Attention + Neural MT

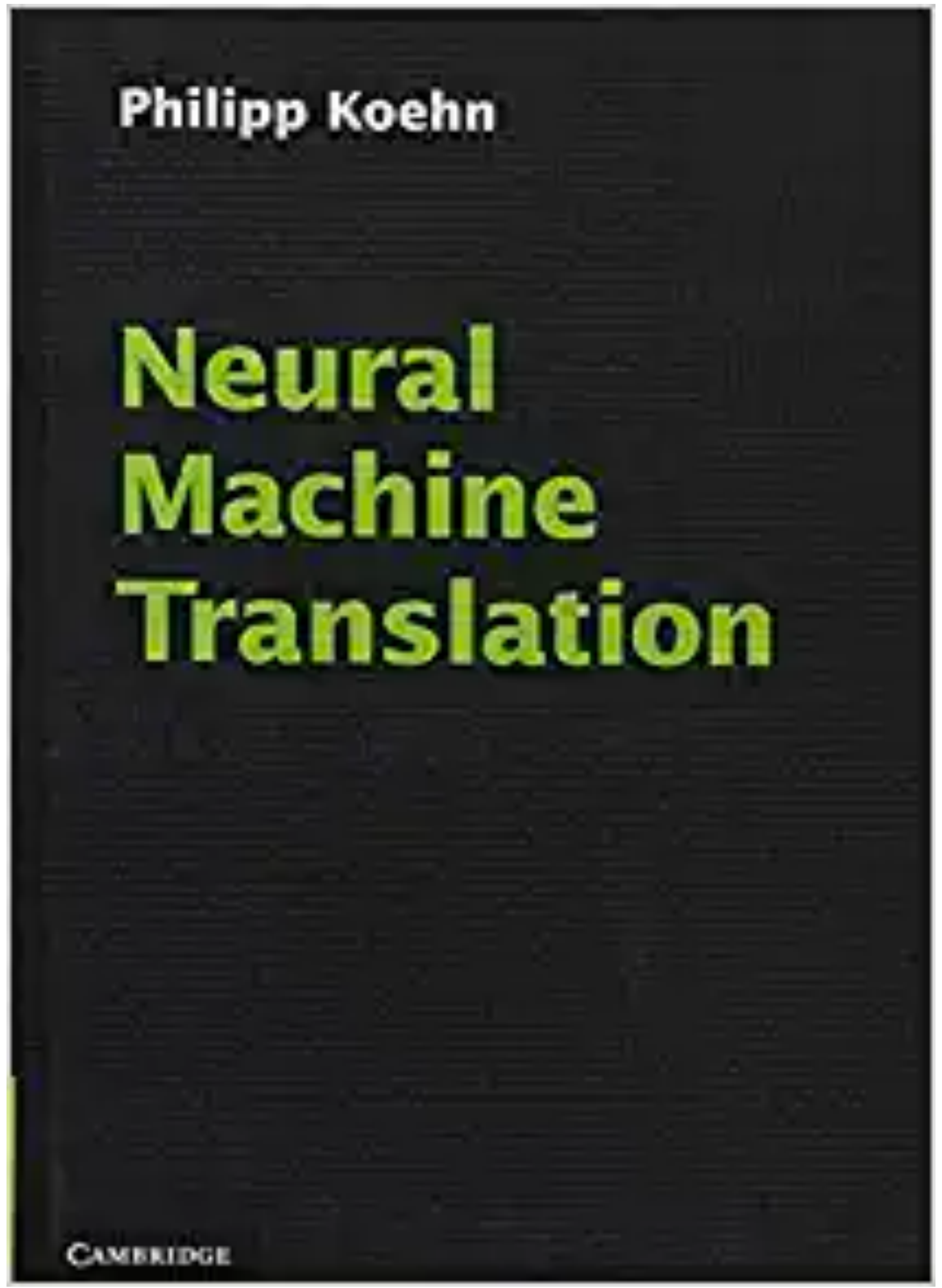
Wei Xu

(many slides from Greg Durrett)

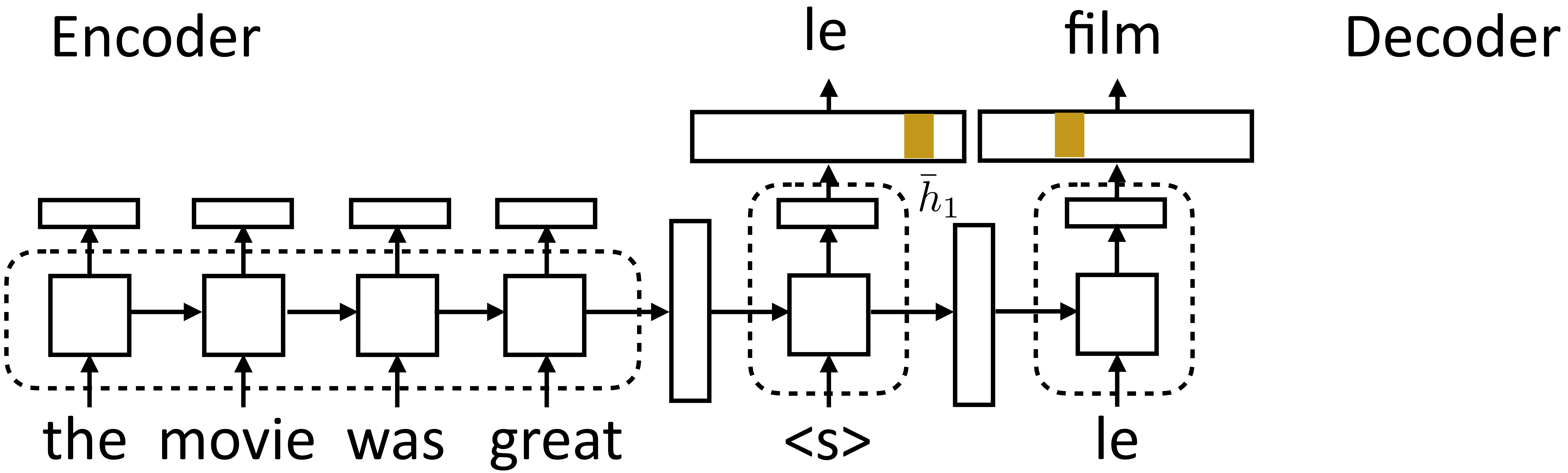
This Lecture

- Attention
- Copy/Pointer Network
- Neural Machine Translation

- Reading — Eisenstein 18.3-18.5



Recap: Seq2Seq Model



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

Results: Encoder-Decoder MT

- Kalchbrenner & blunsom (2013), Bahanau et al. (2014), Cho et al. (2014)
- Sutskever et al. (2014) paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

- SOTA = 37.0 then — not all that competitive…

Sutskever et al. (2014)

Attention

Problems with Seq2seq Models

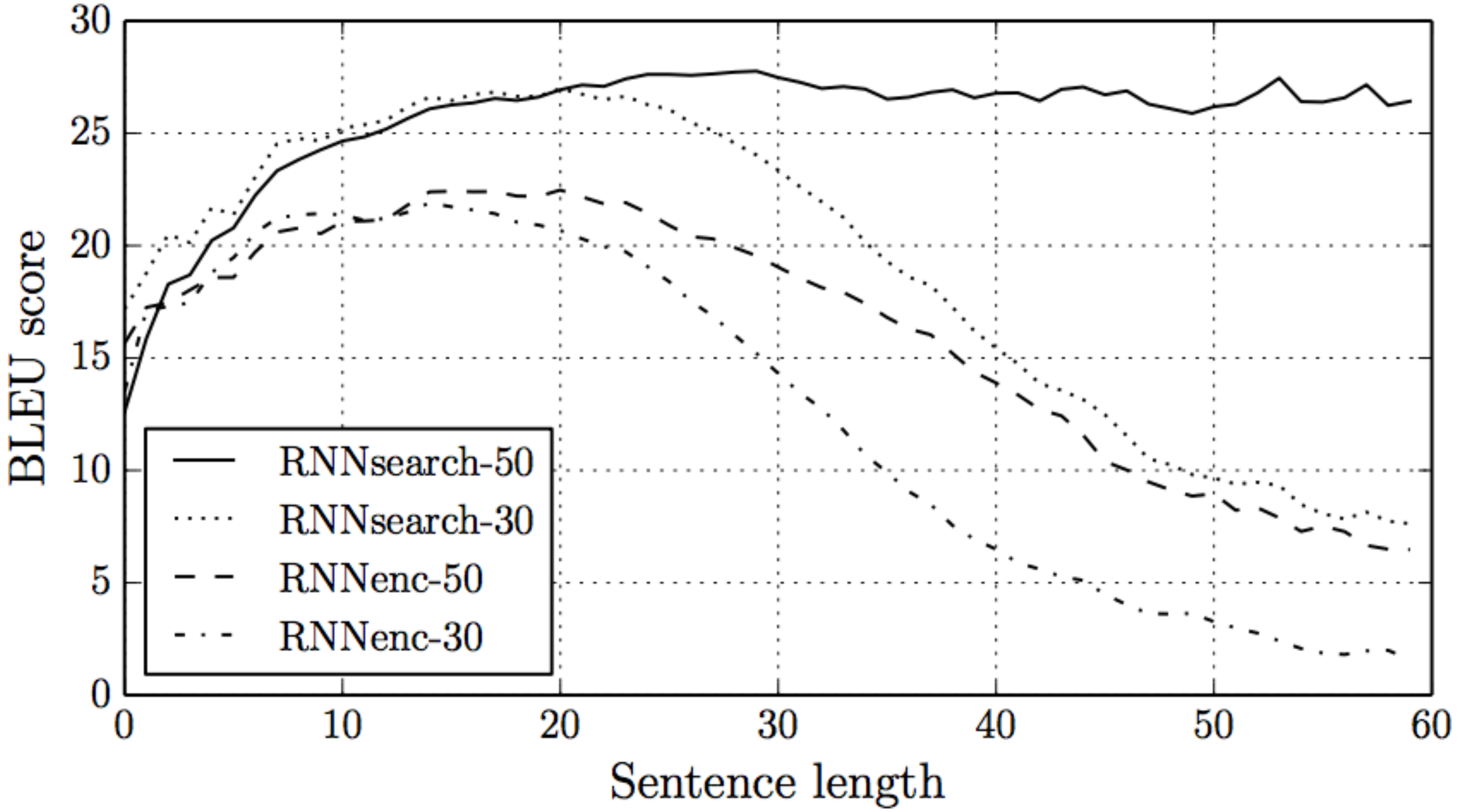
- Encoder-decoder models like to repeat themselves:

