Neural MT + Copy/Pointer

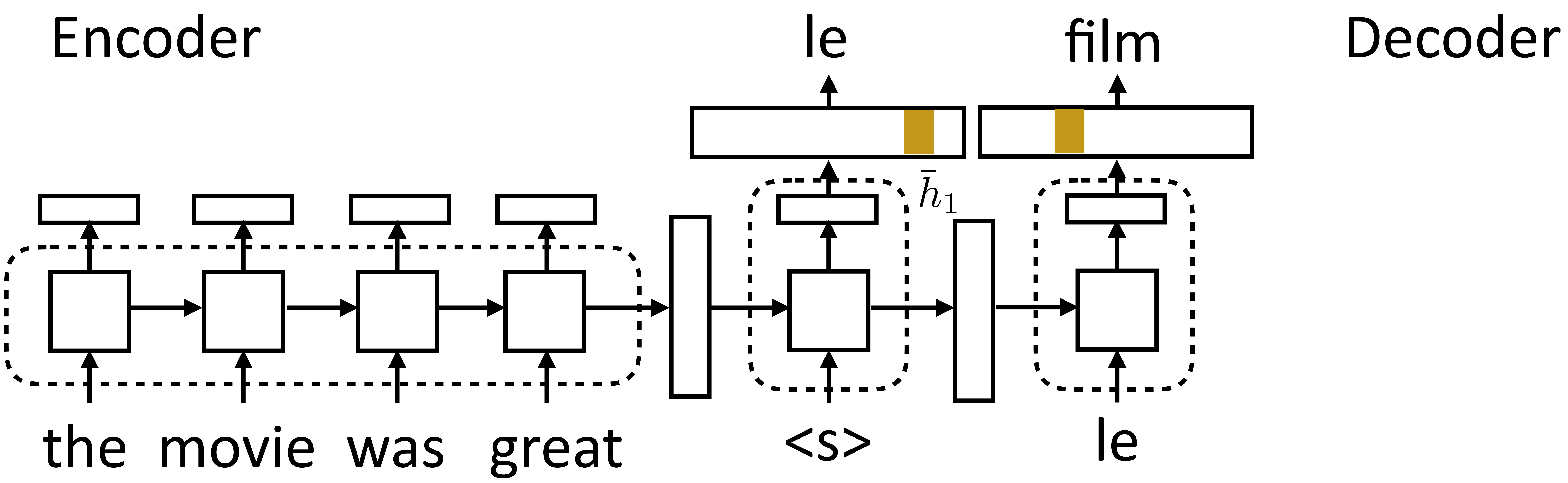
Wei Xu

(many slides from Greg Durrett)

This Lecture

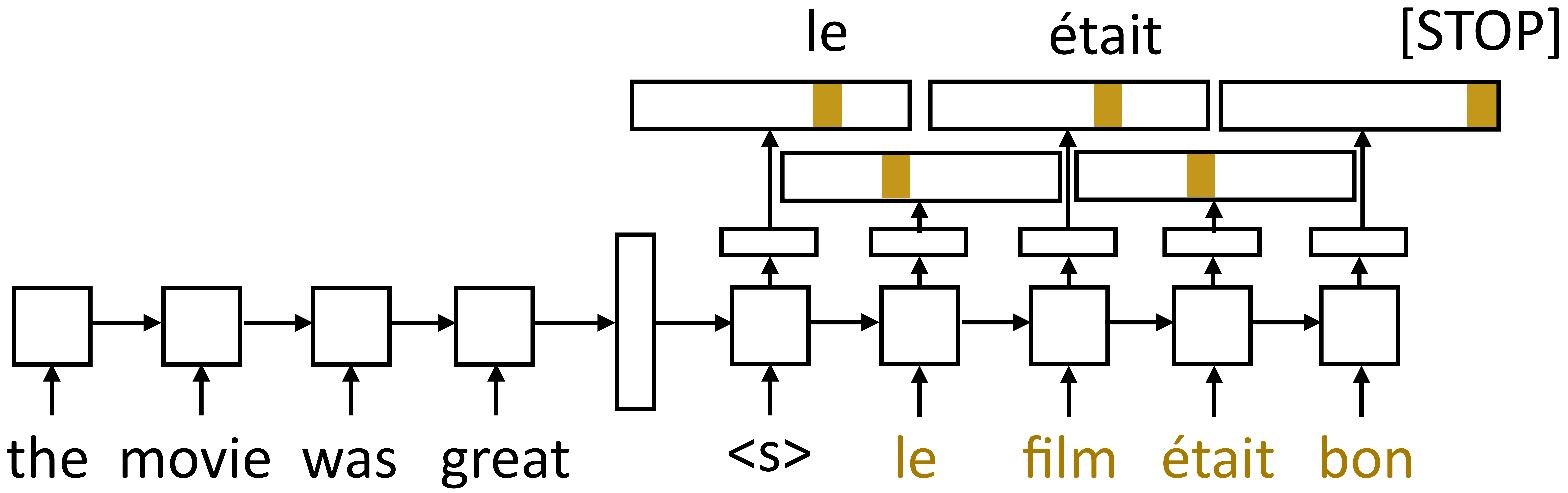
- Sequence-to-Sequence Model
- Attention Mechanism
- **Neural MT and other Applications**
- **Copy/Pointer Network**
- **Transformer Architecture (if time)**

Recap: Seq2Seq Model



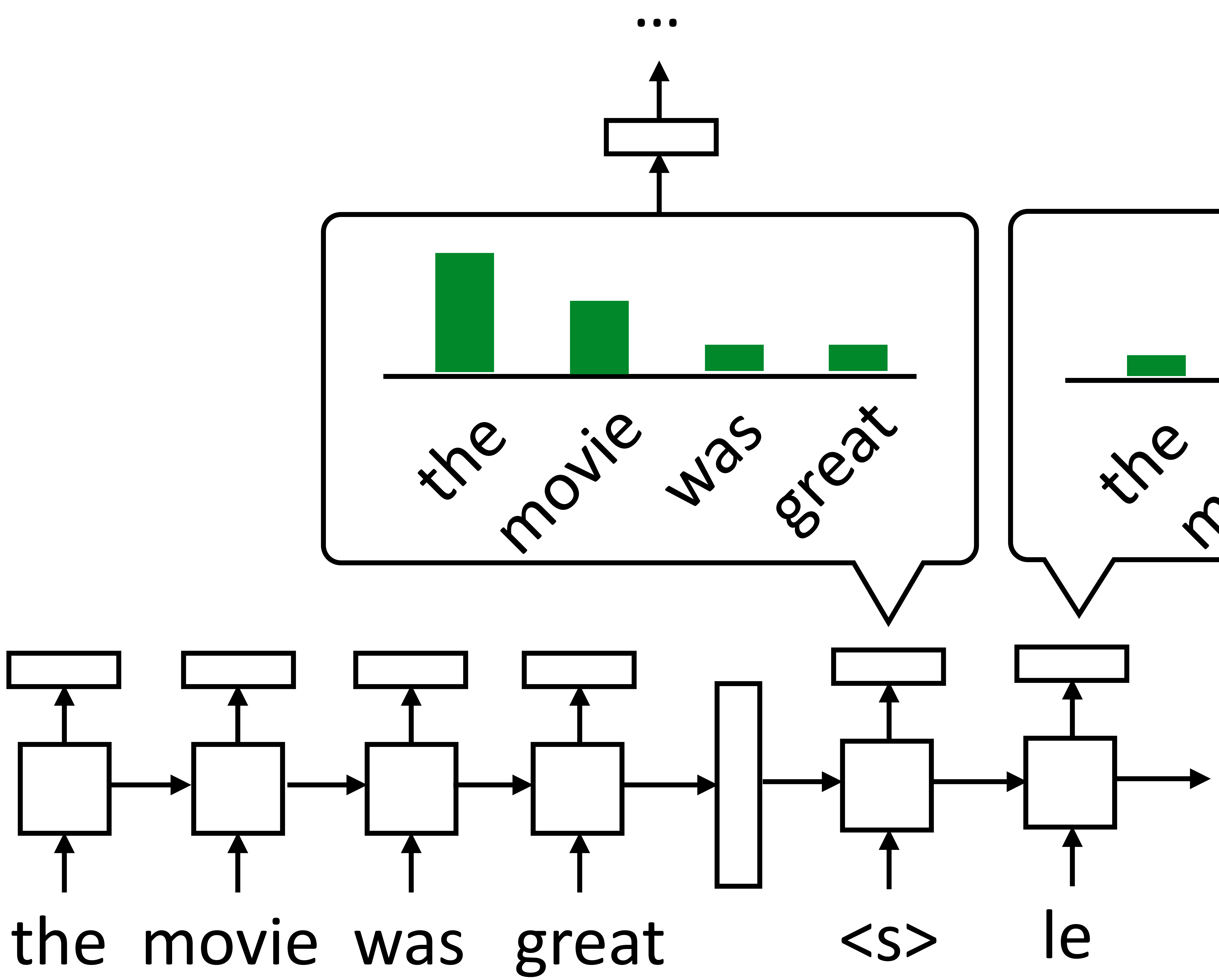
- Encoder: consumes sequence of tokens, produces a vector. Analogous to encoders for classification/tagging tasks $P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W\bar{h}_i)$
- Decoder: separate module, single cell. Takes two inputs: hidden state (vector *h* or tuple (*h*, *c*)) and previous token. Outputs token + new state

