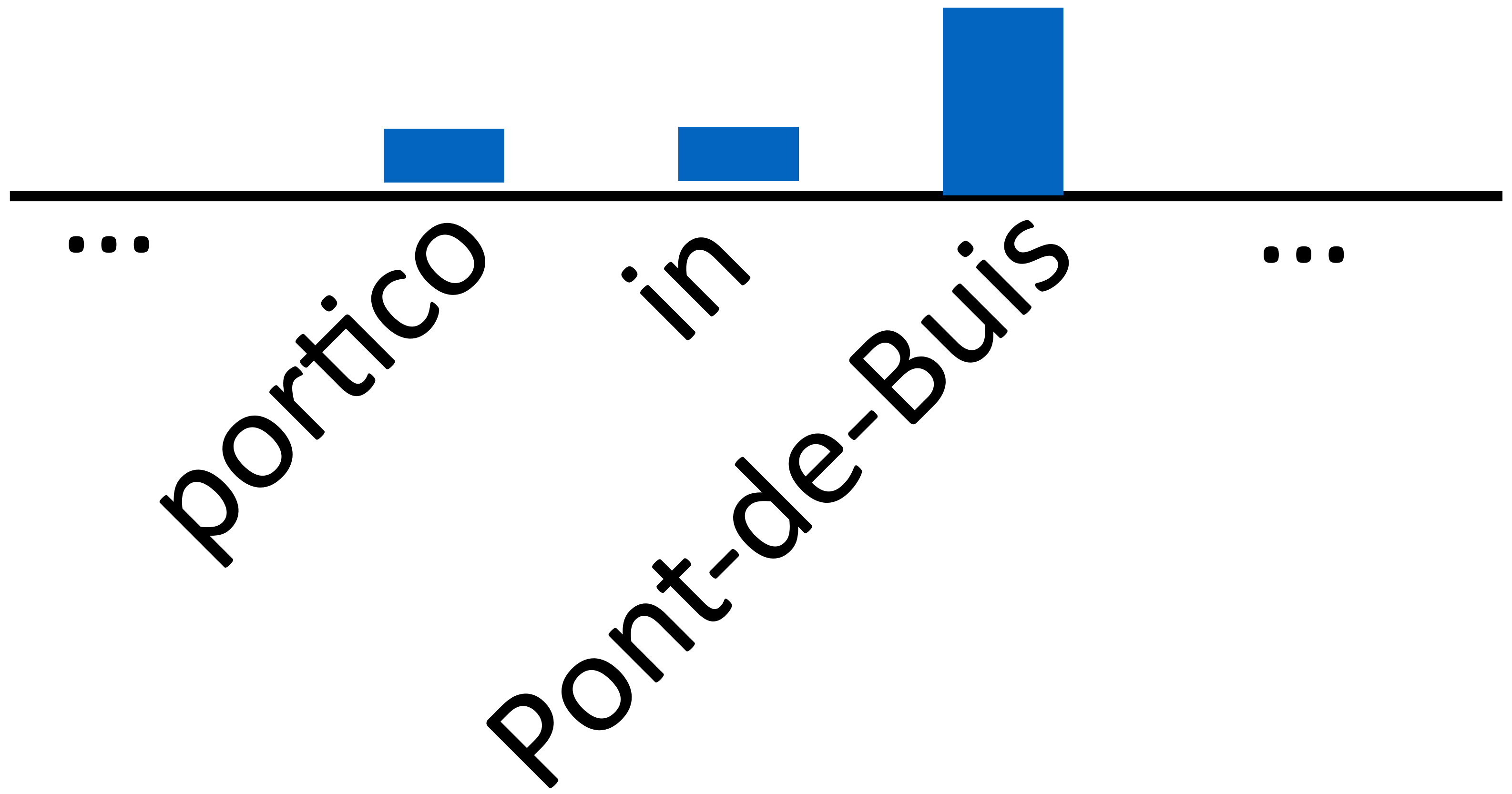
$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W[c_i; \bar{h}_i])$$



- Pointer network ($P_{\text{pointer}}$): predict from *source* words, instead of target vocabulary

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





# Pointer Generator Mixture Models

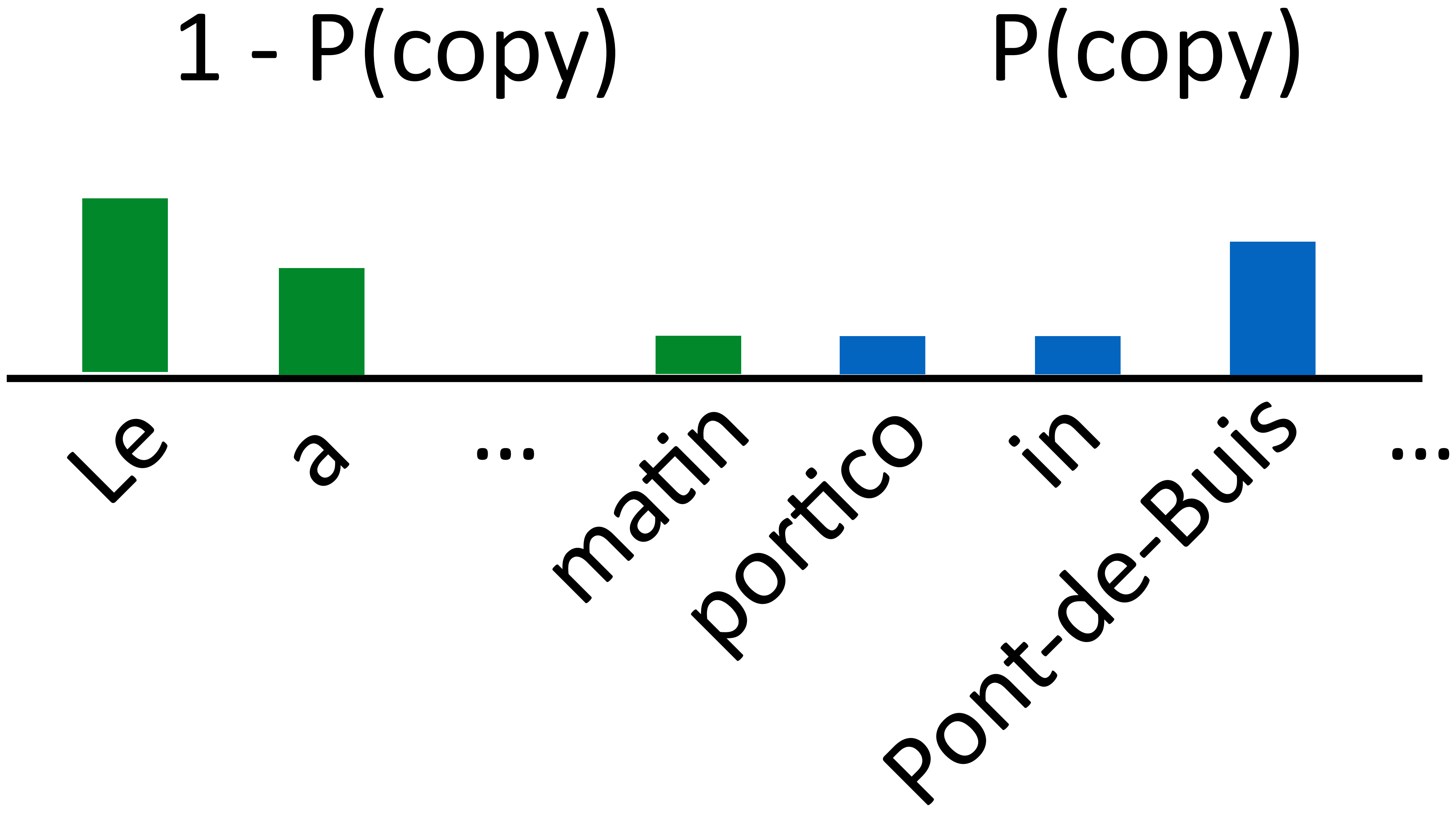
- Define the decoder model as a mixture model of $P_{\text{vocab}}$ and $P_{\text{pointer}}$

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

- Predict P(copy) based on decoder state, input, etc.