Un garçon joue dans la neige → *A boy plays in the snow* ***boy plays boy plays***

- Often a byproduct of training these models poorly. Input is forgotten by the LSTM so it gets stuck in a "loop" of generation the same output tokens again and again.
- Need some notion of input coverage or what input words we've translated

Problems with Seq2seq Models

- Bad at long sentences: 1) a fixed-size hidden representation doesn't scale; 2) LSTMs still have a hard time remembering for really long sentences



RNNenc: the model we've discussed so far

RNNsearch: uses attention

Bahdanau et al. (2014)

Problems with Seq2seq Models

- 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

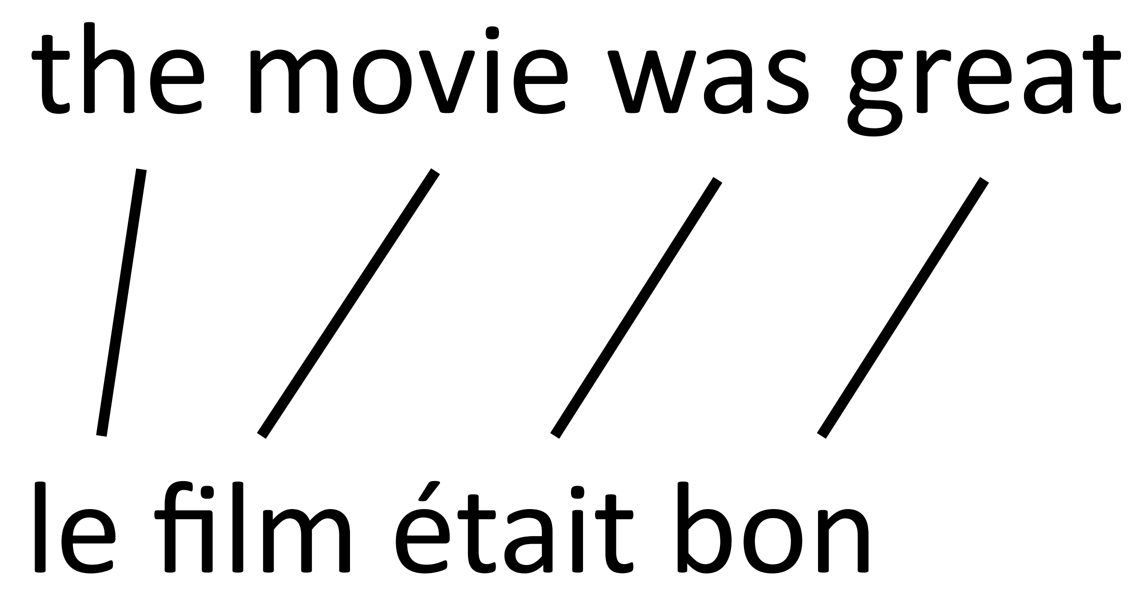
nn: Le `unk` de `unk` à `unk` , . . . [truncated] . . . , a été pris le jeudi matin

- Encoding these rare words into a vector space is really hard
- In fact, we don't want to encode them, we want a way of directly looking back at the input and copying them (Pont-de-Buis)

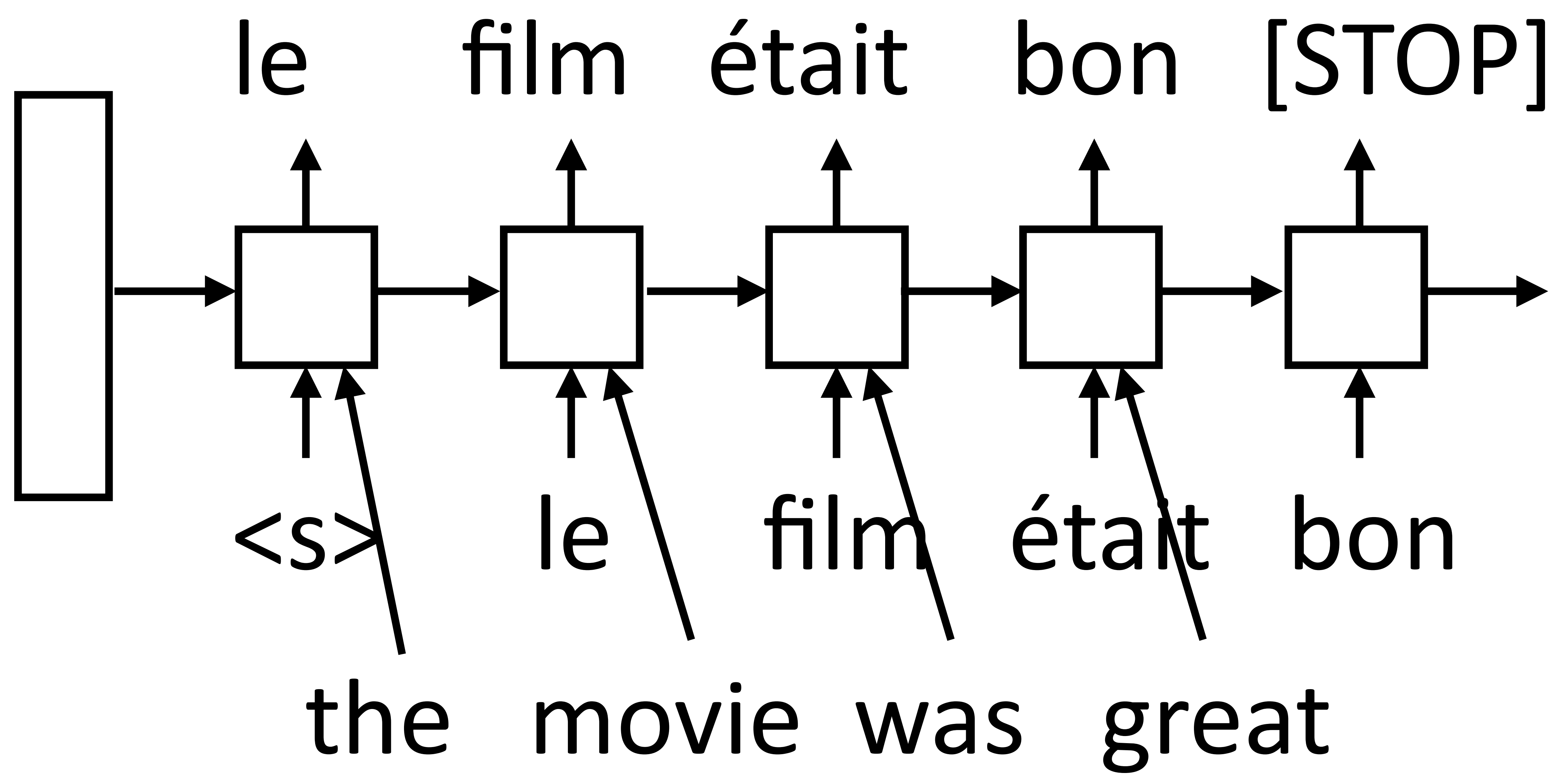
Jean et al. (2015), Luong et al. (2015)

Aligned Inputs

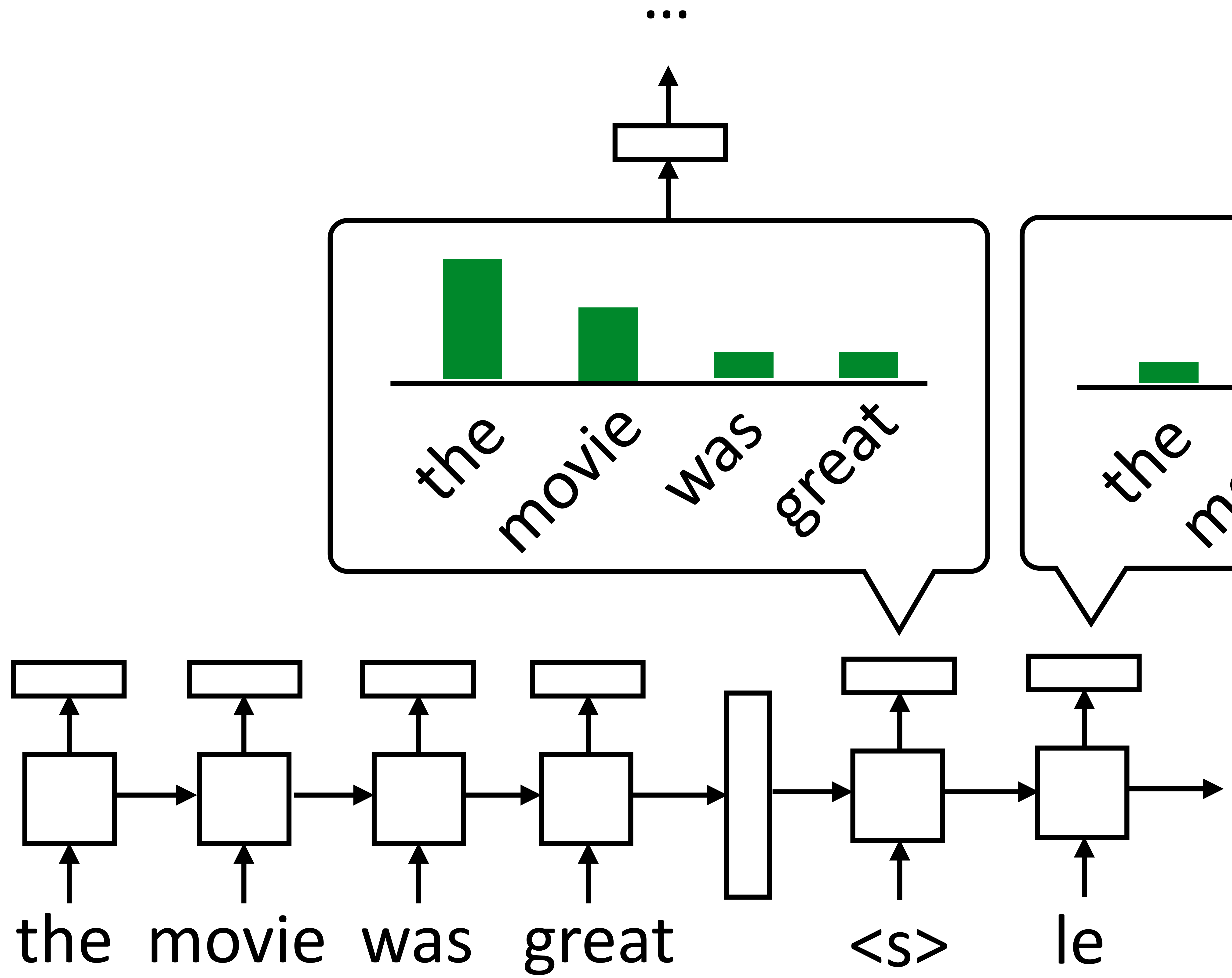
- Suppose we knew the source and target would be word-by-word translated (recall the word alignment we talked about in phrase-based MT)



- Can look at the corresponding input word when translating — this could scale!
- Less burden on the hidden states
- How can we achieve this without hardcoding it?