Recall: Training Seq2Seq Model



- Objective: maximize $\sum_{(\mathbf{x},\mathbf{y})} \sum_{i=1}^{n} \log P(y_i^* | \mathbf{x}, y_1^*, \ldots, y_{i-1}^*)$
- One loss term for each target-sentence word, feed the correct word regardless of model's prediction

Recall: Attention

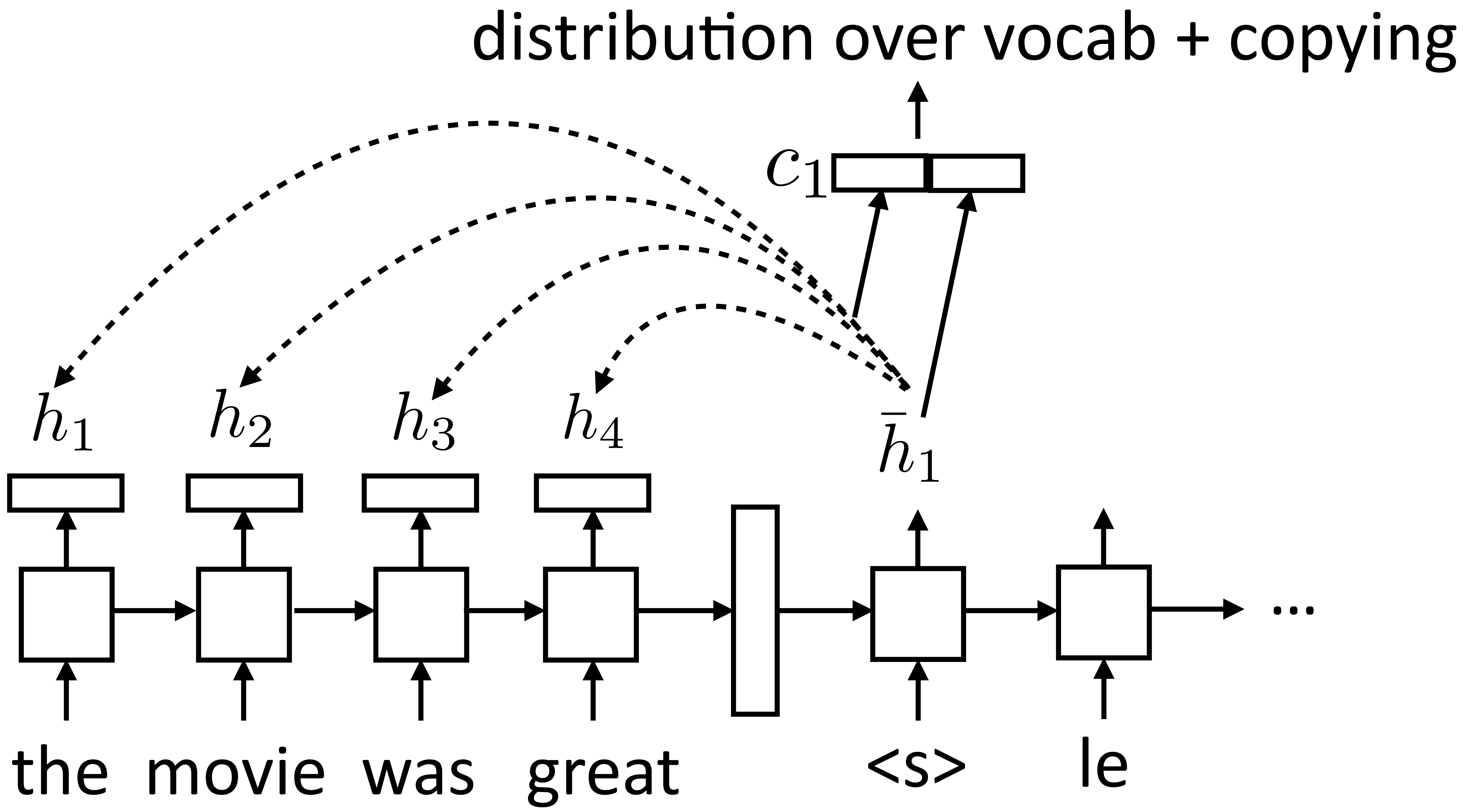


- At each decoder state, compute a distribution over source inputs based on current decoder state
- Use that in output layer