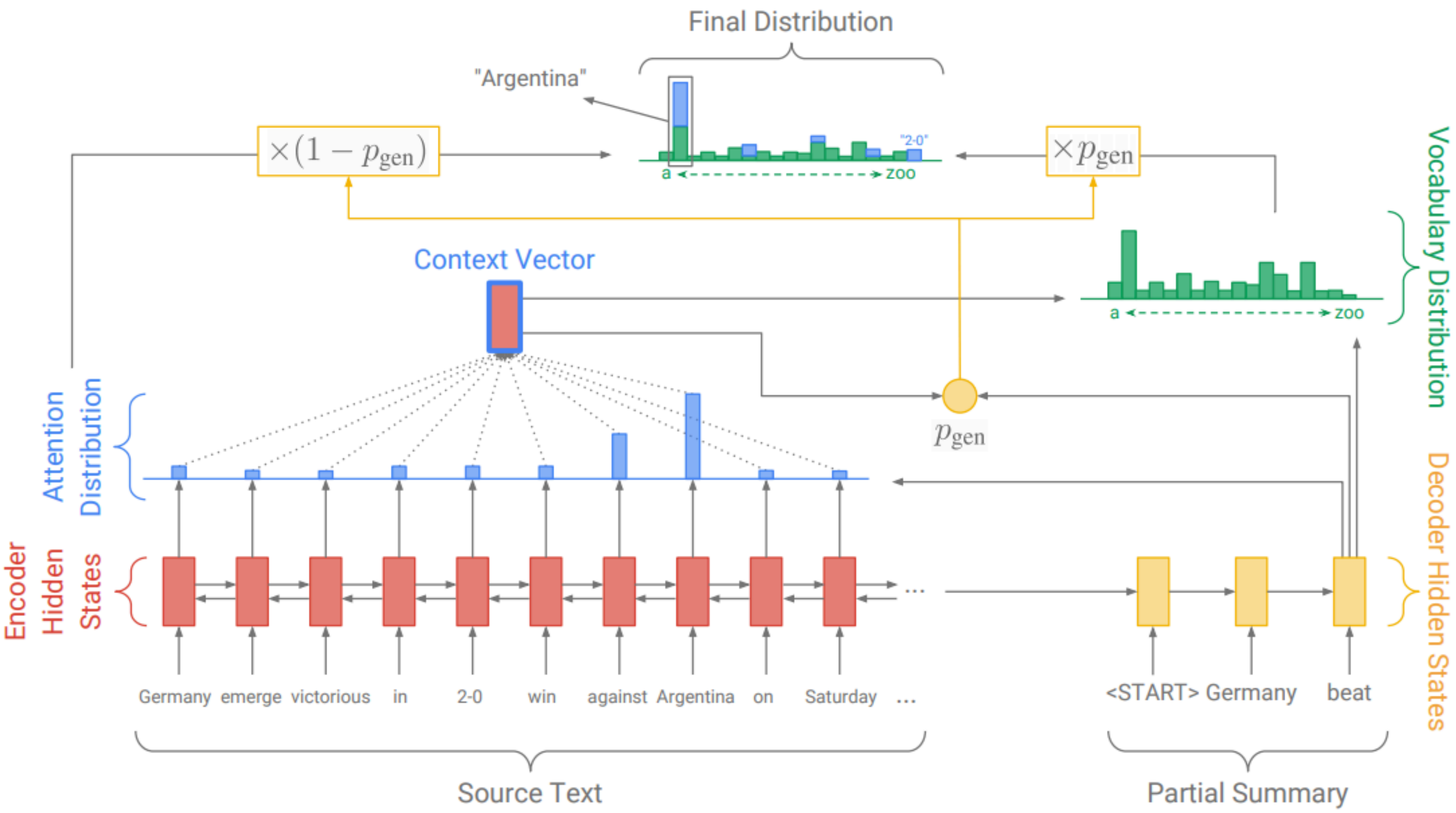
- Marginalize over copy variable during training and inference

- Model will be able to both generate and copy, flexibly adapt between the two



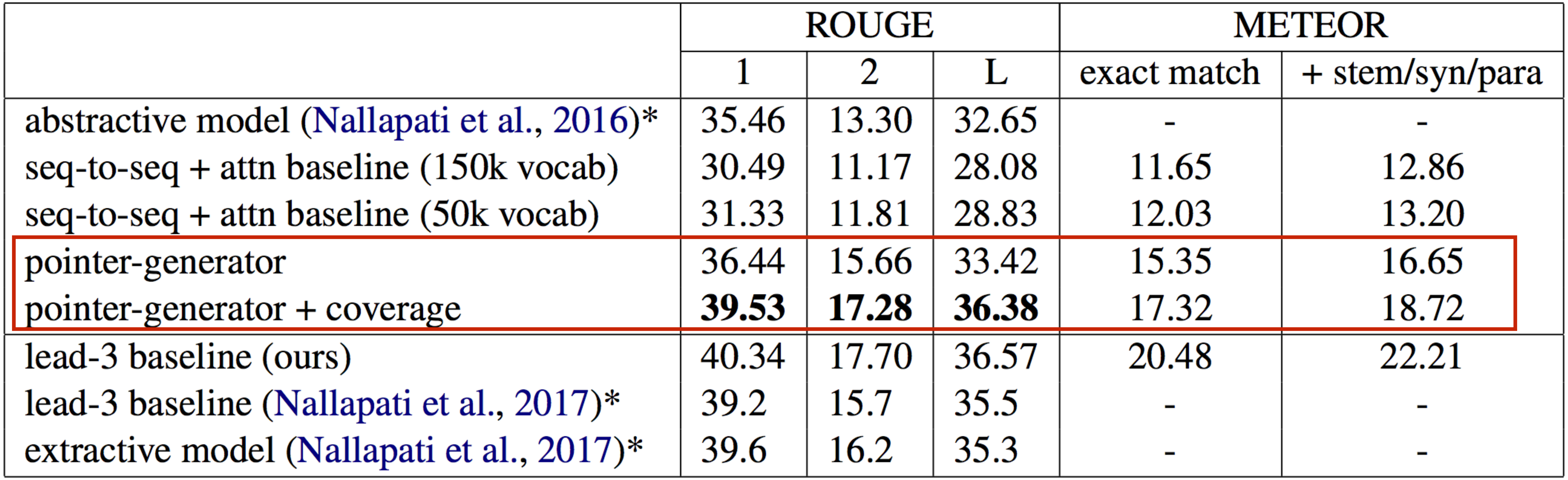
Gulcehre et al. (2016), Gu et al. (2016)

# Copying in Summarization



See et al. (2017)