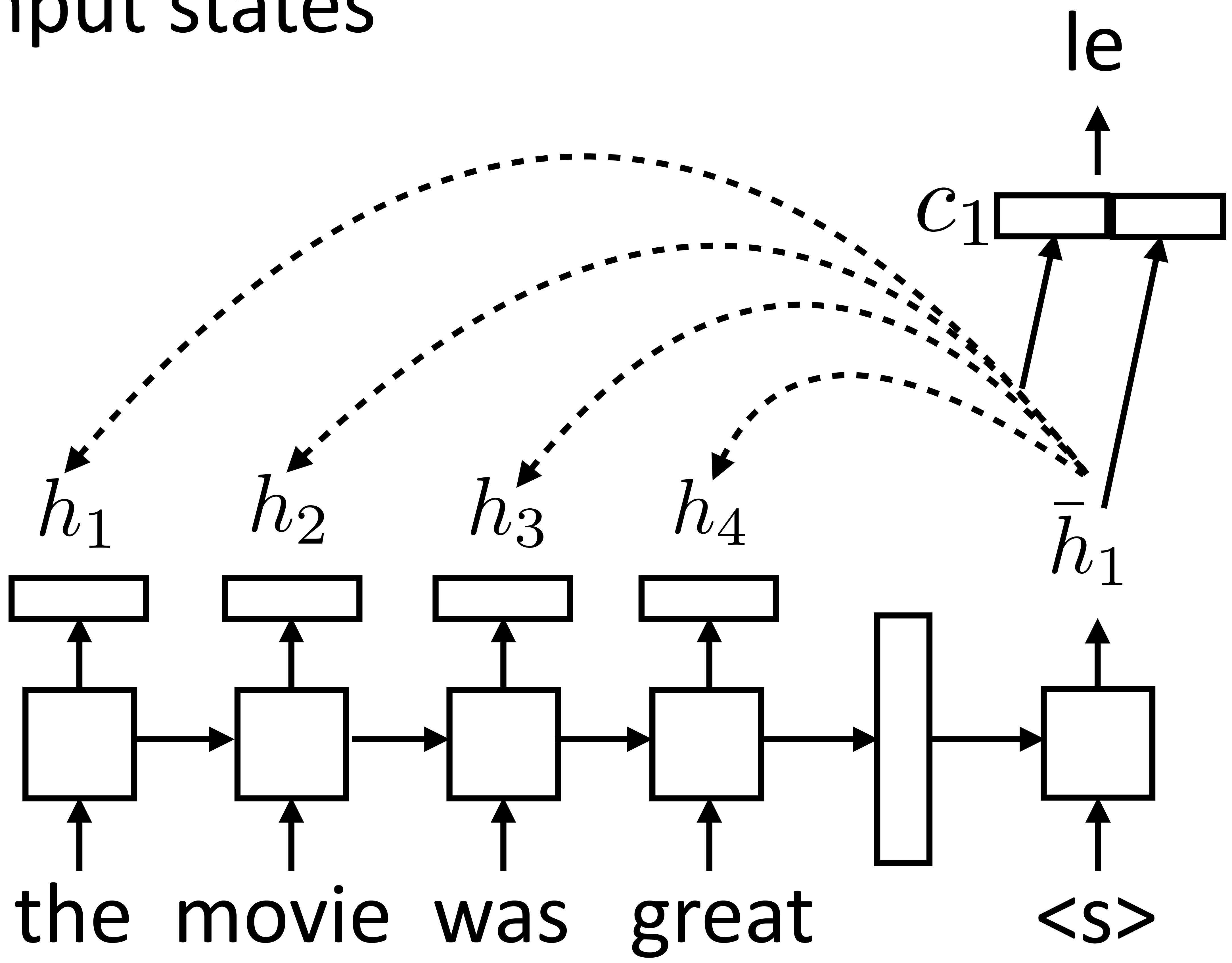
Attention



- At each decoder state, compute a distribution over source inputs based on current decoder state
- Use the weighted sum of input tokens to predict output

Attention

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



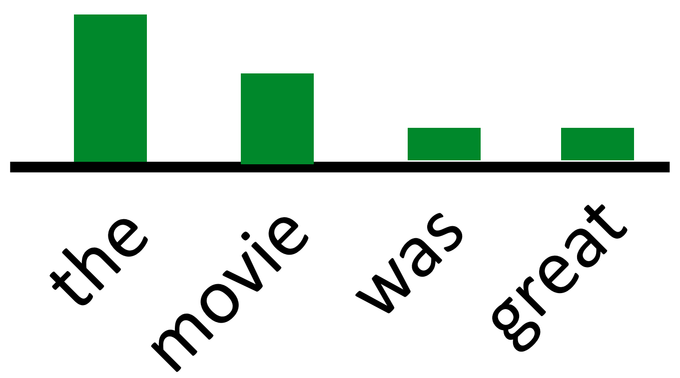
- No attn: $P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \mathrm{softmax}(W\bar{h}_i)$

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

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

- Weighted sum of input hidden states (vector)

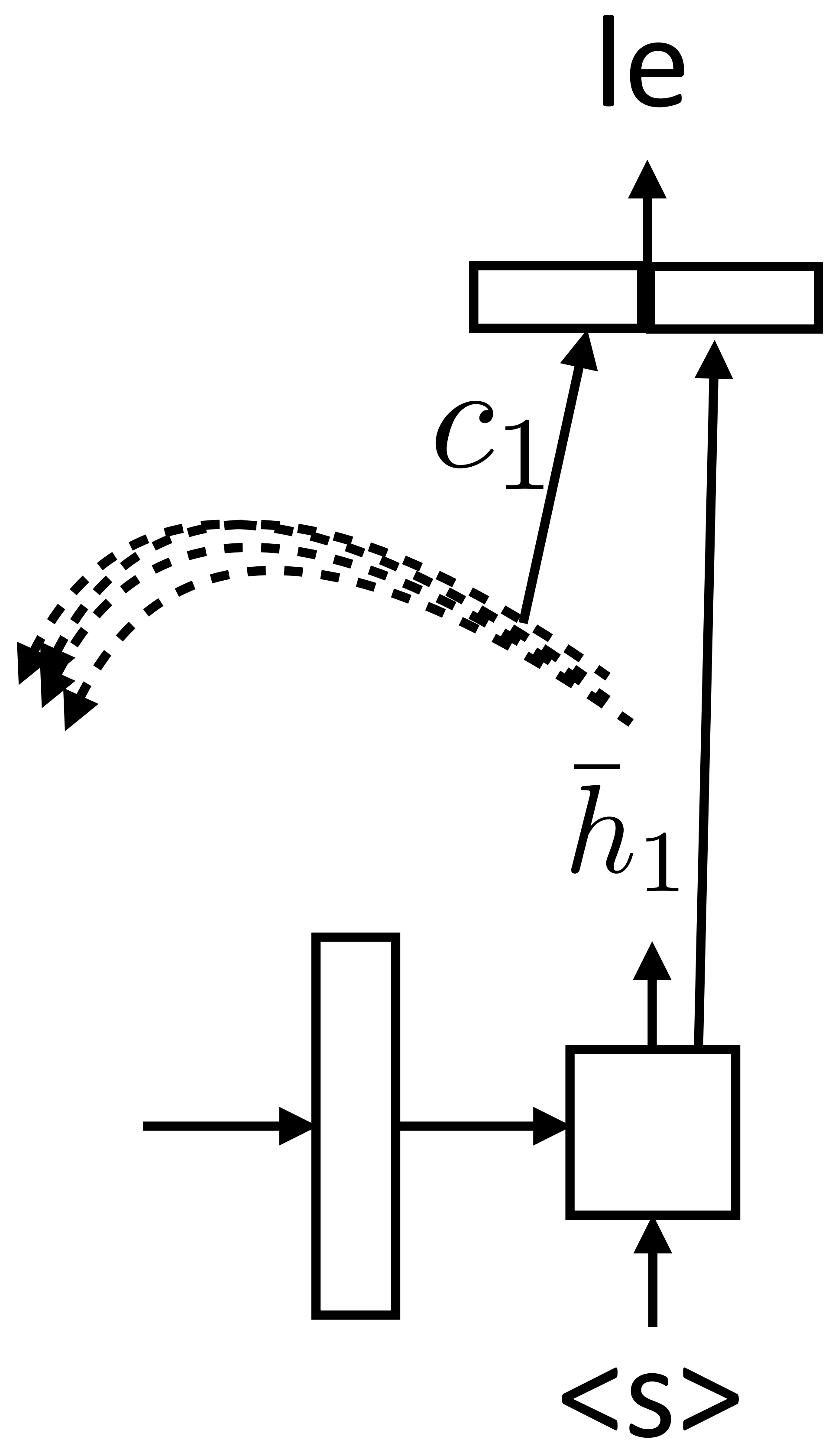
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$



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

- Some function f (e.g., dot product)