Neural MT

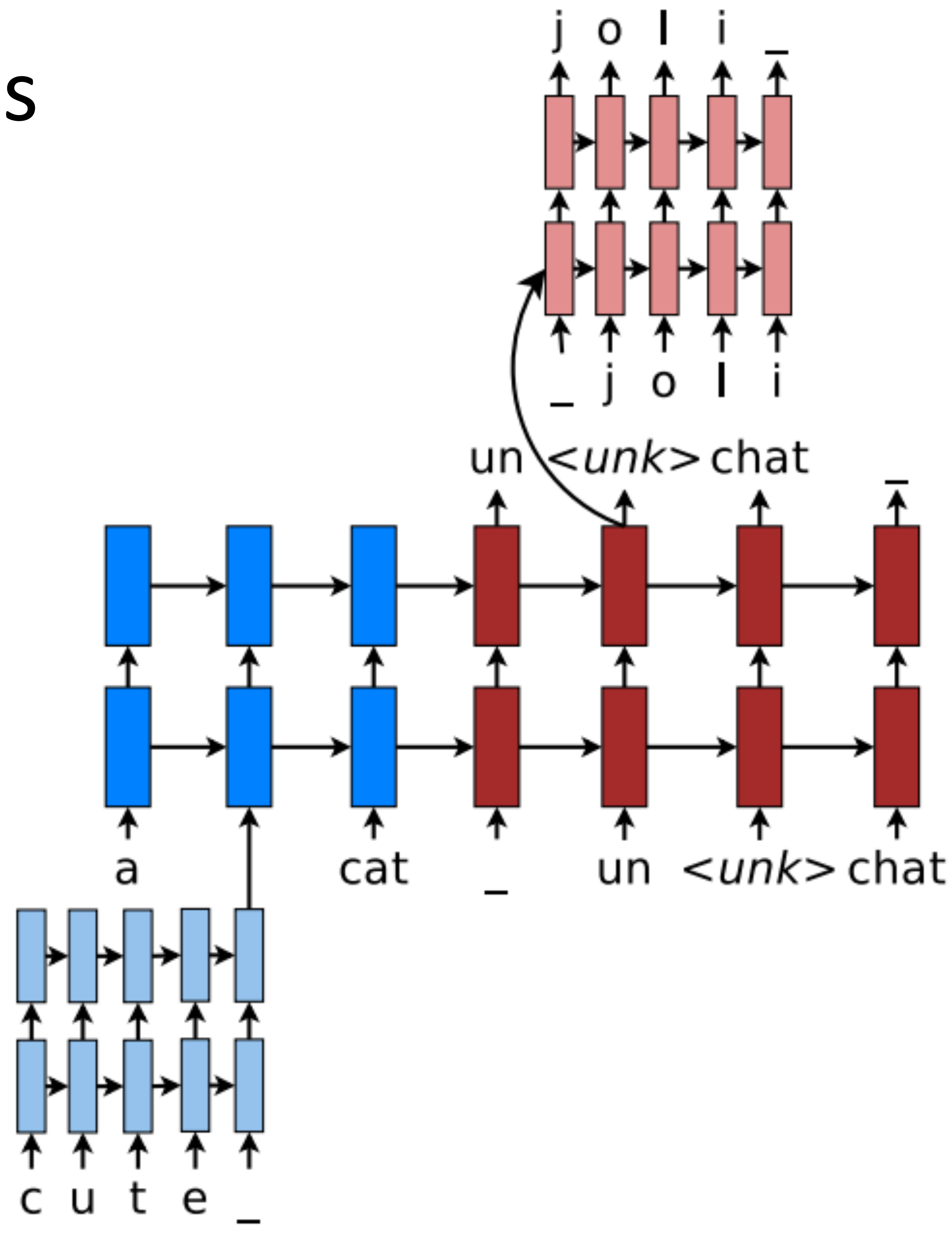
Encoder-Decoder MT

- encoder-decoder with attention and copying for rare words



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
- Models like this in part contributed to dynamic computation graph frameworks becoming popular



Luong et al. (2016)

Handling Rare Words

- Words are a difficult unit to work with: copying can be cumbersome, word vocabularies get very large
- Character-level models don't work well
- Solution: "word pieces" (which may be full words but may be subwords)

Input: *_the* ***_eco tax*** *_port i co _in _Po nt - de - Bu is* ...

Output: *_le _port ique* ***_éco taxe*** *_de _Pont - de - Bui s*

- Can help with transliteration; capture shared linguistic characteristics between languages (e.g., transliteration, shared word root, etc.)

Wu et al. (2016)

Byte Pair Encoding (BPE)