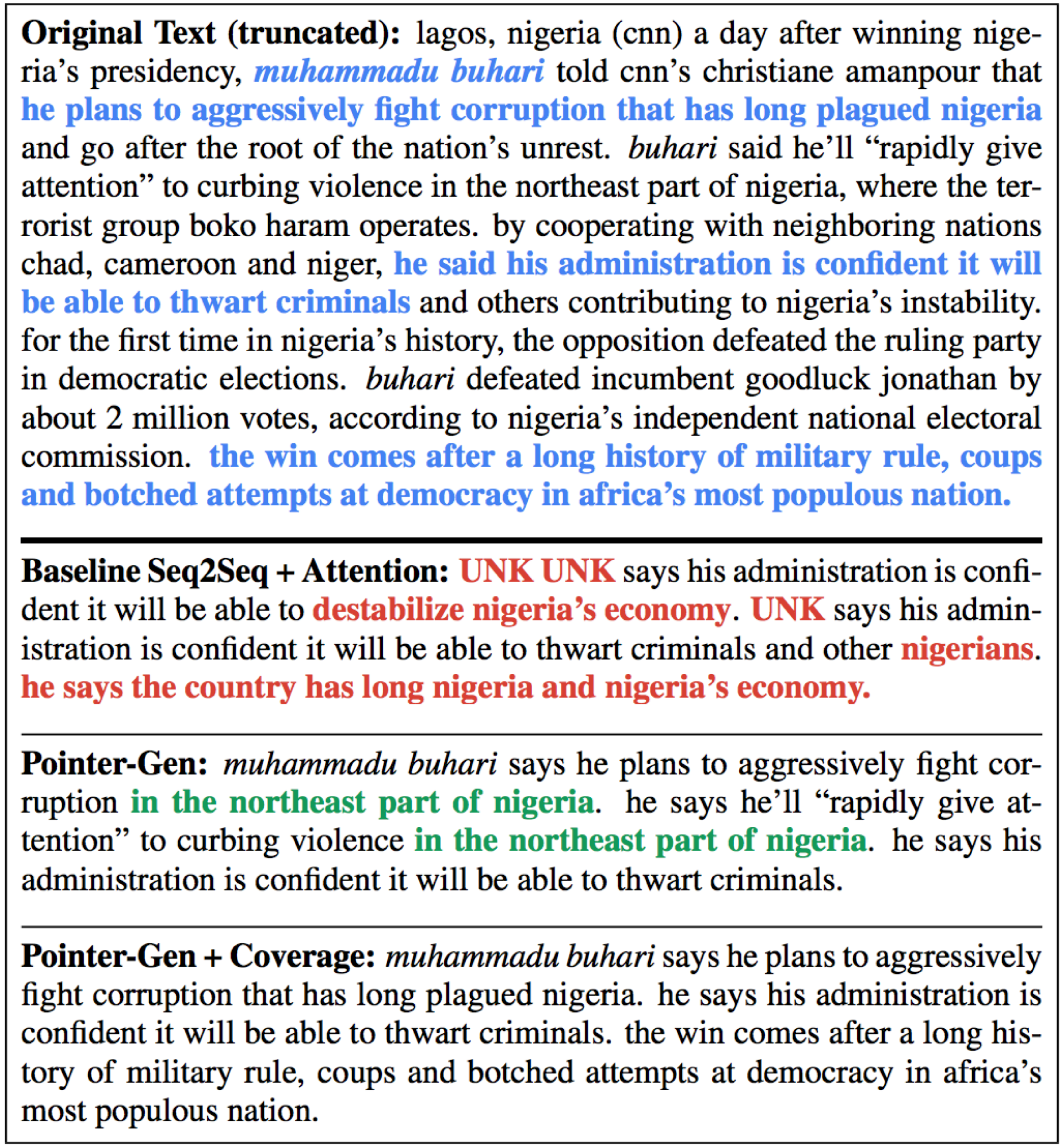
# Copying in Summarization



| | ROUGE | | | METEOR | |
|---|---|---|---|---|---|
| | 1 | 2 | L | exact match | + stem/syn/para |
| abstractive model (Nallapati et al., 2016)* | 35.46 | 13.30 | 32.65 | - | - |
| seq-to-seq + attn baseline (150k vocab) | 30.49 | 11.17 | 28.08 | 11.65 | 12.86 |
| seq-to-seq + attn baseline (50k vocab) | 31.33 | 11.81 | 28.83 | 12.03 | 13.20 |
| pointer-generator | 36.44 | 15.66 | 33.42 | 15.35 | 16.65 |
| pointer-generator + coverage | **39.53** | **17.28** | **36.38** | 17.32 | 18.72 |
| lead-3 baseline (ours) | 40.34 | 17.70 | 36.57 | 20.48 | 22.21 |
| lead-3 baseline (Nallapati et al., 2017)* | 39.2 | 15.7 | 35.5 | - | - |
| extractive model (Nallapati et al., 2017)* | 39.6 | 16.2 | 35.3 | - | - |

See et al. (2017)

# Copying in Summarization



**Original Text (truncated):** lagos, nigeria (cnn) a day after winning nigeria's presidency, ***muhammadu buhari*** told cnn's christiane amanpour that **he plans to aggressively fight corruption that has long plagued nigeria** and go after the root of the nation's unrest. *buhari* said he'll "rapidly give attention" to curbing violence in the northeast part of nigeria, where the terrorist group boko haram operates. by cooperating with neighboring nations chad, cameroon and niger, **he said his administration is confident it will be able to thwart criminals** and others contributing to nigeria's instability. for the first time in nigeria's history, the opposition defeated the ruling party in democratic elections. *buhari* defeated incumbent goodluck jonathan by about 2 million votes, according to nigeria's independent national electoral commission. **the win comes after a long history of military rule, coups and botched attempts at democracy in africa's most populous nation.**

**Baseline Seq2Seq + Attention:** UNK UNK says his administration is confident it will be able to **destabilize nigeria's economy**. UNK says his administration is confident it will be able to thwart criminals and other **nigerians**. **he says the country has long nigeria and nigeria's economy.**

**Pointer-Gen:** *muhammadu buhari* says he plans to aggressively fight corruption **in the northeast part of nigeria**. he says he'll "rapidly give attention" to curbing violence **in the northeast part of nigeria**. he says his administration is confident it will be able to thwart criminals.

**Pointer-Gen + Coverage:** *muhammadu buhari* says he plans to aggressively fight corruption that has long plagued nigeria. he says his administration is confident it will be able to thwart criminals. the win comes after a long history of military rule, coups and botched attempts at democracy in africa's most populous nation.

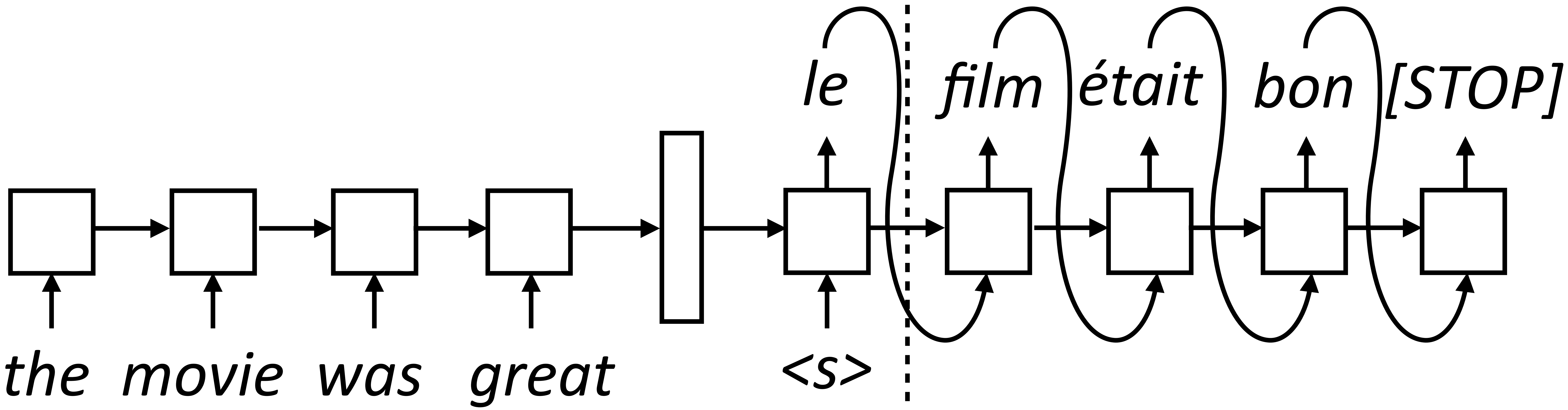
Figure 1: Comparison of output of 3 abstractive summarization models on a news article. The baseline model makes **factual errors**, a **nonsensical sentence** and struggles with OOV words *muhammadu buhari*. The pointer-generator model is accurate but **repeats itself**. Coverage eliminates repetition. The final summary is composed from **several fragments**.