Attention



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

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

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

$$f(\bar{h}_i, h_j) = \tanh(W[\bar{h}_i, h_j])$$

- Bahdanau+ (2014): additive

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

- Luong+ (2015): dot product

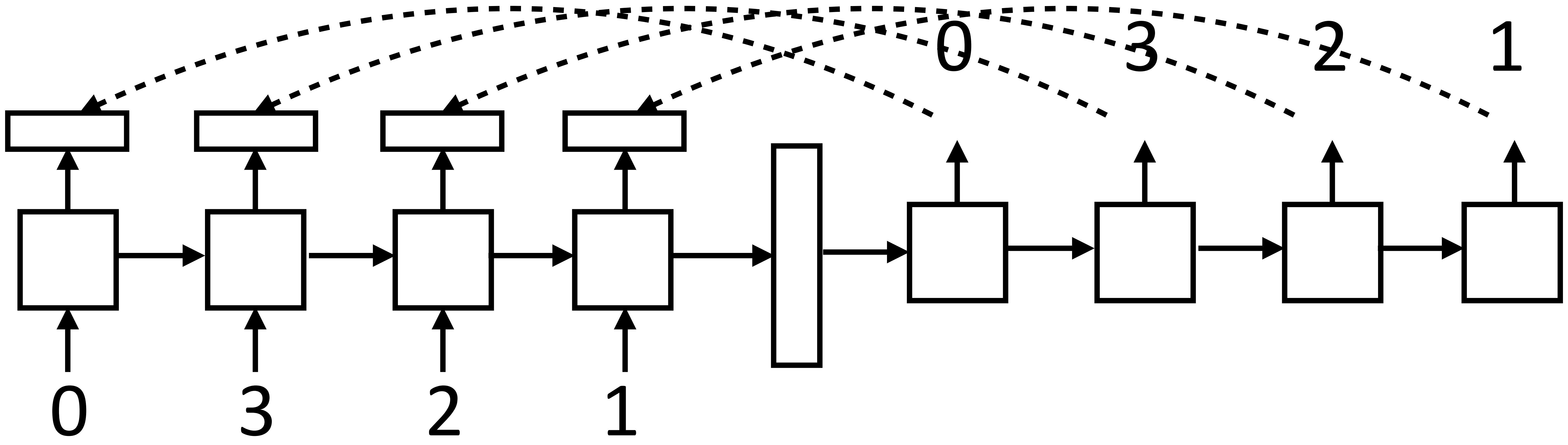
$$f(\bar{h}_i, h_j) = \bar{h}_i^\top W h_j$$

- Luong+ (2015): bilinear

- Note that this all uses outputs of hidden layers

What can attention do?

- Learning to copy — how might this work?

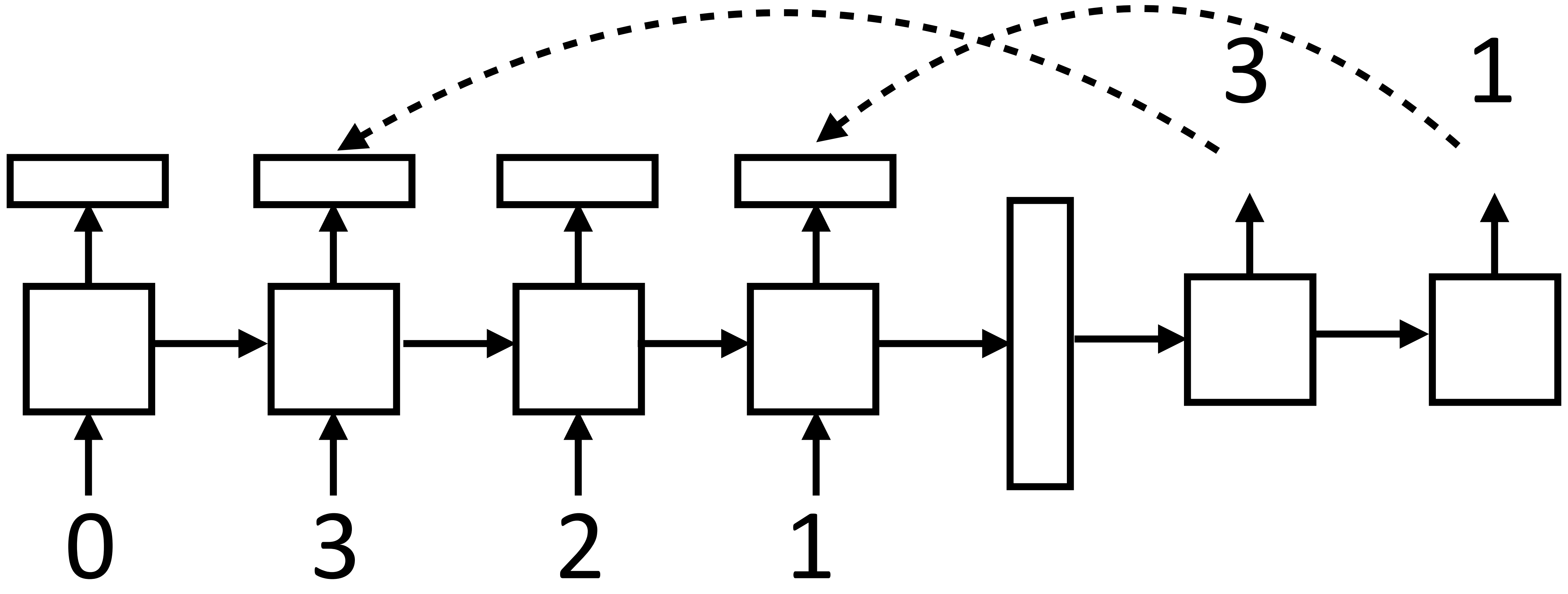


- LSTM can learn to count with the right weight matrix
- This is a kind of position-based addressing

Luong et al. (2015)

What can attention do?

- Learning to subsample tokens



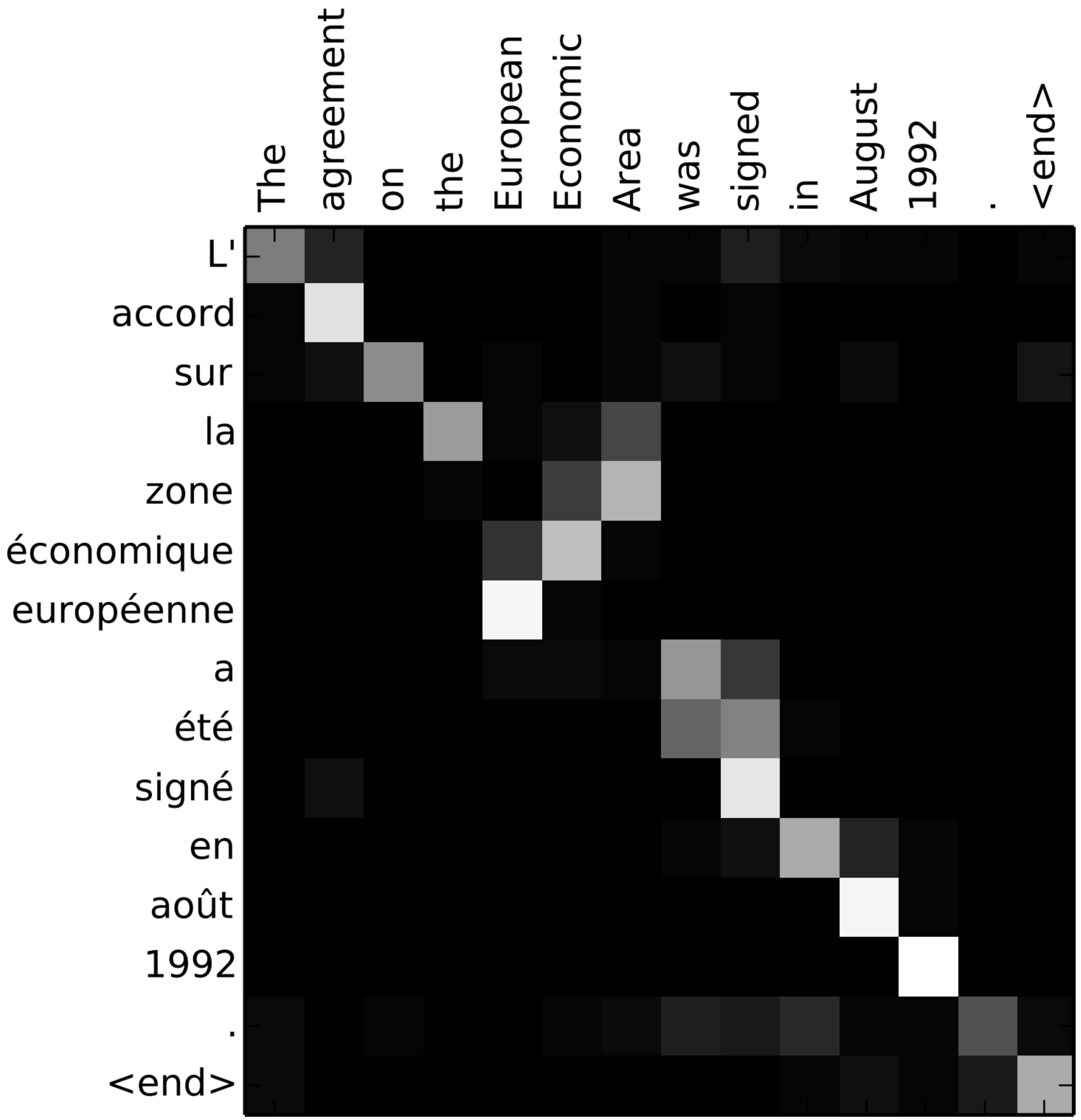
- Need to count (for ordering) and also determine which tokens are in/out

- Content-based addressing

Luong et al. (2015)

Attention

- Encoder hidden states capture contextual source word identity (“soft” word alignment)
- Decoder hidden states are now mostly responsible for selecting what to attend to
- Doesn’t take a complex hidden state to walk monotonically through a sentence and spit out word-by-word translations