- Start with every individual byte (basically character) as its own symbol

```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences
- Merge the most frequent pair of adjacent characters

- Do this either over your vocabulary (original version) or over a large corpus (more common version)
- Final vocabulary size is often in 10k ~ 30k range for each language
- Most SOTA NMT systems use this on both source + target

Sennrich et al. (2016)

Word Pieces

while voc size < target voc size:

Build a language model over your corpus

Merge pieces that lead to highest improvement in language model perplexity

- SentencePiece library from Google: unigram LM
- Result: way of segmenting input appropriate for translation

Comparison

(a)

Original:	furiously
BPE:	_fur | iously
Unigram LM:	_fur | ious | ly

(b)

Original:	tricycles
BPE:	_t | ric | y | cles
Unigram LM:	_tri | cycle | s

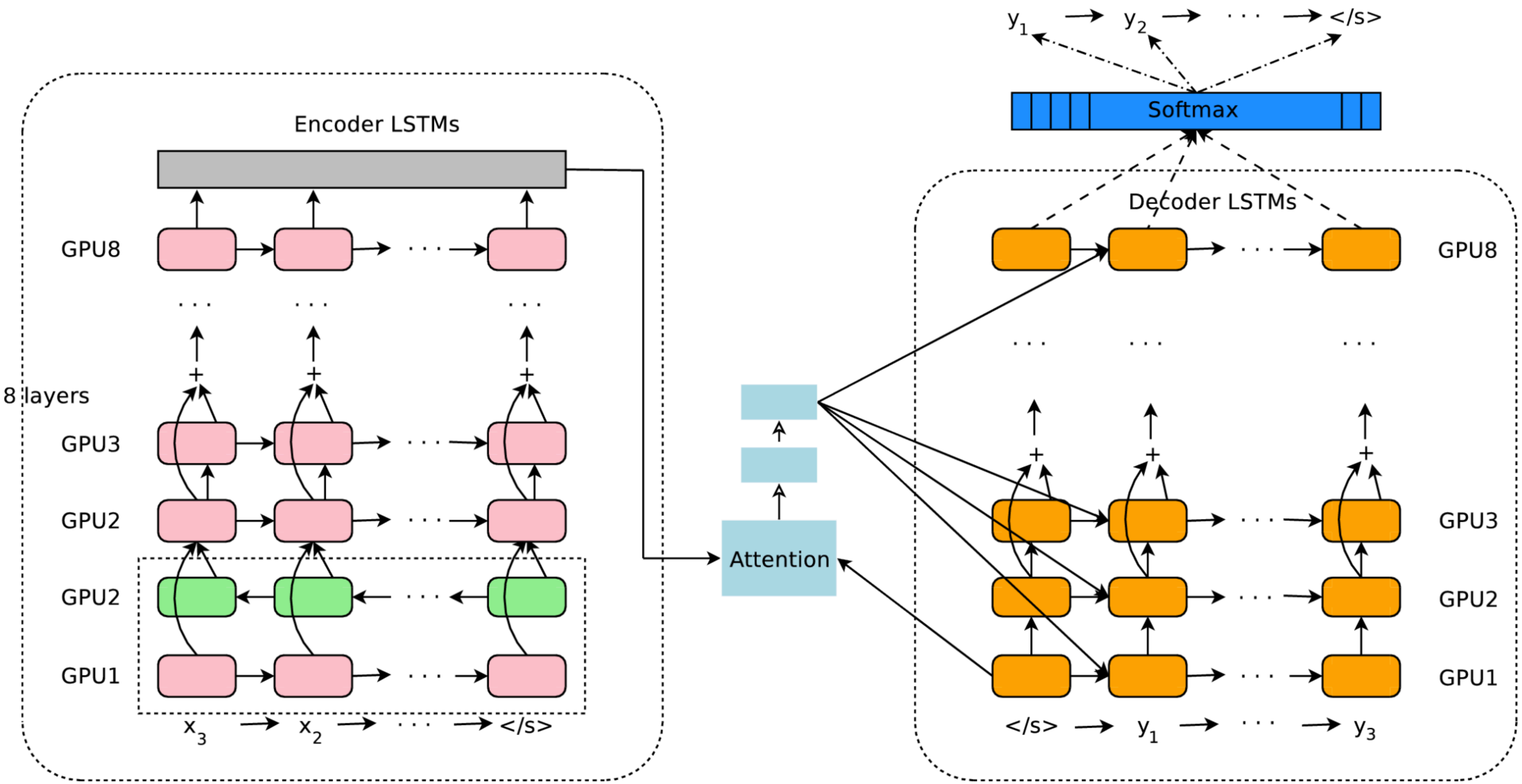
(c)

Original:	Completely preposterous suggestions
BPE:	_Comple | t | ely | _prep | ost | erous | _suggest | ions
Unigram LM:	_Complete | ly | _pre | post | er | ous | _suggestion | s

- BPE produces less linguistically plausible units than word pieces (unigram LM)
- Some evidence that unigram LM works better in pre-trained transformer models

Bostrom and Durrett (2020)

Google’s NMT System



- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k

Wu et al. (2016)

Google’s NMT System

English-French:

Google’s phrase-based system: 37.0 BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: 37.5 BLEU