See et al. (2017)

# 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**

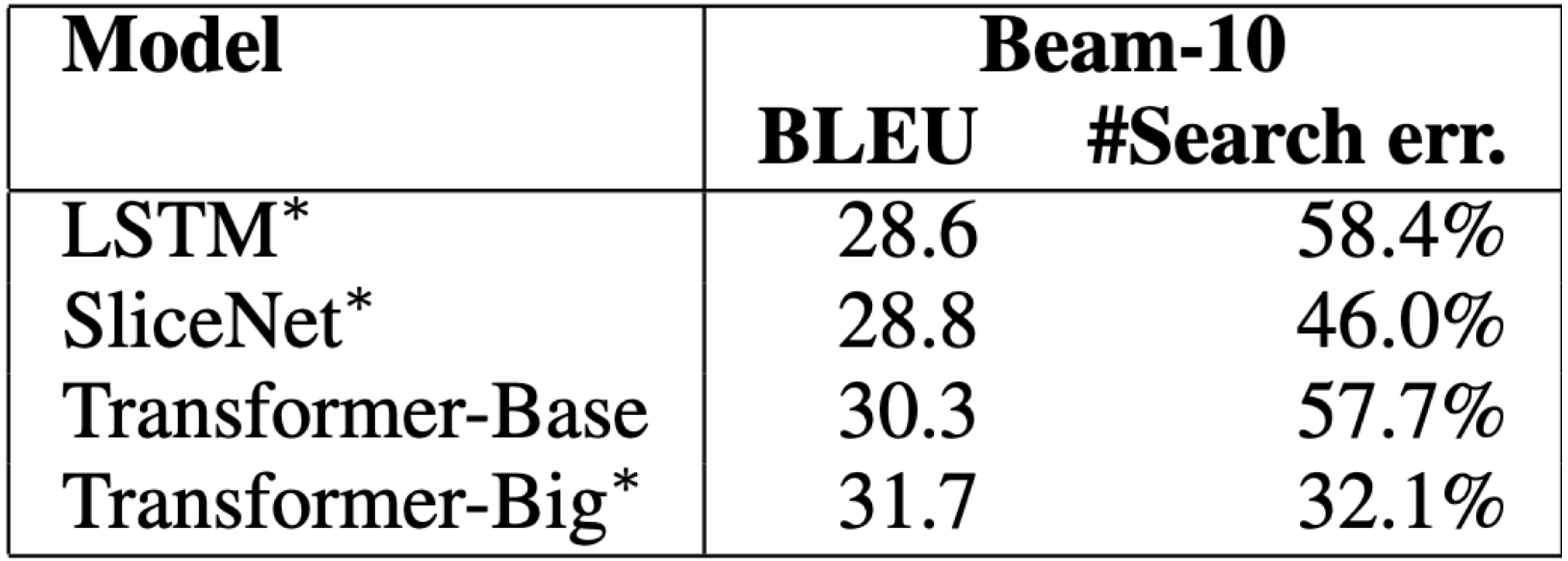
$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W\bar{h})$$

(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)



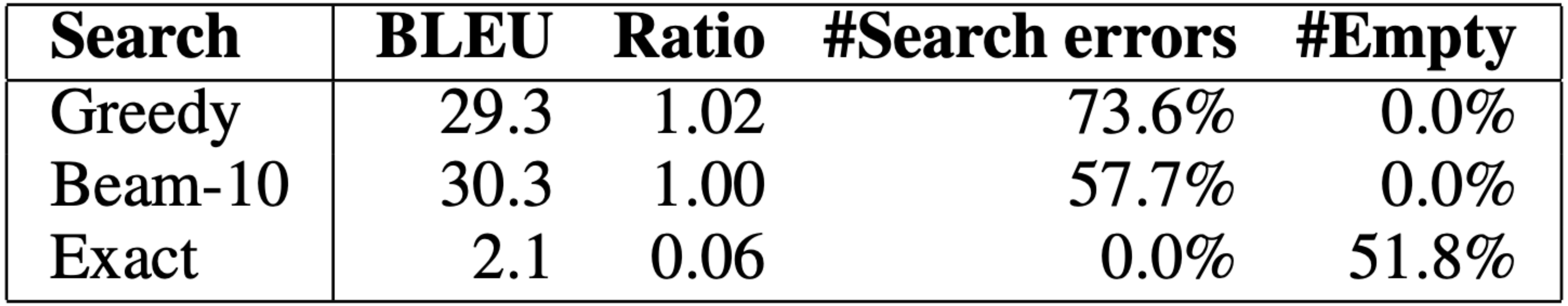
| Model | Beam-10 | |
|---|---|---|
| | BLEU | #Search err. |
| LSTM* | 28.6 | 58.4% |
| SliceNet* | 28.8 | 46.0% |
| Transformer-Base | 30.3 | 57.7% |
| Transformer-Big* | 31.7 | 32.1% |

A sentence is classified as search error if the decoder does not find the global best model score.

Stahlberg and Byrne (2019)

# “Problems” with Beam Decoding

- For machine translation, the highest probability sequence is often the empty string, i.e.. a single </s> token!   (>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

- Exact inference uses depth-first search, but cut off branches that fall below a lower bound.

Stahlberg and Byrne (2019)

# Sampling

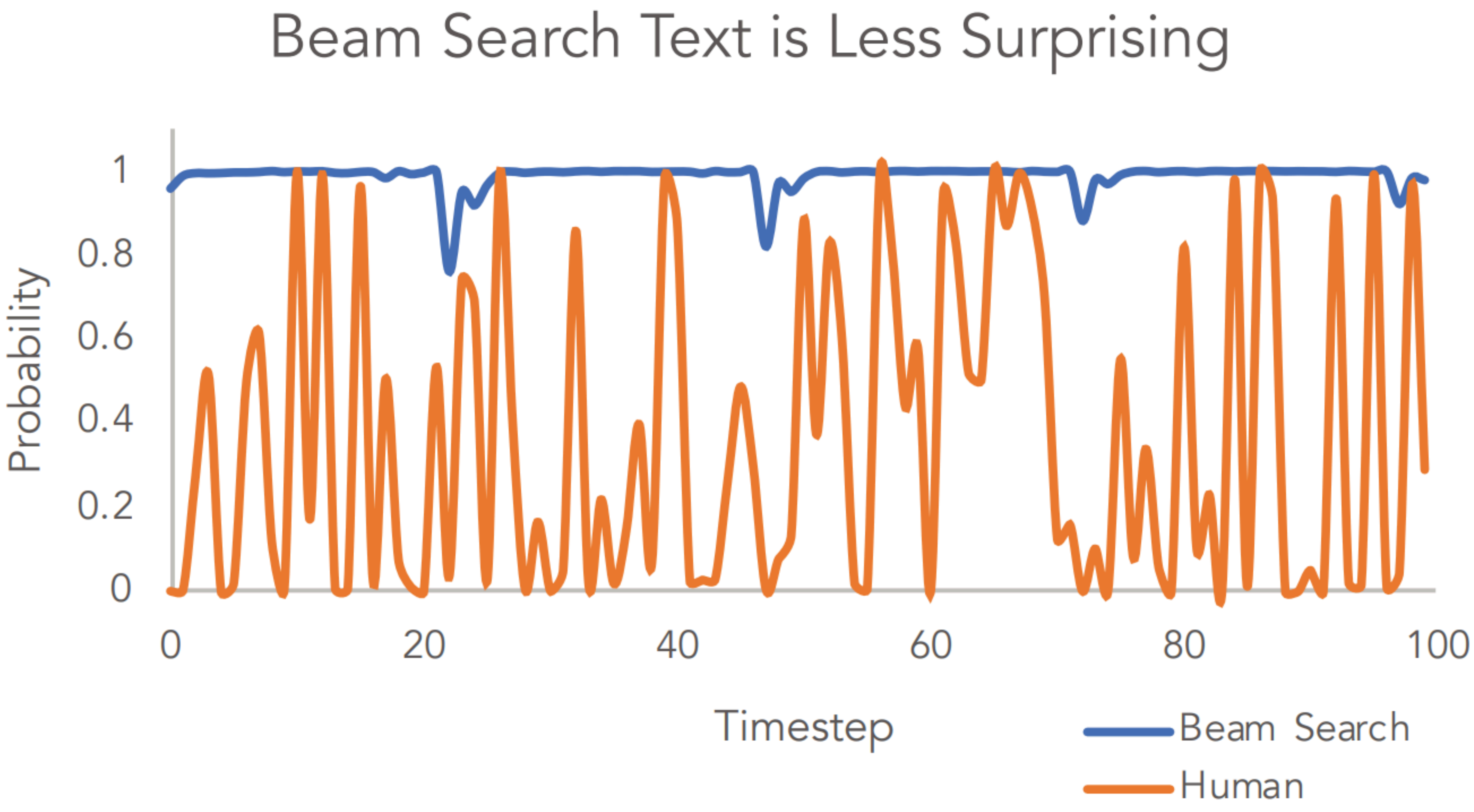
- Beam search may give many similar sequences, and these actually may be *too close* to the optimal. Can sample instead:

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

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

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

# Beam Search vs. Sampling



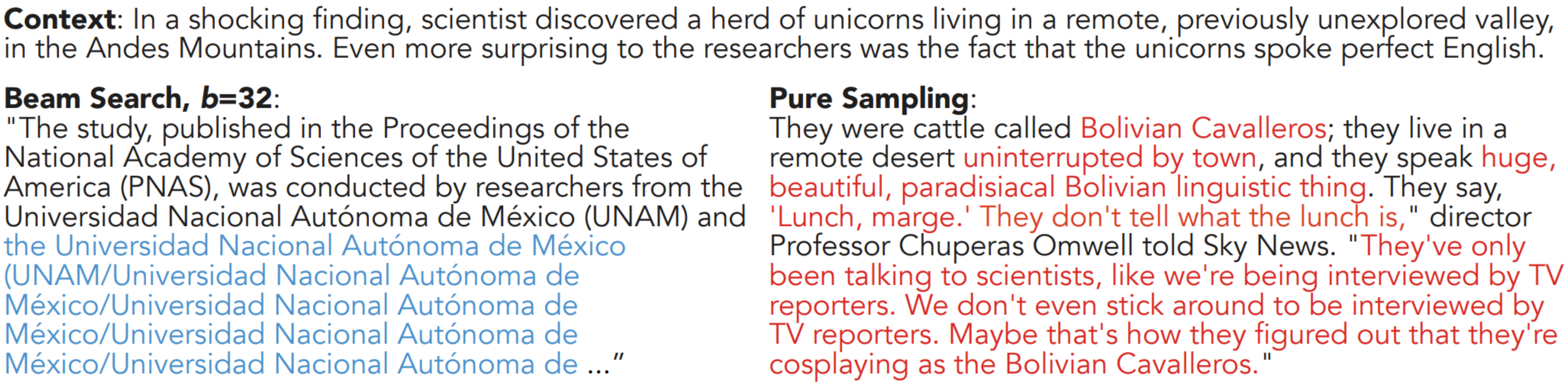
Holtzman et al. (2019)

# Decoding Strategies

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

# 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 ..."

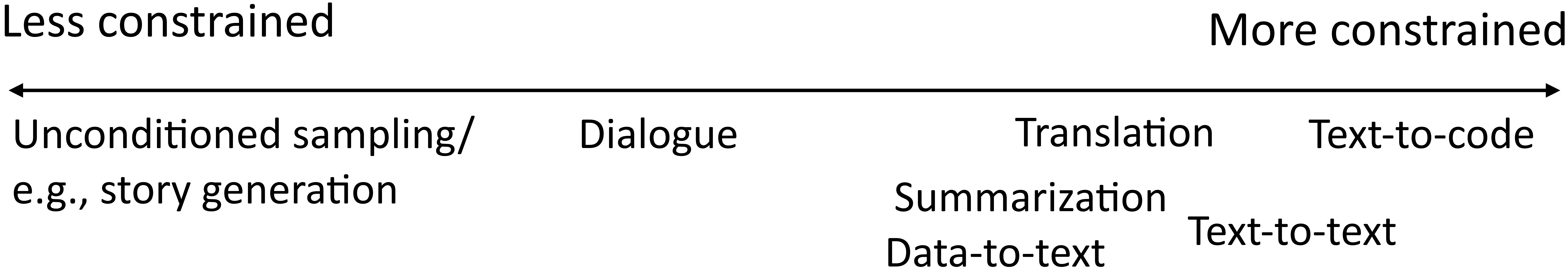
**Pure Sampling**:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've 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)