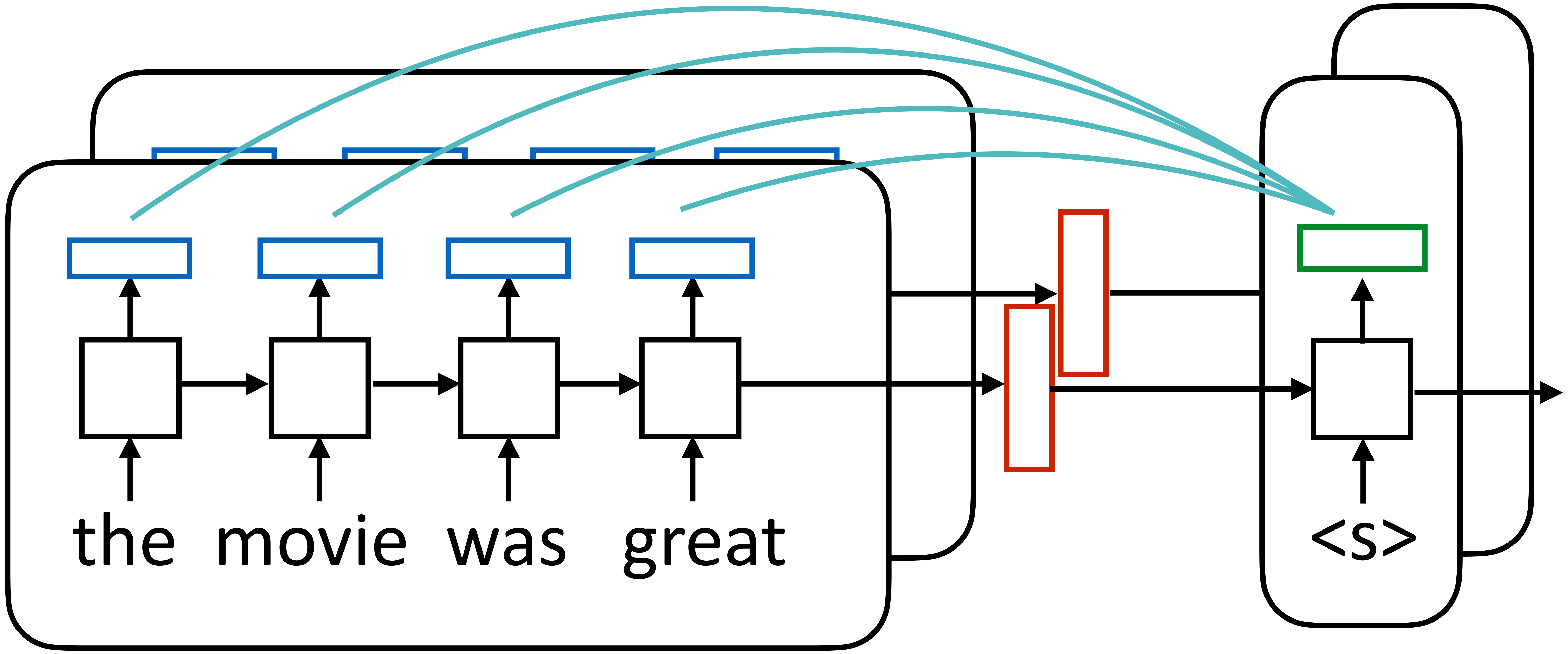


α_{ij}

Bahdanau et al. (2014)

Batching Attention

token outputs: batch size x sentence length x dimension



hidden state: batch size x hidden size

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

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

sentence outputs: batch size x hidden size

attention scores = batch size x sentence length

c = batch size x hidden size

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

- Make sure tensors are the right size!

Luong et al. (2015)

Some MT Results

“Early” Neural MT

Effective Approaches to Attention-based Neural Machine Translation

Minh-Thang Luong **Hieu Pham** **Christopher D. Manning**
Computer Science Department, Stanford University, Stanford, CA 94305
{lmthang,hyhieu,manning}@stanford.edu

Abstract

An attentional mechanism has lately been used to improve neural machine translation (NMT) by selectively focusing on parts of the source sentence during translation. However, there has been little work exploring useful architectures for attention-based NMT. This paper examines two simple and effective classes of attentional mechanism: a *global* approach which always attends to all source words and a *local* one that only looks at a subset of source words at a time. We demonstrate the effectiveness of both approaches on the

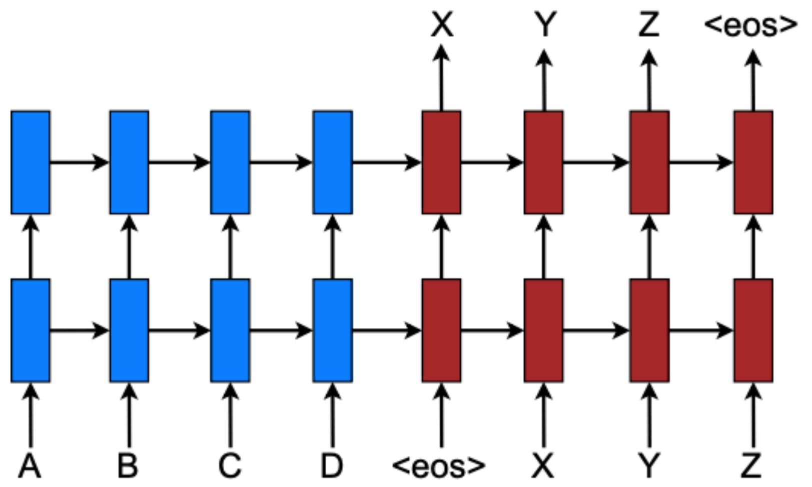


Figure 1: **Neural machine translation** – a stacking recurrent architecture for translating a source sequence A B C D into a target sequence X Y Z. Here, <eos> marks the end of a sentence.

ing plain SGD, (c) a simple learning rate schedule is employed – we start with a learning rate of 1; after 5 epochs, we begin to halve the learning rate every epoch, (d) our mini-batch size is 128, and (e) the normalized gradient is rescaled whenever its norm exceeds 5. Additionally, we also use dropout with probability 0.2 for our LSTMs as suggested by (Zaremba et al., 2015). For dropout models, we train for 12 epochs and start halving the learning rate after 8 epochs. For local attention models, we empirically set the window size $D = 10$.

Our code is implemented in MATLAB. When running on a single GPU device Tesla K40, we achieve a speed of 1K *target* words per second. It takes 7–10 days to completely train a model.

- TensorFlow first released in Nov 2015.
- PyTorch first released in 2016.

Luong et al. (2015)

MT Examples

src	In einem Interview sagte Bloom jedoch , dass er und Kerr sich noch immer lieben .
ref	However , in an interview , Bloom has said that he and *Kerr* still love each other .
best	In an interview , however , Bloom said that he and *Kerr* still love .
base	However , in an interview , Bloom said that he and **Tina** were still <unk> .

- best = with attention, base = no attention
- NMT systems can hallucinate words, especially when not using attention — phrase-based doesn't do this

Luong et al. (2015)

MT Examples

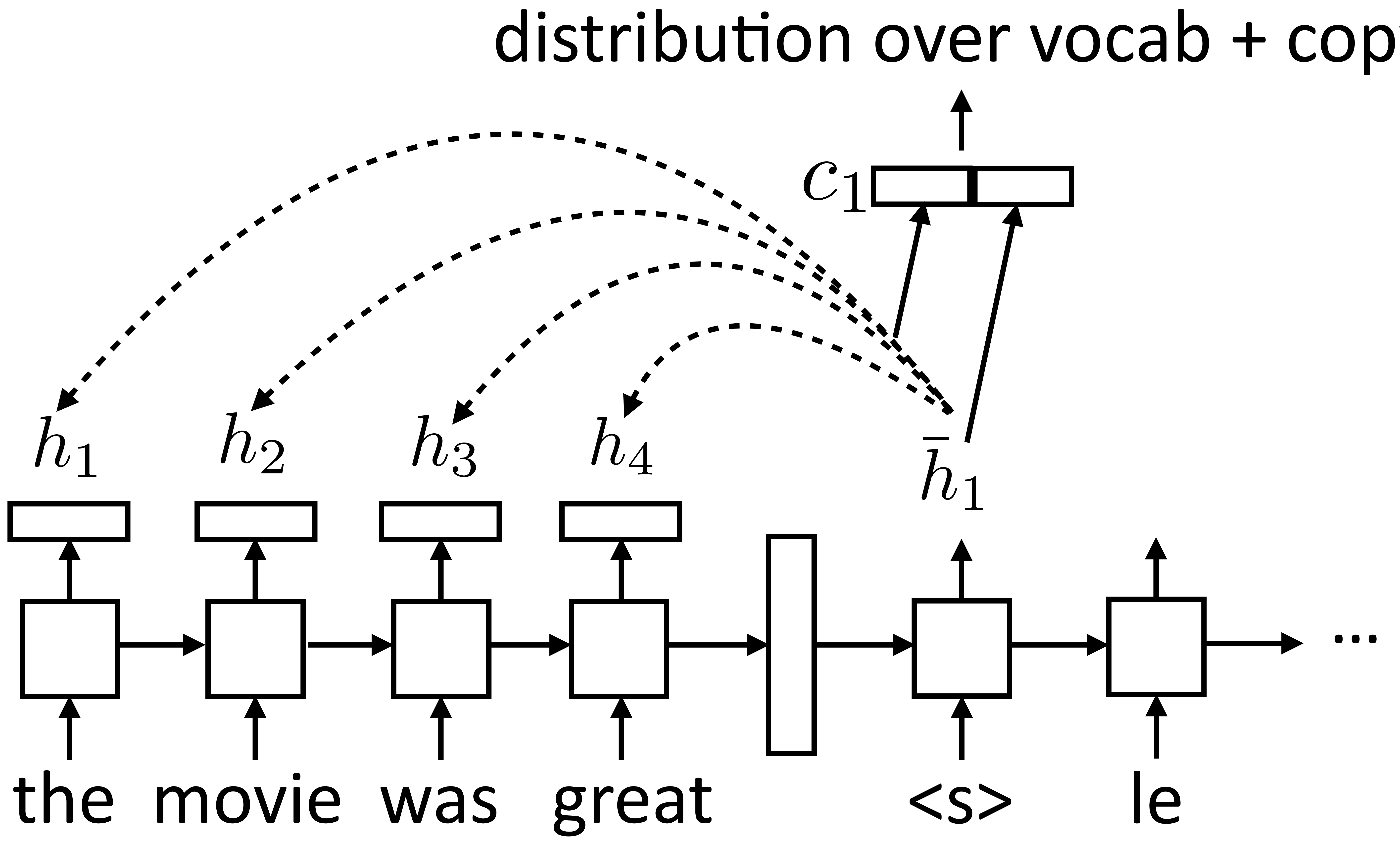
src	Wegen der von Berlin und der Europäischen Zentralbank verhängten strengen Sparpolitik in Verbindung mit der Zwangsjacke , in die die jeweilige nationale Wirtschaft durch das Festhalten an der gemeinsamen Währung genötigt wird , sind viele Menschen der Ansicht , das Projekt Europa sei zu weit gegangen
ref	The *austerity imposed by Berlin and the European Central Bank , coupled with the straitjacket* imposed on national economies through adherence to the common currency , has led many people to think Project Europe has gone too far .
best	Because of the strict *austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket* in which the respective national economy is forced to adhere to the common currency , many people believe that the European project has gone too far .
base	Because of the pressure **imposed by the European Central Bank and the Federal Central Bank with the strict austerity** imposed on the national economy in the face of the single currency , many people believe that the European project has gone too far .