Google’s 32k word pieces: 38.95 BLEU

English-German:

Google’s phrase-based system: 20.7 BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: 23.0 BLEU

Google’s 32k word pieces: 24.2 BLEU

Wu et al. (2016)

Google's NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT but not in PBMT

"sled"

"walker"

The right-most column shows the human ratings on a scale of 0 (complete nonsense) to 6 (perfect translation)

Wu et al. (2016)

Human Evaluation (En-Es)

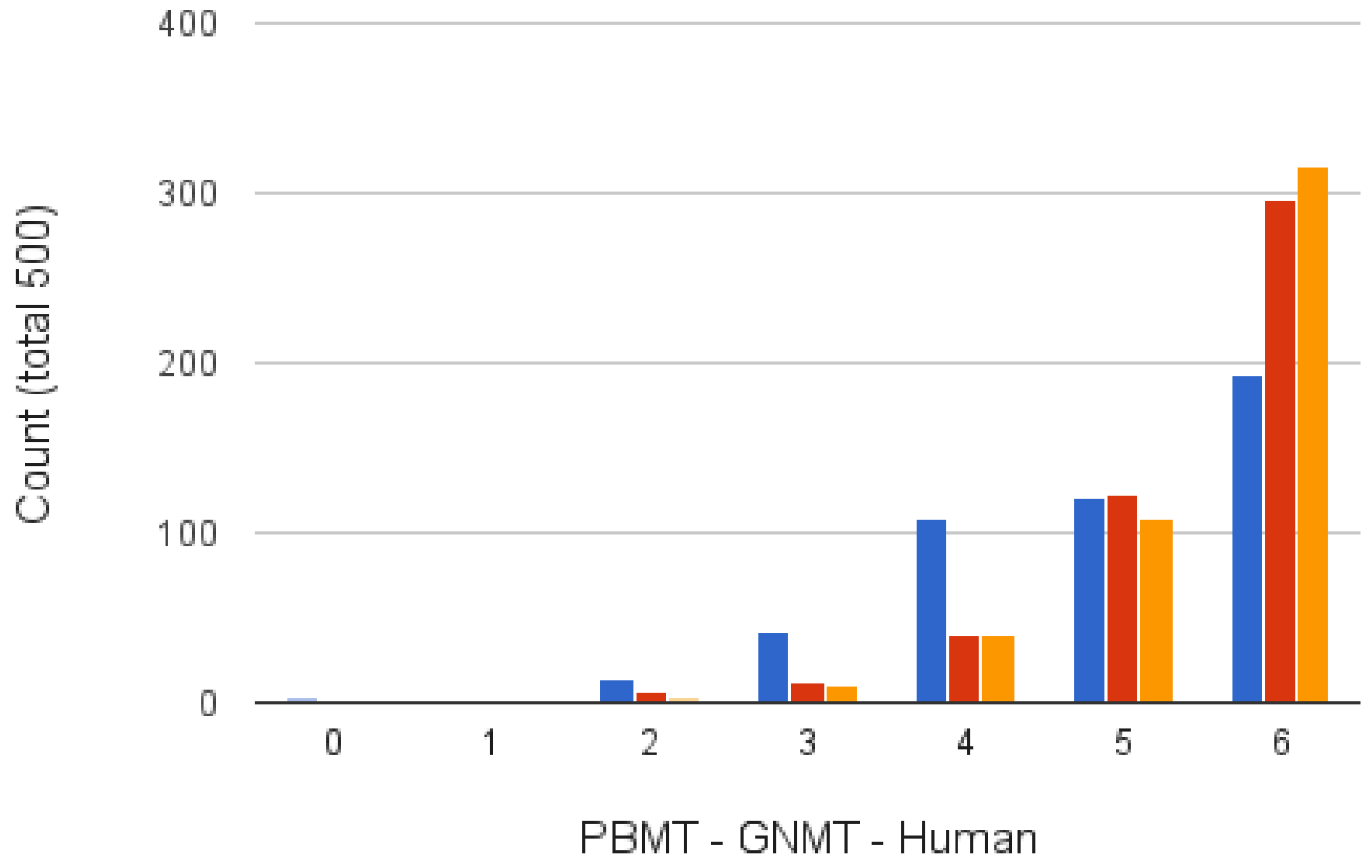


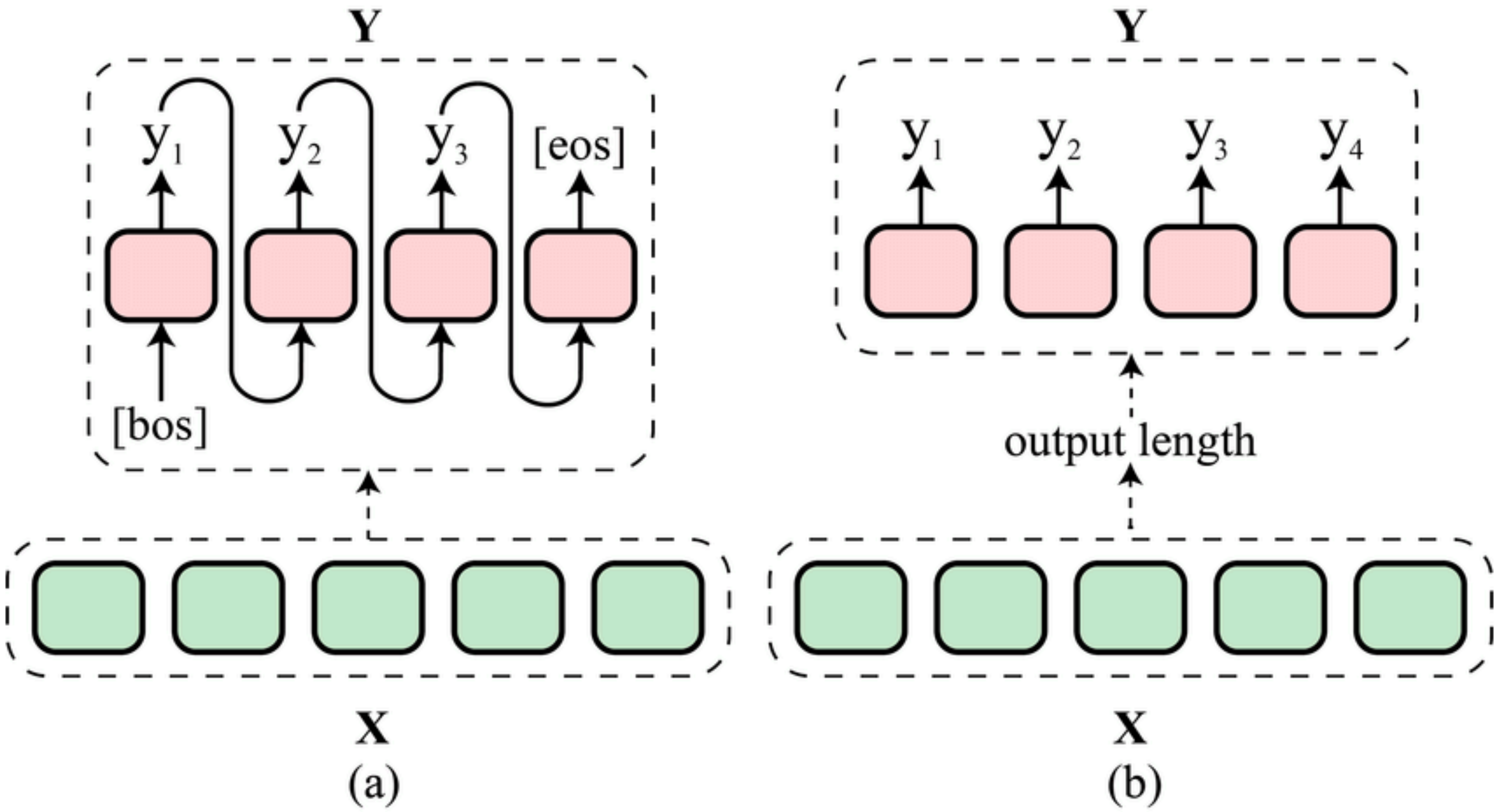
Figure 6: Histogram of side-by-side scores on 500 sampled sentences from Wikipedia and news websites for a typical language pair, here English → Spanish (PBMT blue, GNMT red, Human orange). It can be seen that there is a wide distribution in scores, even for the human translation when rated by other humans, which shows how ambiguous the task is. It is clear that GNMT is much more accurate than PBMT.

- Similar to human-level performance *on English-Spanish*

Wu et al. (2016)

Frontiers in MT

Non-Autoregressive NMT



- Q: why non-autoregressive? Pros and cons?

Gu et al. (2018), Ghazvininejad et al. (2019), Kasai et al. (2020)

Low-Resource MT

- Particular interest in deploying MT systems for languages with little or no parallel data

Burmese, Indonesian, Turkish

Transfer	BLEU My→En	Id→En	Tr→En
baseline (no transfer)	4.0	20.6	19.0
transfer, train	17.8	27.4	20.3
transfer, train, reset emb, train	13.3	25.0	20.0
transfer, train, reset inner, train	3.6	18.0	19.1