# 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/ e.g., story generation

Dialogue

Translation

Text-to-code

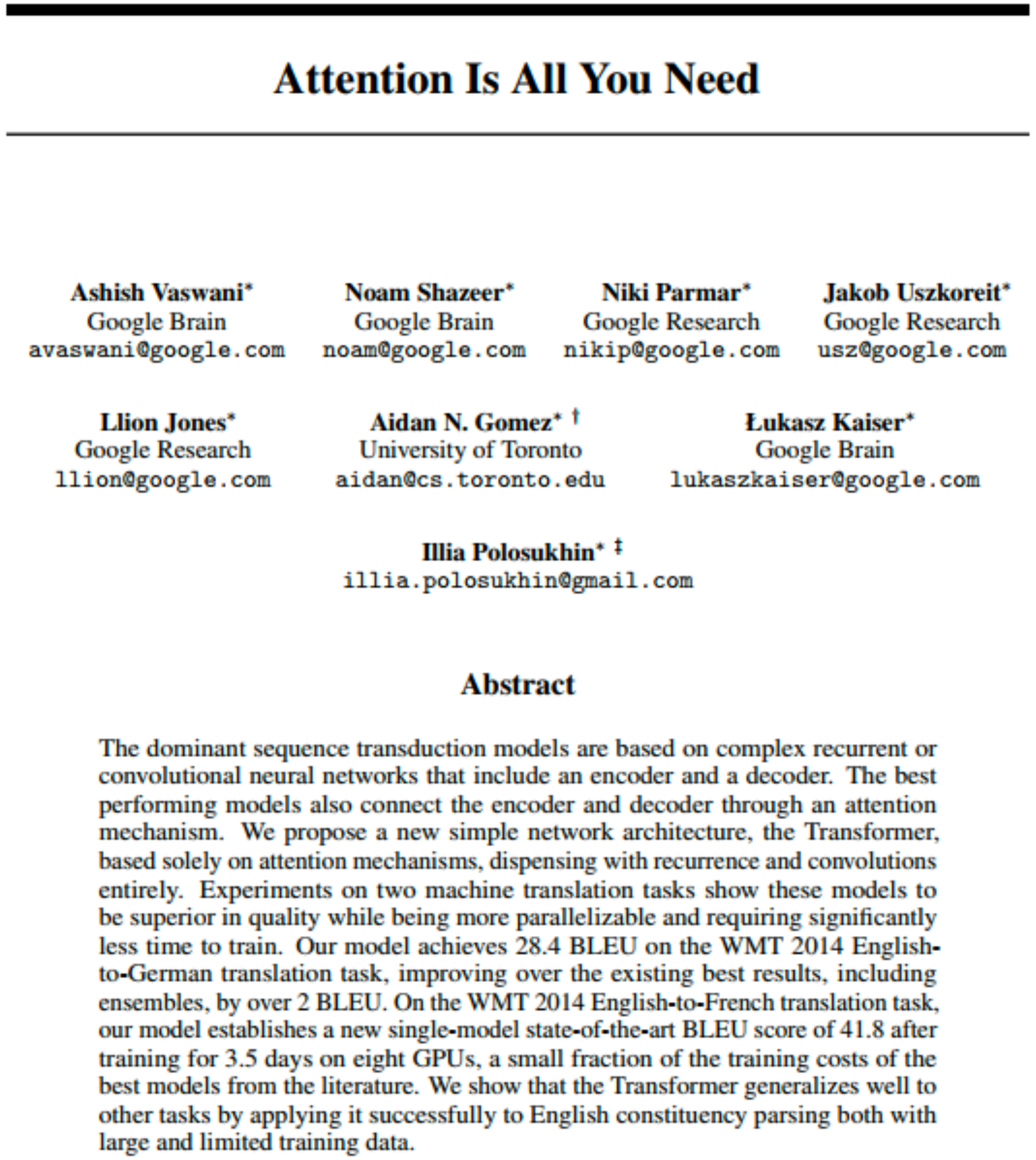
Summarization

Data-to-text

Text-to-text

# Transformers

# Attention is All You Need



## Attention Is All You Need

**Ashish Vaswani*** Google Brain avaswani@google.com

**Noam Shazeer*** Google Brain noam@google.com

**Niki Parmar*** Google Research nikip@google.com

**Jakob Uszkoreit*** Google Research usz@google.com

**Llion Jones*** Google Research llion@google.com

**Aidan N. Gomez* †** University of Toronto aidan@cs.toronto.edu

**Łukasz Kaiser*** Google Brain lukaszkaiser@google.com

**Illia Polosukhin* ‡** illia.polosukhin@gmail.com

### Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

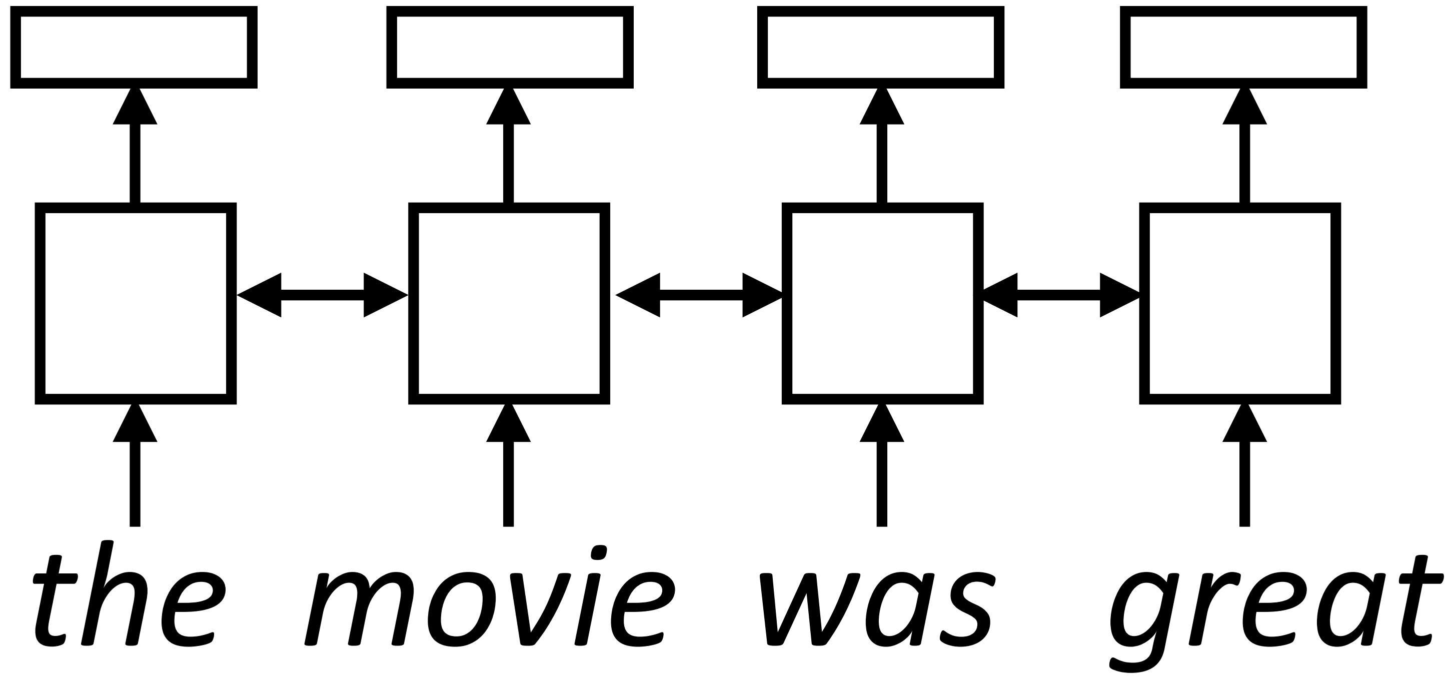
Vaswani et al. (2017)

# Readings

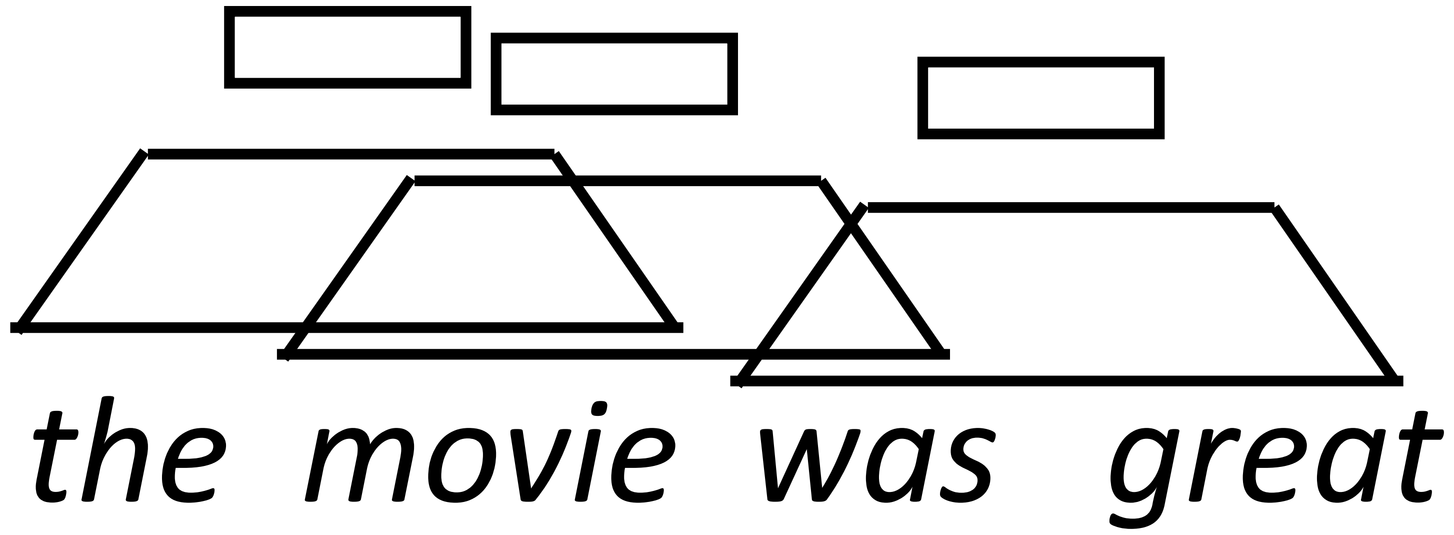
- “The Annotated Transformer” by Sasha Rush
  https://nlp.seas.harvard.edu/2018/04/03/attention.html
- “The Illustrated Transformer” by Jay Lamar
  http://jalammar.github.io/illustrated-transformer/

# 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?

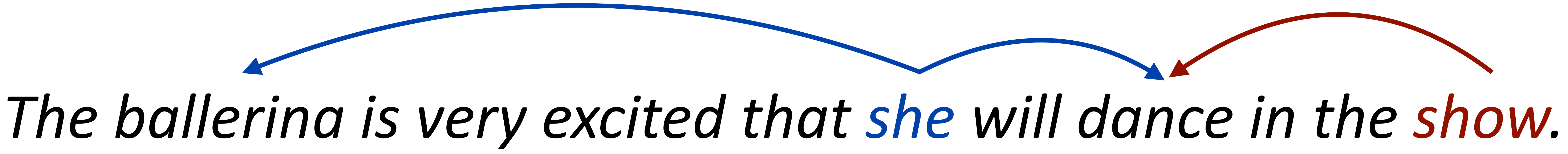
*The ballerina is very excited that she will dance in the show.*

- Q: What words need to be contextualized here?