- best = with attention, base = no attention

Luong et al. (2015)

Encoder-Decoder MT

- Better encoder-decoder with attention and handling of rare words

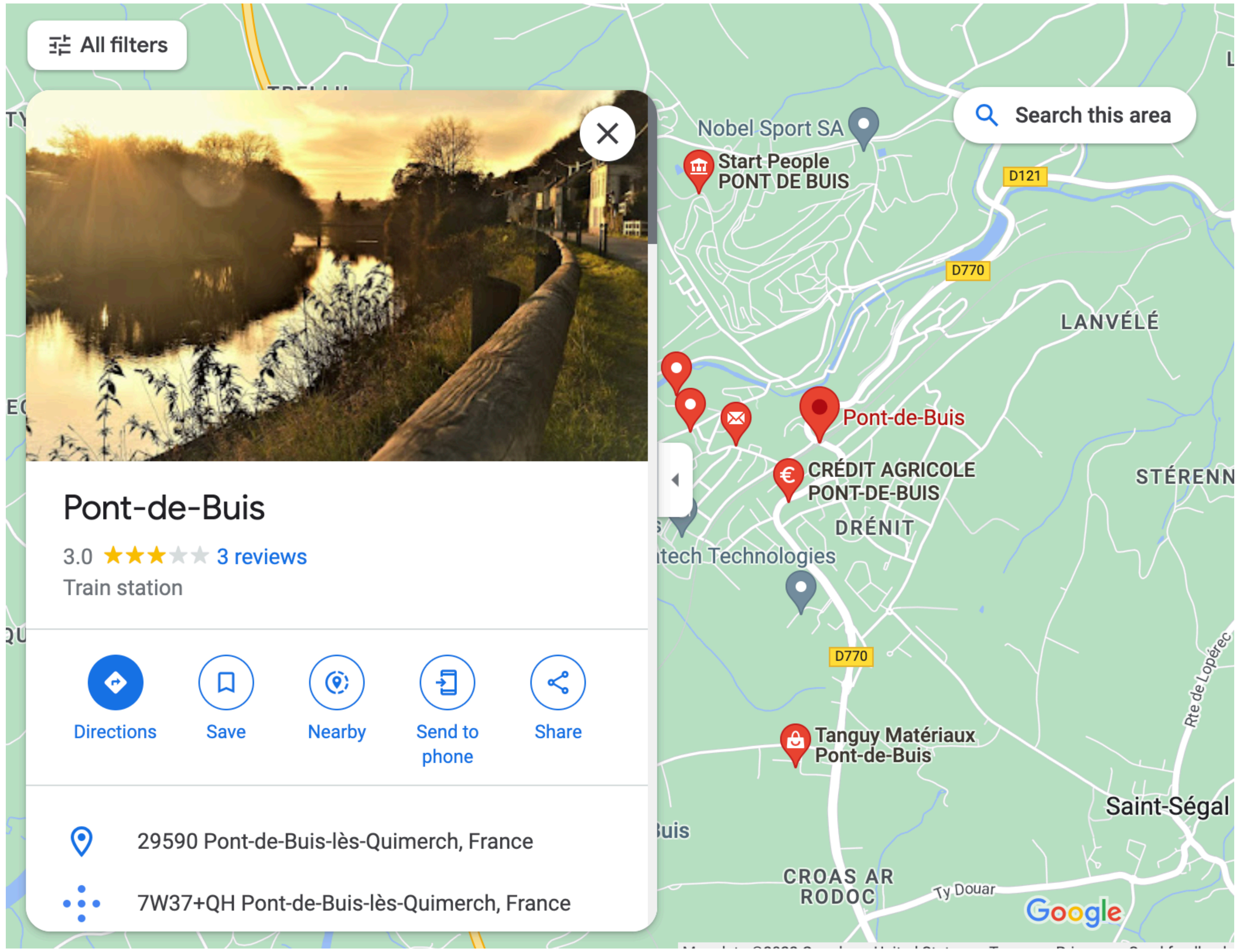


Luong et al. (2015)

Copy / Pointer Networks

Rare/Unknown Words

The ecotax portico in Pont-de-Buis, around which a violent demonstration against the tax took place on Saturday, was taken down on Thursday morning.



Unknown Words

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

copy

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

- Attention mechanism:

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

from attention

from RNN hidden state

- Problems: want to be able to copy named entities like Pont-de-Buis, but 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

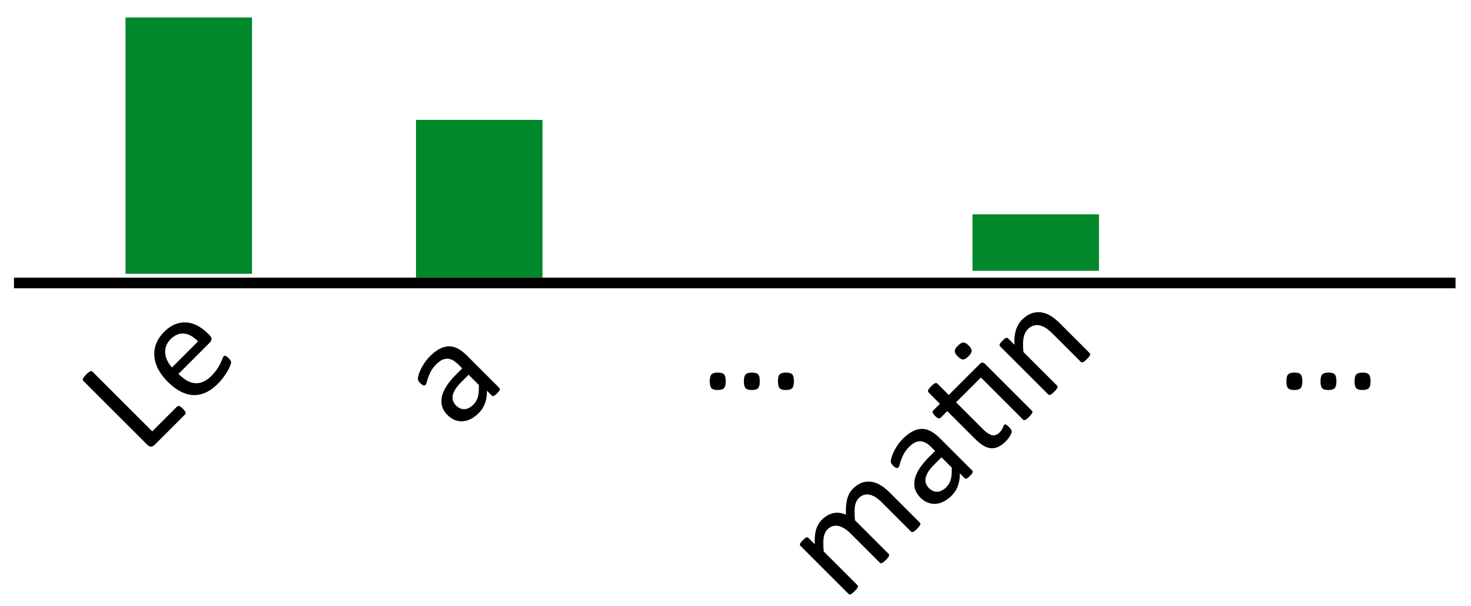
Pont-de-Buis
ecotax

- 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.