Table 3: Investigating the model's capability to restore its quality if we reset the parameters. We use En→De as the parent.

- BPE allows us to transfer models even without training on a specific language

- Pre-trained models can help further

Aji et al. (2020)

Unsupervised MT

Approach	Train/Val	Test	Loss
Supervised MT	L1-L2	L1-L2	$\mathcal{L}^{MT}_{x\to y} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim(\mathcal{X},\mathcal{Y})}\left[-\log p_{x\to y}(\mathbf{y}\|\mathbf{x})\right]$
Unsupervised MT	L1, L2	L1-L2	$\mathcal{L}^{BT}_{x\leftrightarrow y} = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[-\log p_{y\to x}(\mathbf{x}\|g^*(\mathbf{x}))\right] + \mathbb{E}_{\mathbf{y}\sim\mathcal{Y}}\left[-\log p_{x\to y}(\mathbf{y}\|h^*(\mathbf{y}))\right]$ g^*, h^*: sentence predictors

- Common principles of unsupervised MT
  - Language models
  - (Iterative) Back-translation!

Lample et al. (2018)

Copying Input/Pointers

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

- Problem: 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

- Vocabulary contains "normal" vocab as well as words in input. Normalizes over both of these:

$$P(y_i = w|\mathbf{x}, y_1, \ldots, y_{i-1}) \propto \begin{cases} \exp W_w[c_i; \bar{h}_i] & \text{if } w \text{ in vocab} \\ \exp h_j^\top V \bar{h}_i & \text{if } w = x_j \end{cases}$$

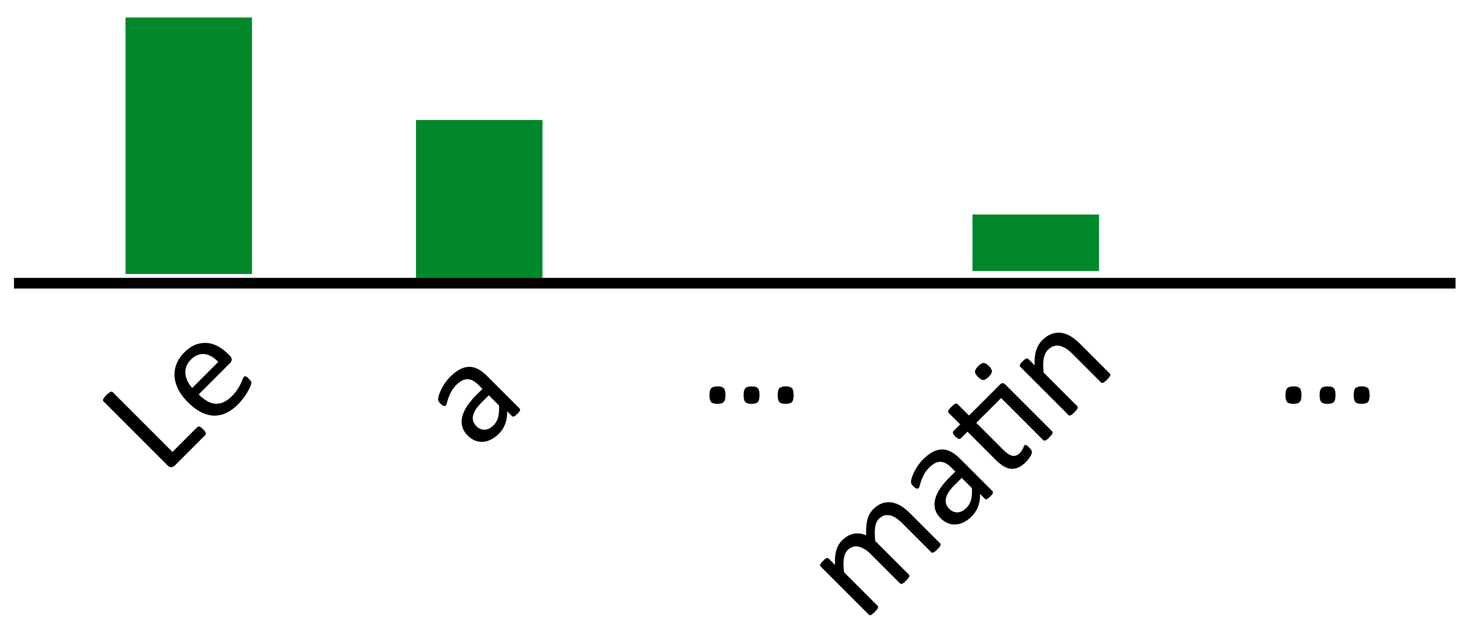
- Bilinear function of input representation + output hidden state

Gu et al. (2016)

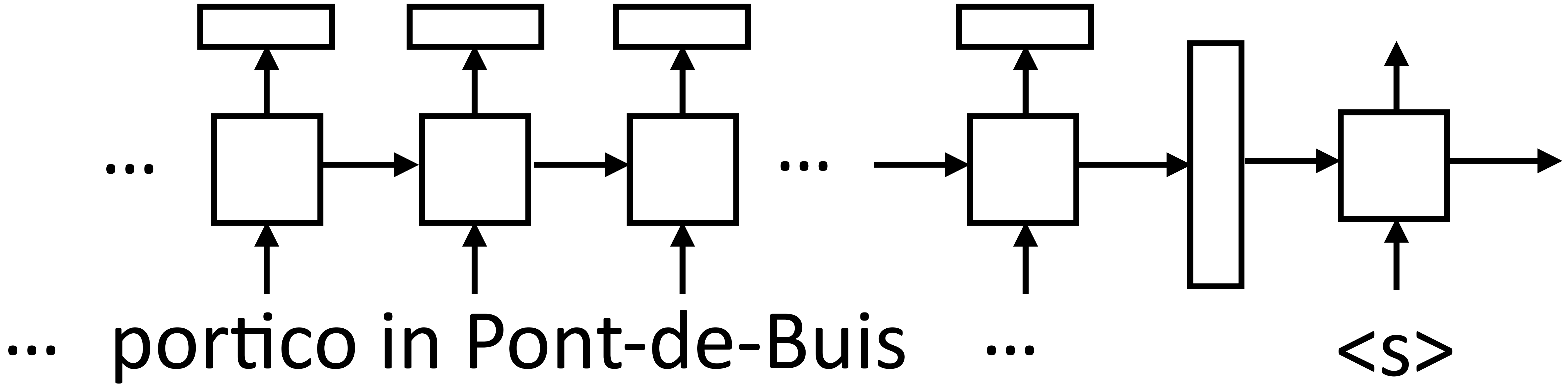
Pointer Network

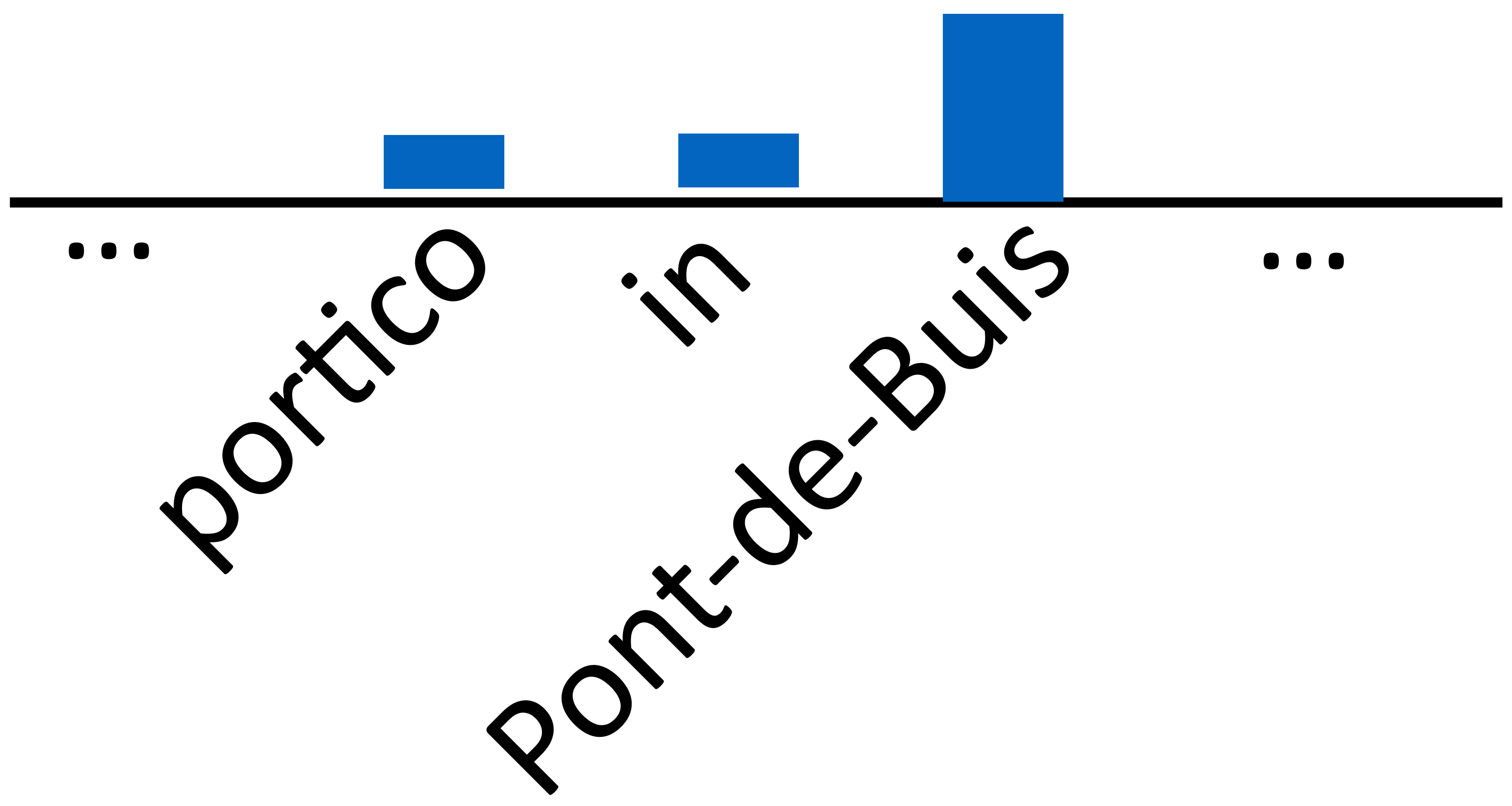