Vaswani et al. (2017)

# Self-Attention

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



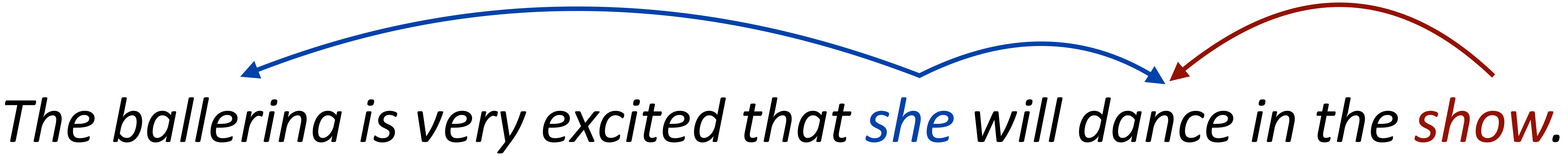
*The ballerina is very excited that she will dance in the show.*

- 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)

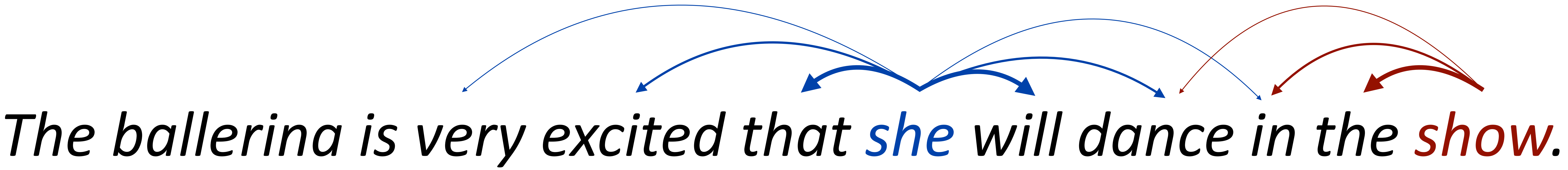
# Self-Attention

- 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

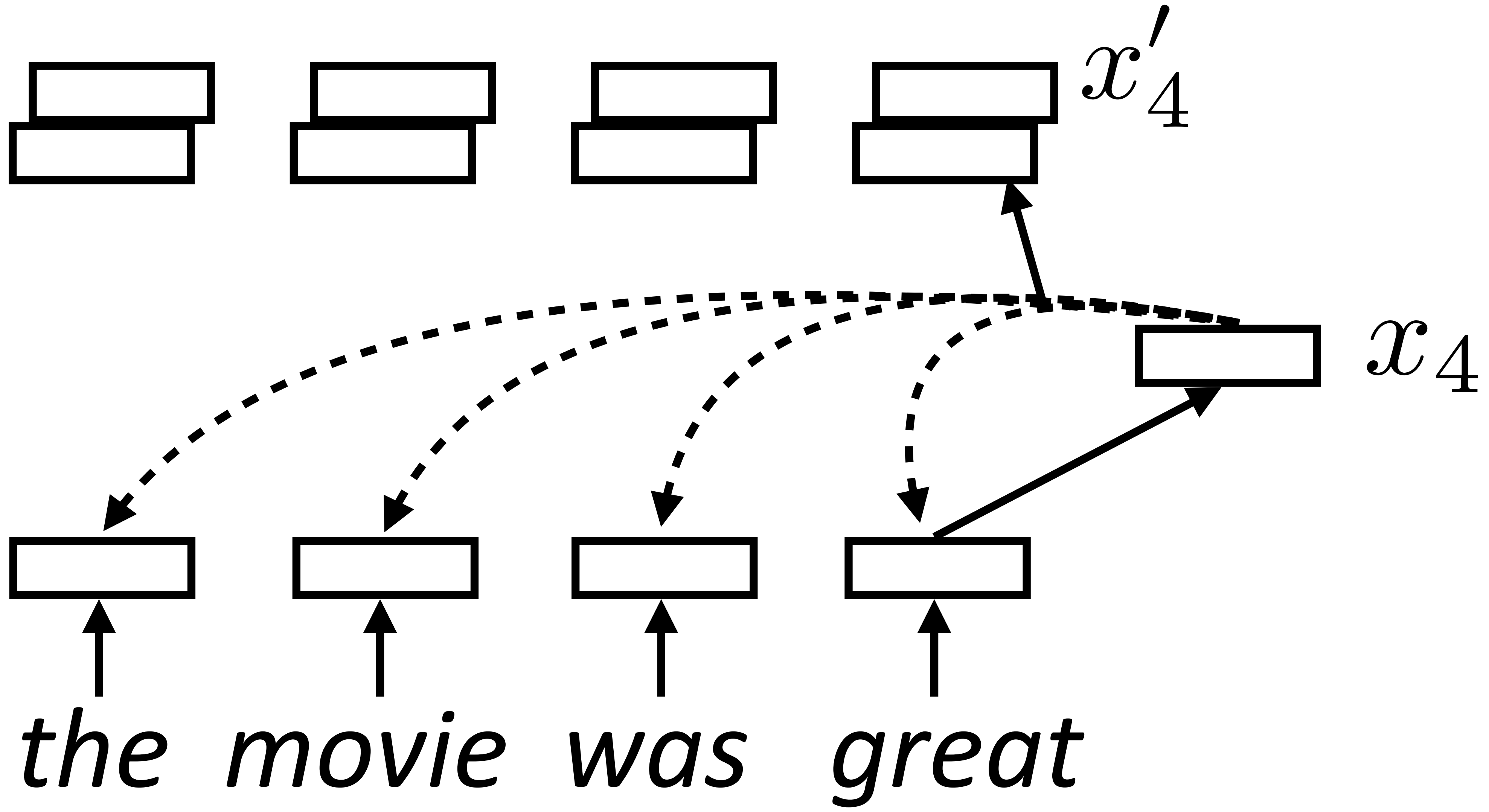
Vaswani et al. (2017)

# Self-Attention

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

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

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

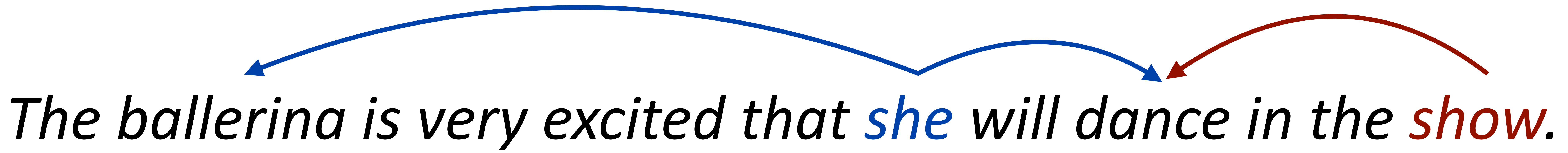


- Multiple “heads” analogous to different convolutional filters. Use parameters *W<sub>k</sub>* and *V<sub>k</sub>* to get different attention values + transform vectors