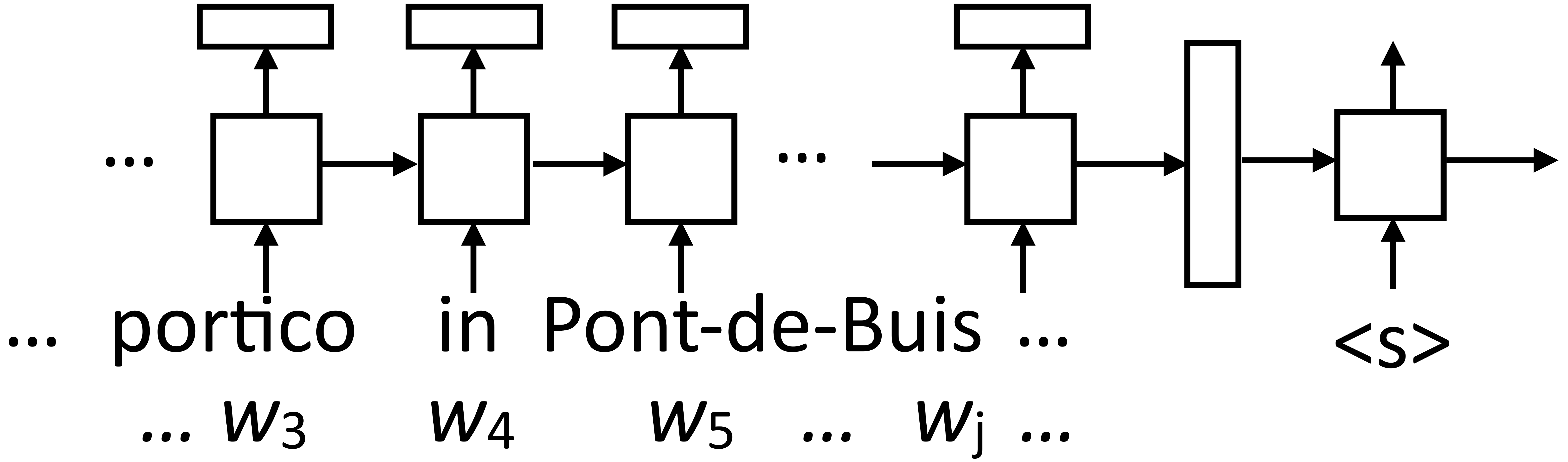
Pointer Network

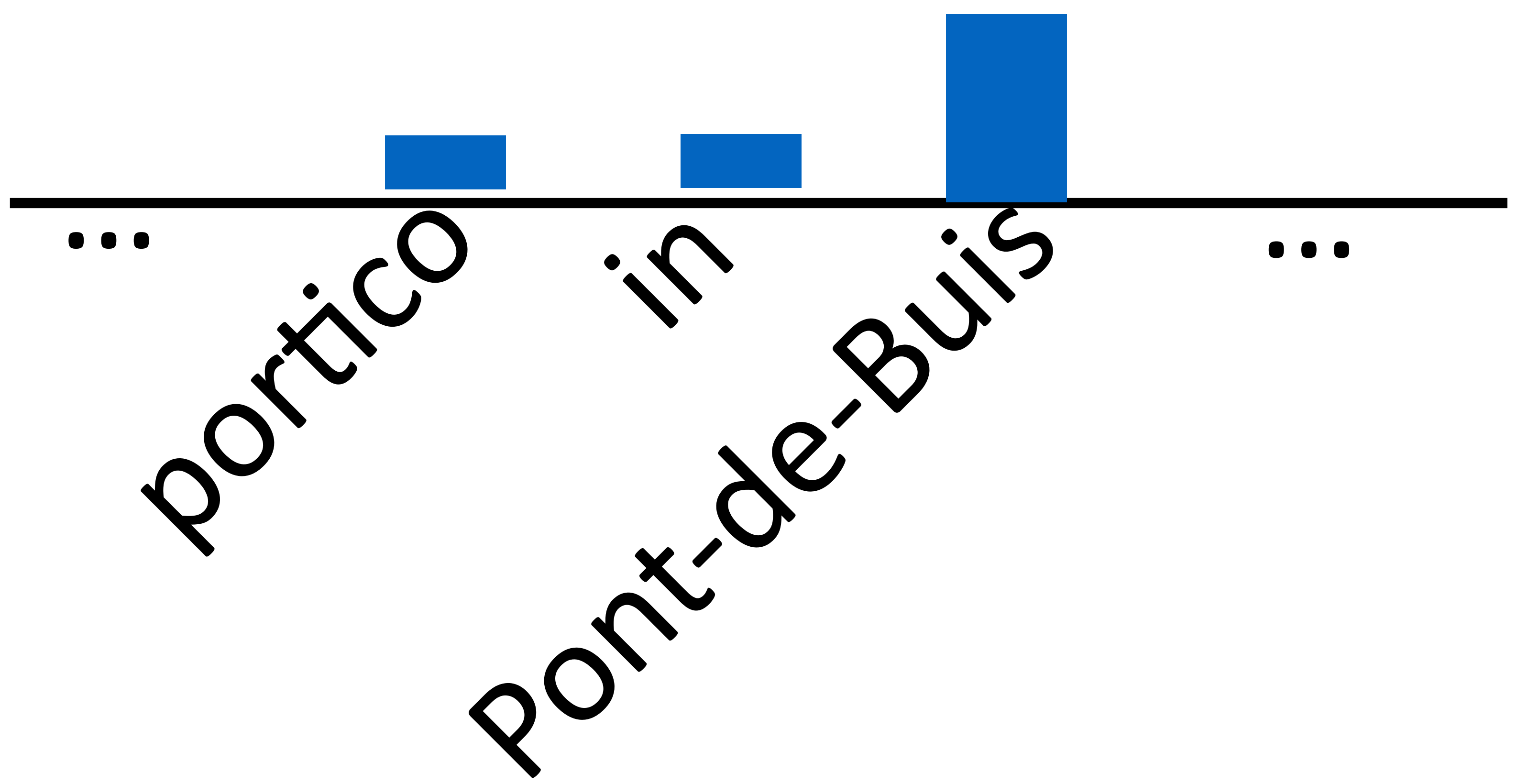
- Standard decoder with attention (P_{vocab}): softmax over vocabulary

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



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





Pointer Generator Mixture Models

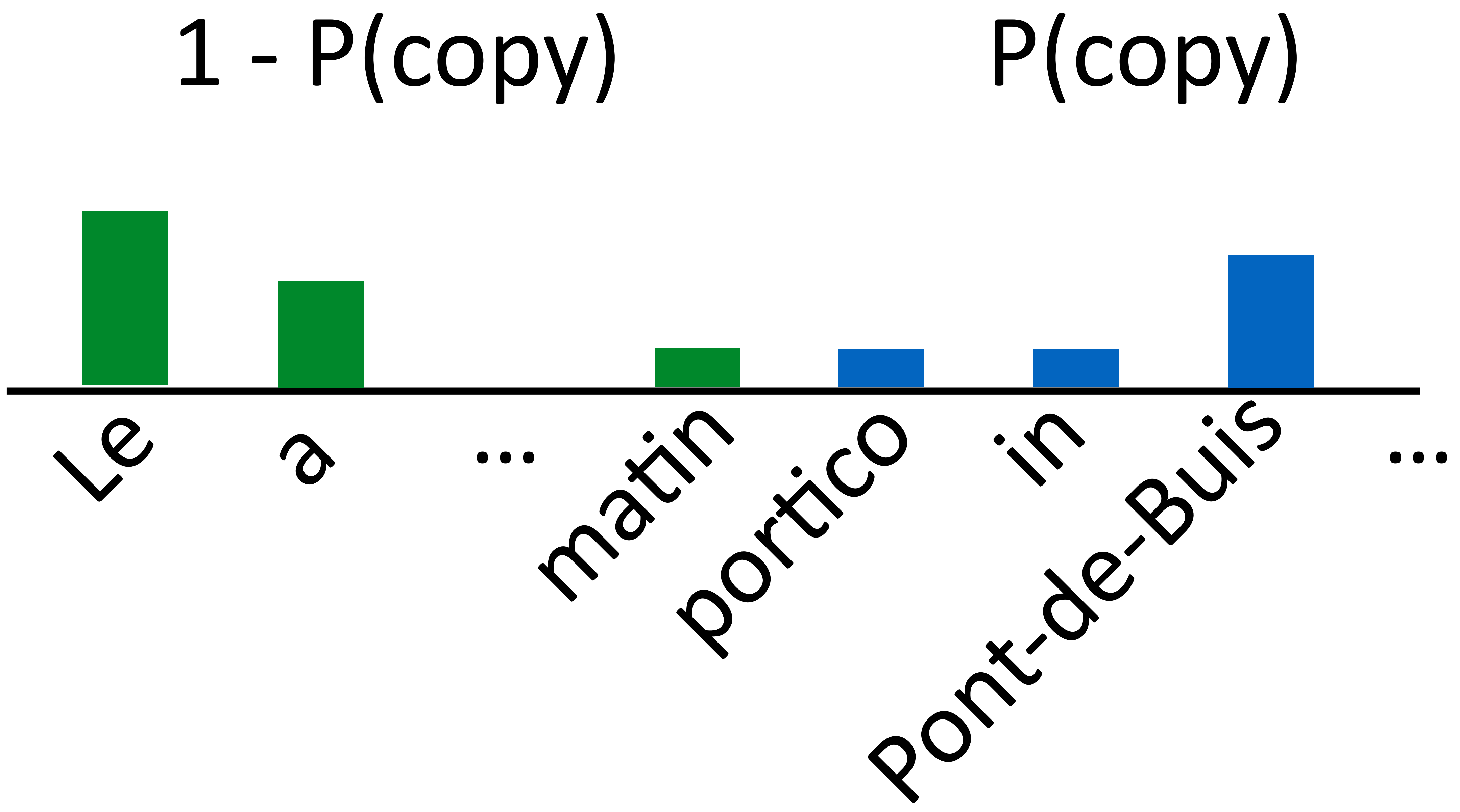
- Define the decoder model as a mixture model of P_{vocab} and P_{pointer}

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

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

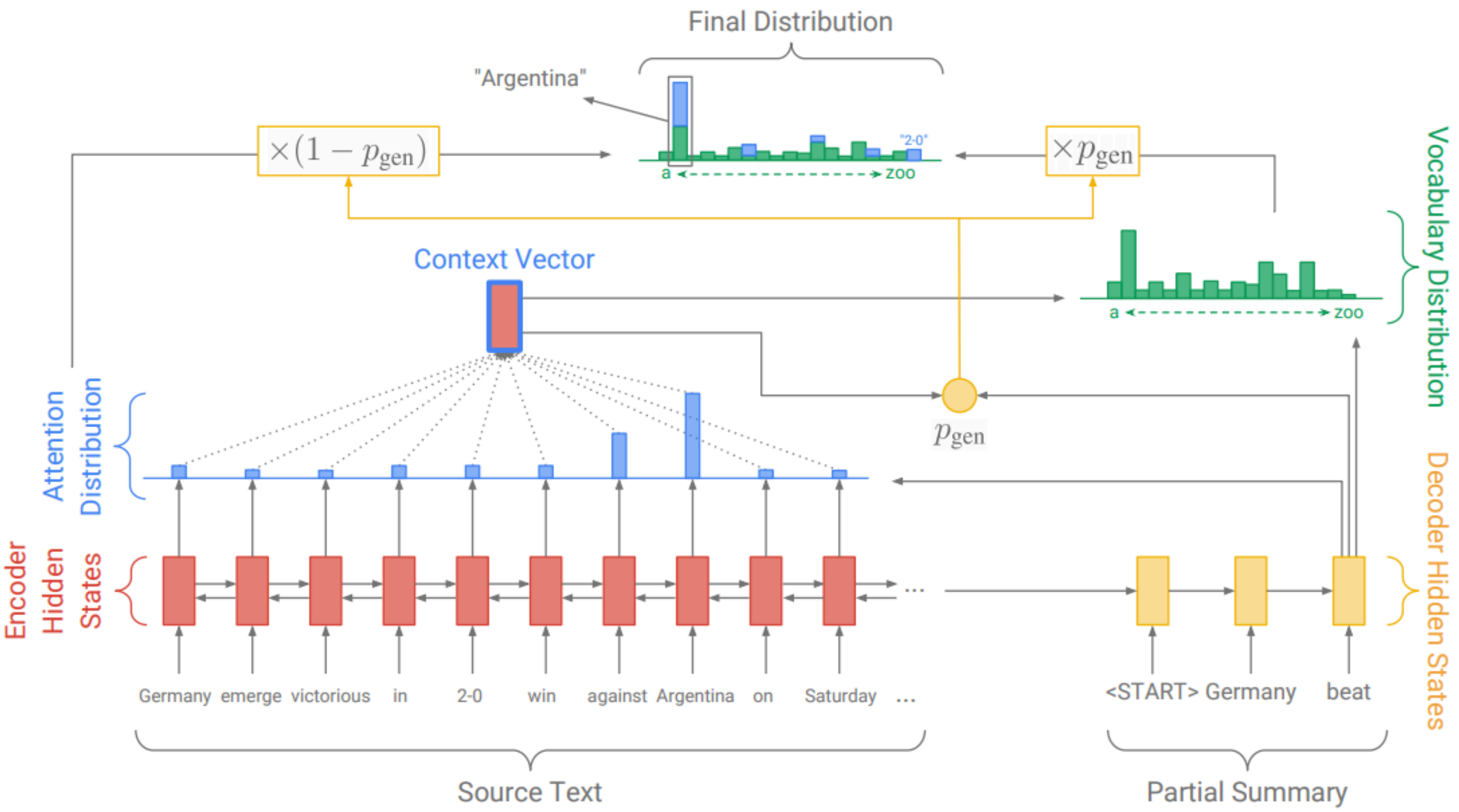
- Marginalize over copy variable during training and inference