- Standard decoder (P_{vocab}): softmax over vocabulary

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



- Pointer network (P_{pointer}): predict from *source* words, instead of target vocabulary $P_{\text{pointer}}(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) \propto \begin{cases} h_j^\top V \bar{h}_i \text{ if } y_i = w_j \\ 0 \text{ otherwise} \end{cases}$





Pointer Generator Mixture Models

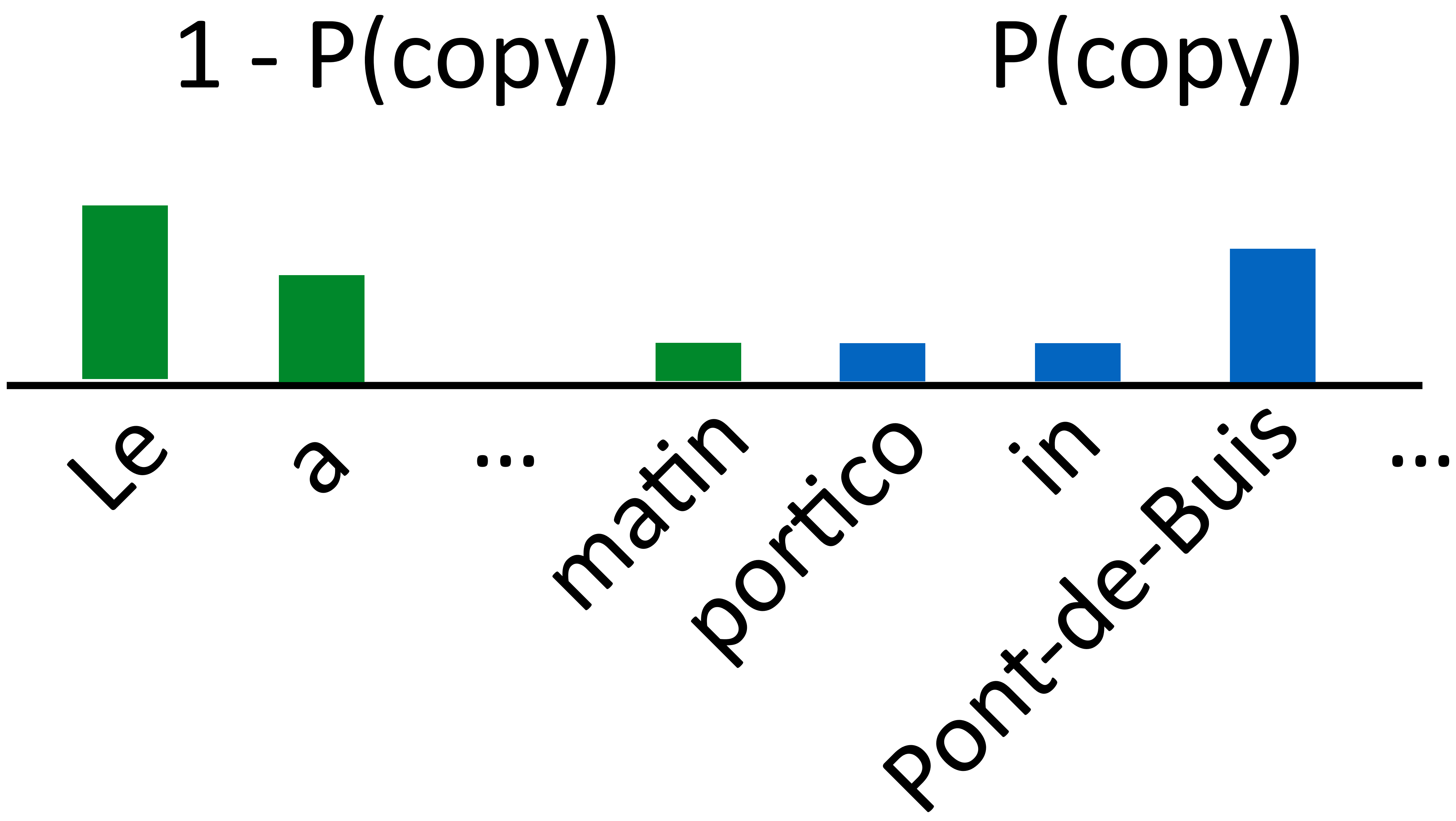
- Define the decoder model as a mixture model of P_{vocab} and P_{pointer}

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

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

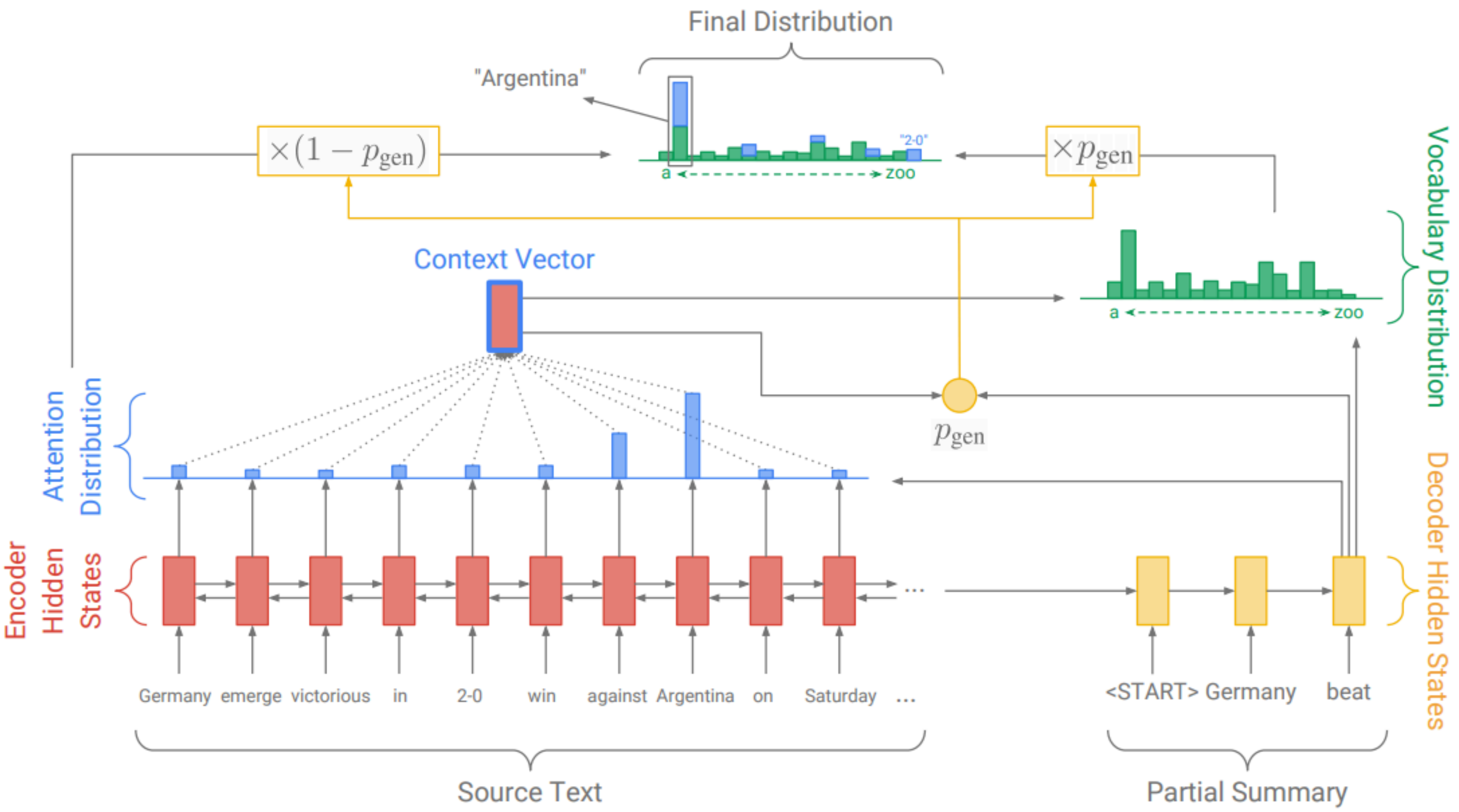
- Marginalize over copy variable during training and inference

- 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.

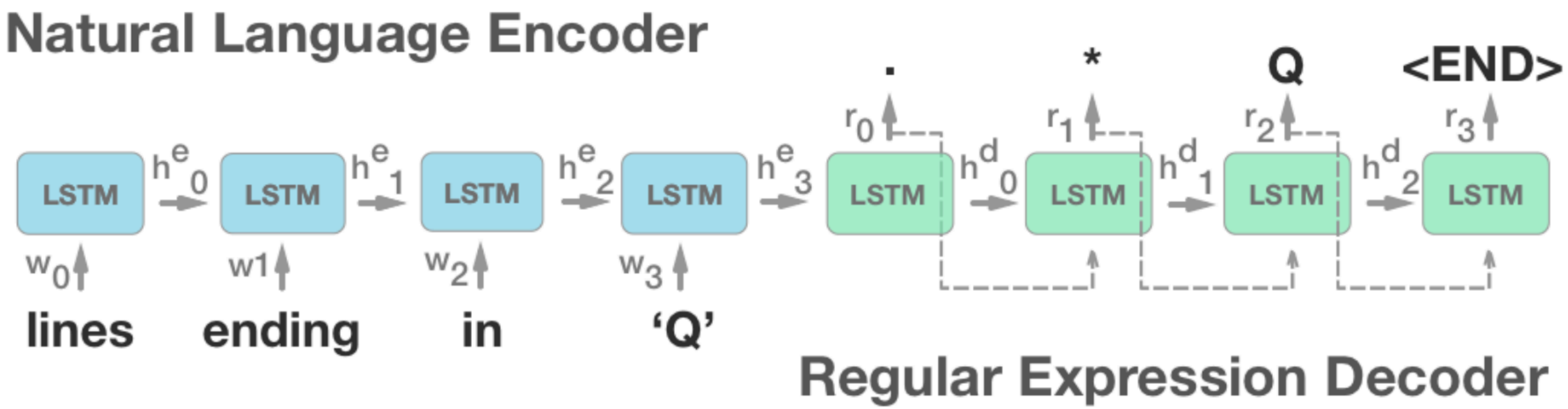
Figure 1: Comparison of output of 3 abstractive summarization models on a news article. The baseline model makes **factual errors**, a **nonsensical sentence** and struggles with OOV words *muhammadu buhari*. The pointer-generator model is accurate but **repeats itself**. Coverage eliminates repetition.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

See et al. (2017)

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) $\wedge$ 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

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

Jia and Liang (2015)

SQL Generation

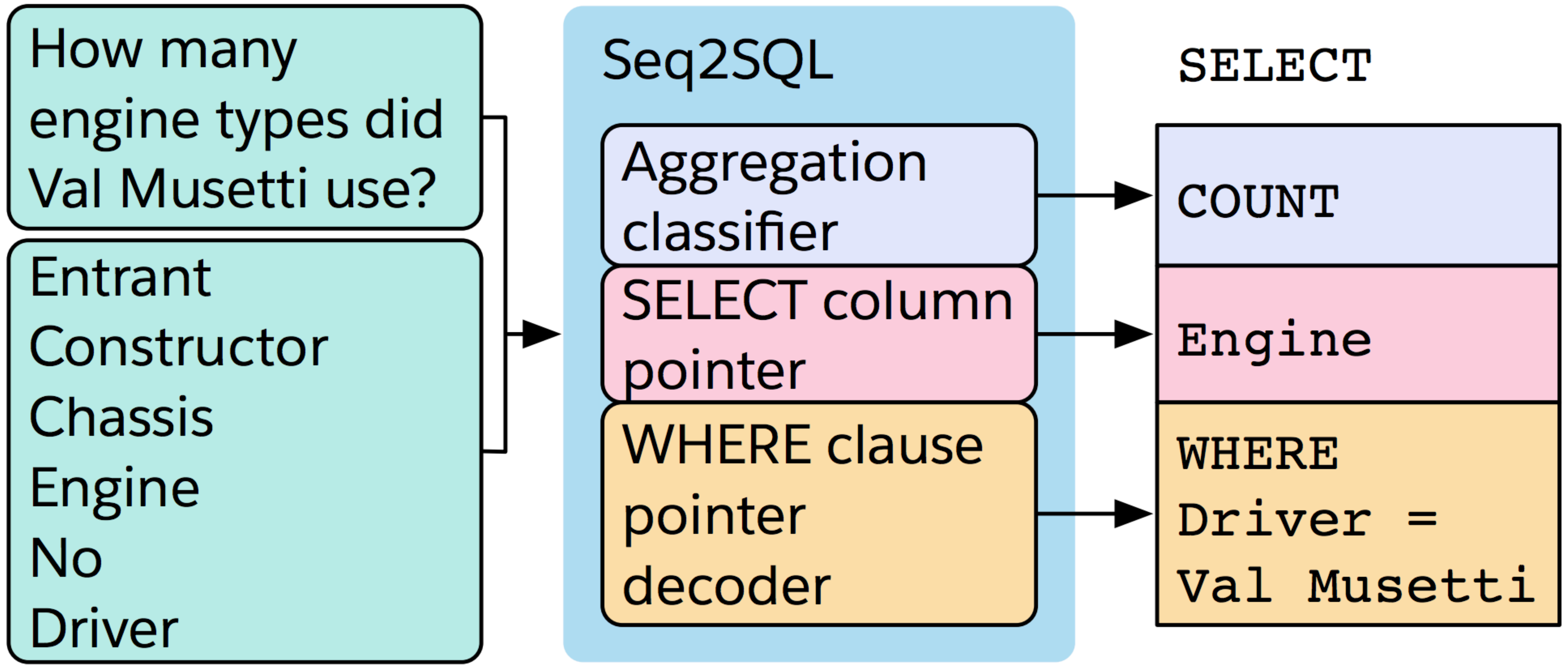
- Convert natural language description into a SQL query against some DB