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

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 |

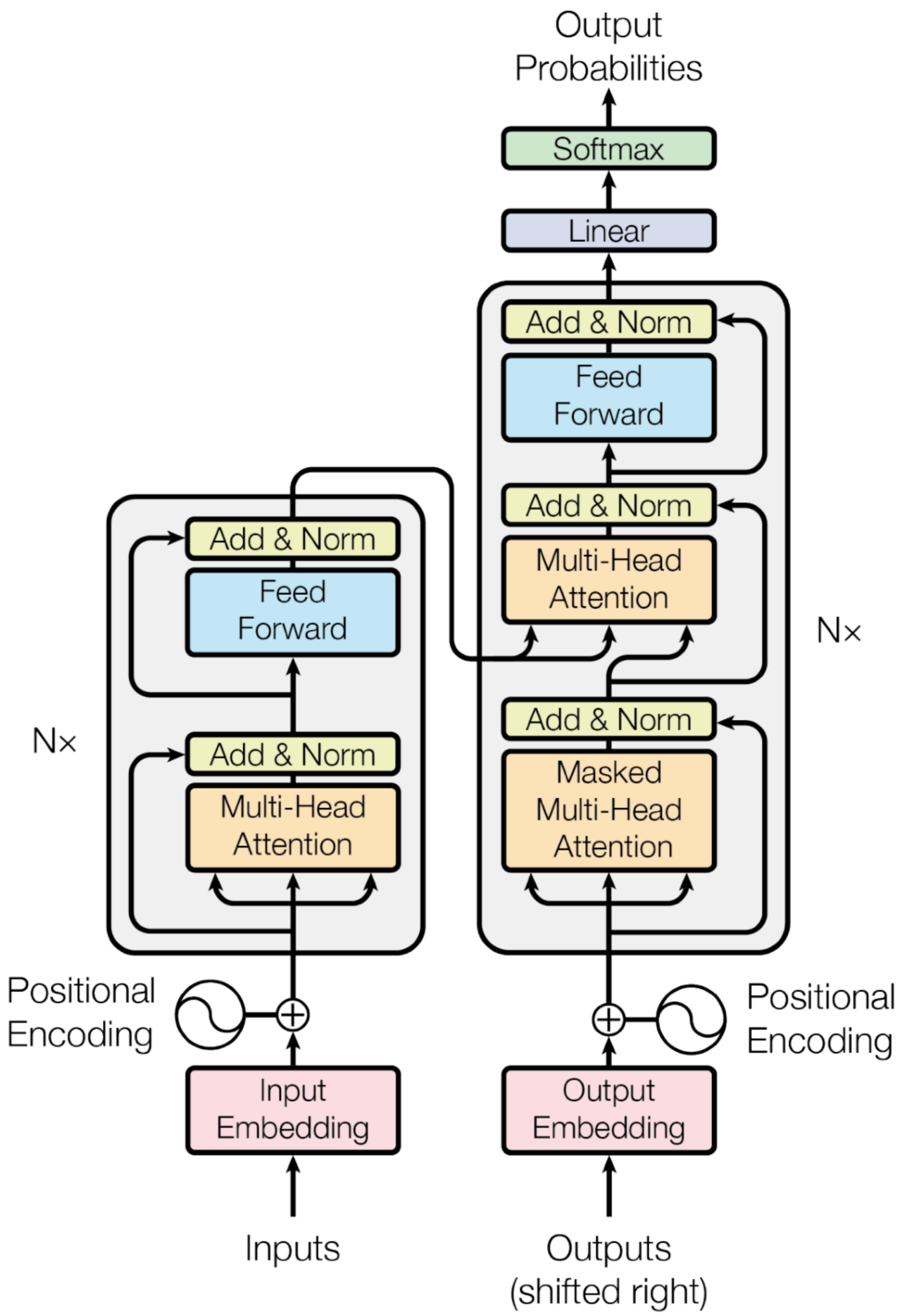
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.5 | 0 | 0.4 | 0 |

- 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; such as in machine translation and natural language generation tasks.

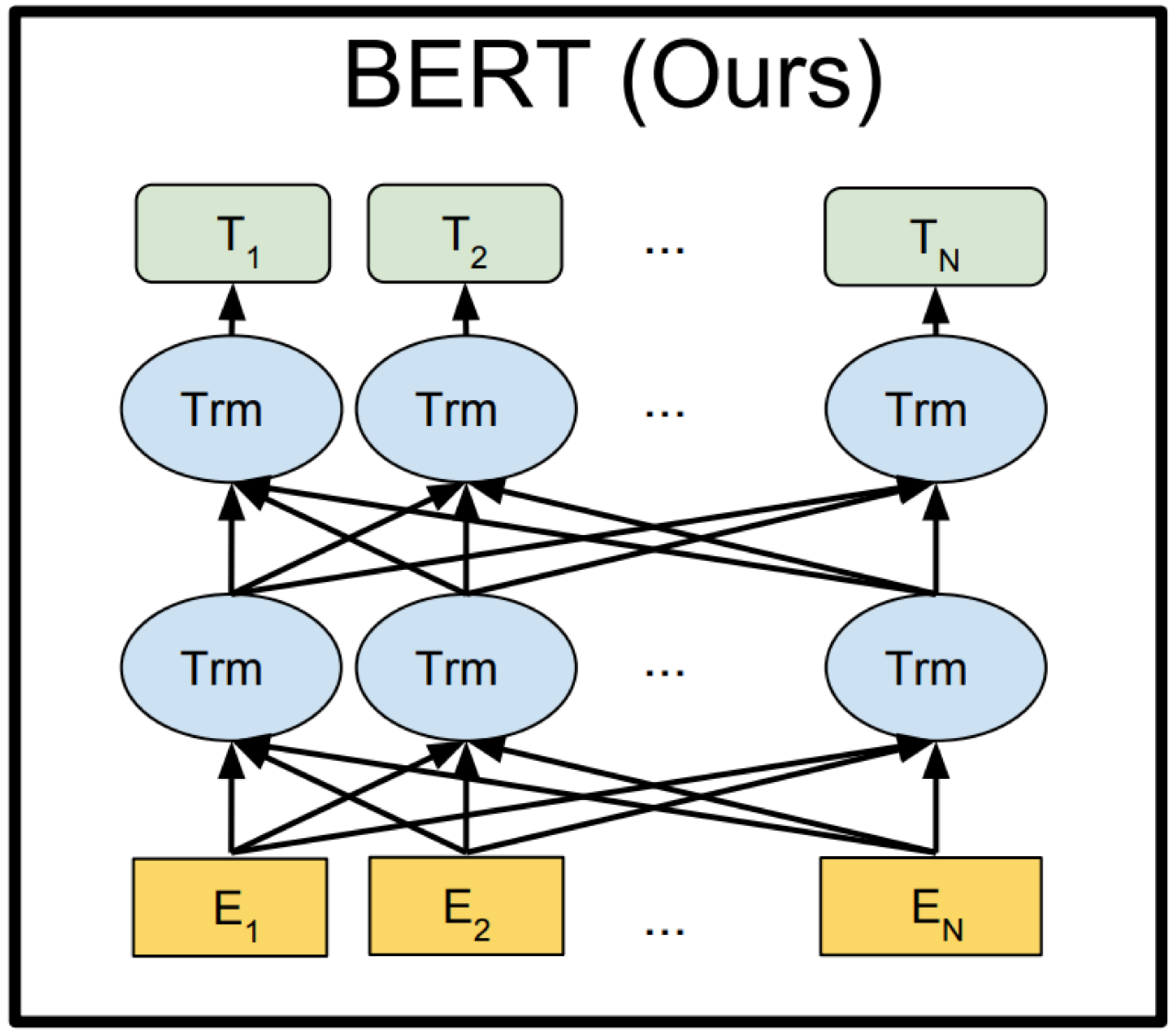


- Encoder and decoder are both transformers
- Decoder consumes the previous generated token (and attends to input), but has *no recurrent state*
- Many other details to get it to work: residual connections, layer normalization, positional encoding, optimizer with learning rate schedule, label smoothing ….

Vaswani et al. (2017)

# Transformer Uses

- 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 (based on LSTM)
- 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
- Carefully choose a decoding strategy
- Up next: Transformers (to finish up)
- Then: pre-trained models