- How to ensure that well-formed SQL is generated?
  - Three components
- How to capture column names + constants?
  - Pointer mechanisms

Question:

How many CFL teams are from York College?

SQL:

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



Zhong et al. (2017)

Text Simplification

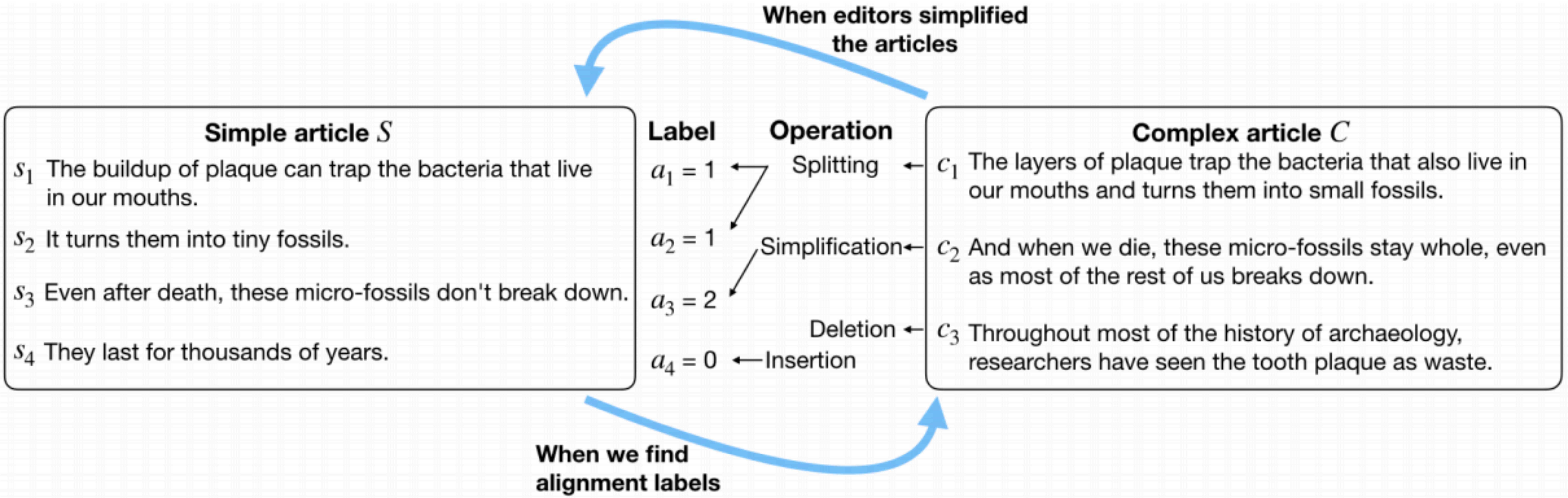


Figure 1: An example of sentence alignment between an original news article (right) and its simplified version (left) in Newsela. The label a_i for each simple sentence s_i is the index of complex sentence c_{a_i} it aligned to.

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))												
$\text{Transformer}_{rand}$	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
$\text{Transformer}_{bert}$	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)												
$\text{Transformer}_{rand}$	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
$\text{Transformer}_{bert}$	**36.6**	**4.5**	31.0	74.3	**6.8**	**13.3**	**36.8**	**3.8**	33.1	73.4	6.8	**13.5**

94k sent. pairs

394k sent. pairs

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.

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

Takeaways

- Can build MT systems with LSTM encoder-decoders, CNNs, or transformers
- Word piece / byte pair models are really effective and easy to use
- State of the art systems are getting pretty good, but lots of challenges remain, especially for low-resource settings
- Next class: Transformer, a very strong model (when data is large enough); training can be tricky