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



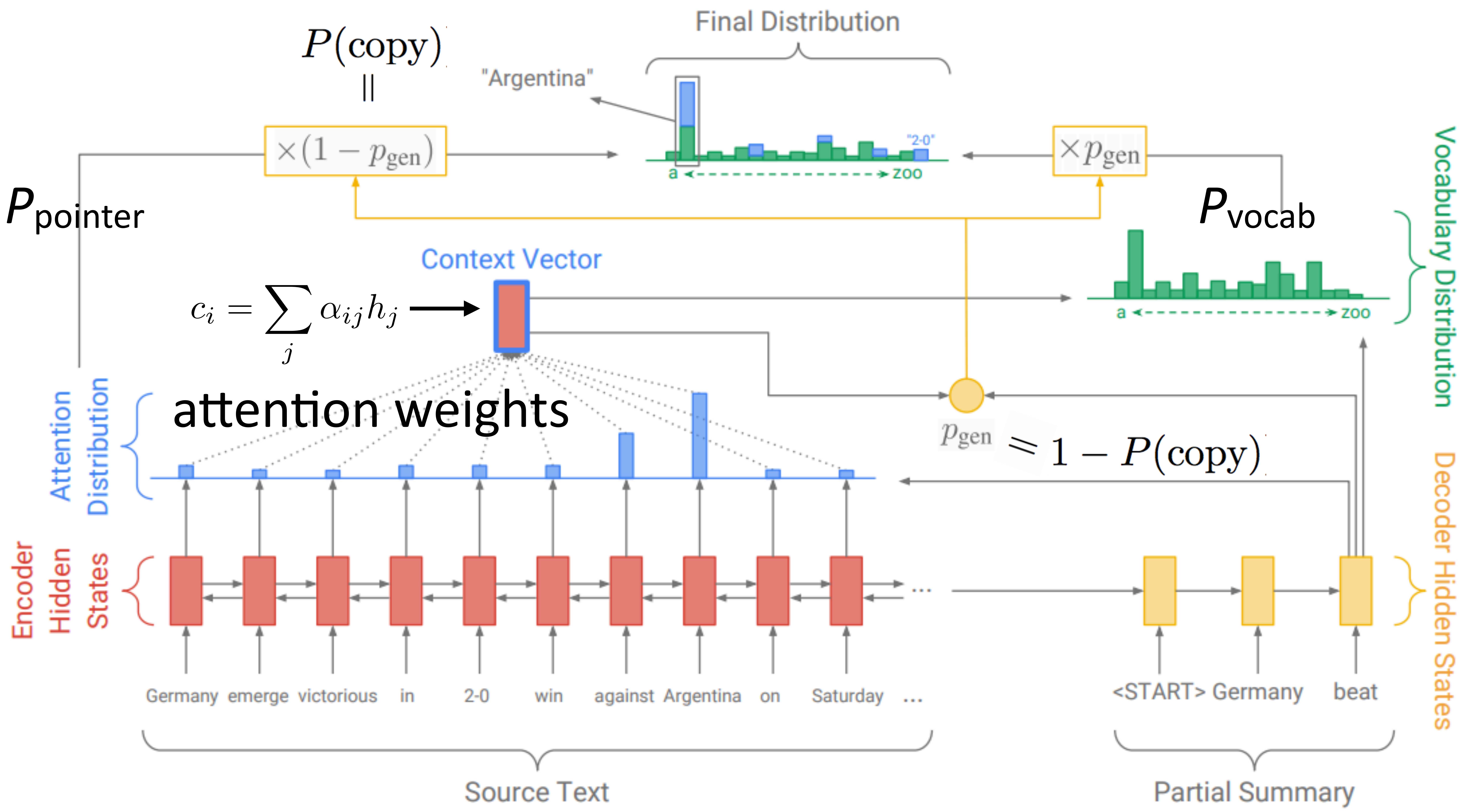
Gulcehre et al. (2016), Gu et al. (2016)

Copying in Summarization



See et al. (2017)

Copying in Summarization



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	-	-

- maintain a coverage vector, which is the sum of attention distributions over all previous decoder timesteps

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)

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU (input reversed)

Sutskever+ (2014) seq2seq ensemble: **34.8** BLEU

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

- But English-French is a really easy language pair and there's *tons* of data for it! Does this approach work for anything harder?

Results: WMT English-German

- 4.5M sentence pairs

Classic phrase-based system: **20.7** BLEU

Luong+ (2014) seq2seq: **14** BLEU

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

- Not nearly as good in absolute BLEU, but not really comparable across languages
- French, Spanish = easiest
  German, Czech = harder
  Japanese, Russian = hard (grammatically different, lots of morphology…)

Tokenization

Recap: Problems with Seq2seq Models

- 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

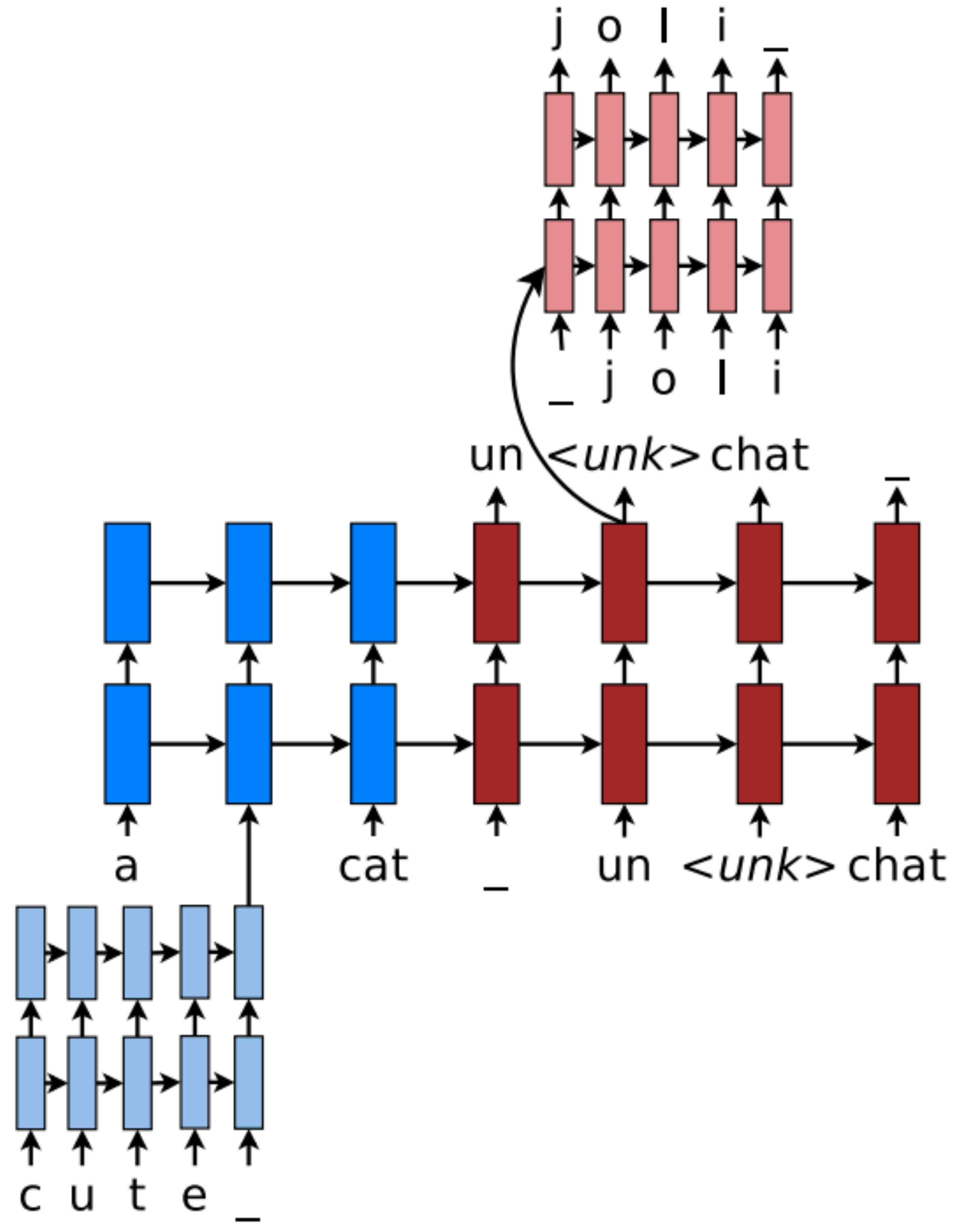
nn: Le `unk` de `unk` à `unk` , . . . [truncated] . . . , a été pris le jeudi matin

- Encoding these rare words into a vector space is really hard

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

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

Algorithm 1 Byte-pair encoding (Sennrich et al., 2016; Gage, 1994)

Input: set of strings D, target vocab size k
procedure BPE(D, k)
    $V \leftarrow$ all unique characters in D
        (about 4,000 in English Wikipedia)
    while $|V| < k$ **do** ▷ Merge tokens
        $t_L, t_R \leftarrow$ Most frequent bigram in D
        $t_{\text{NEW}} \leftarrow t_L + t_R$ ▷ Make new token
        $V \leftarrow V + [t_{\text{NEW}}]$
        Replace each occurrence of t_L, t_R in
            D with t_{NEW}
    end while
    return V
end procedure

Sennrich et al. (2016); Figure from Bostrom and Durrett (2020)

Byte Pair Encoding (BPE)

- 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

- Alternatively, can learn word pieces based on unigram LM:

while voc size < target voc size:

Build a language model over your corpus

Merge pieces that lead to highest improvement in language model perplexity

- Result: way of segmenting input appropriate for translation
- SentencePiece library from Google: unigram LM & BPE
- Large pre-trained language models are all using this too!

Sennrich et al. (2016), Kudo (2018)

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	

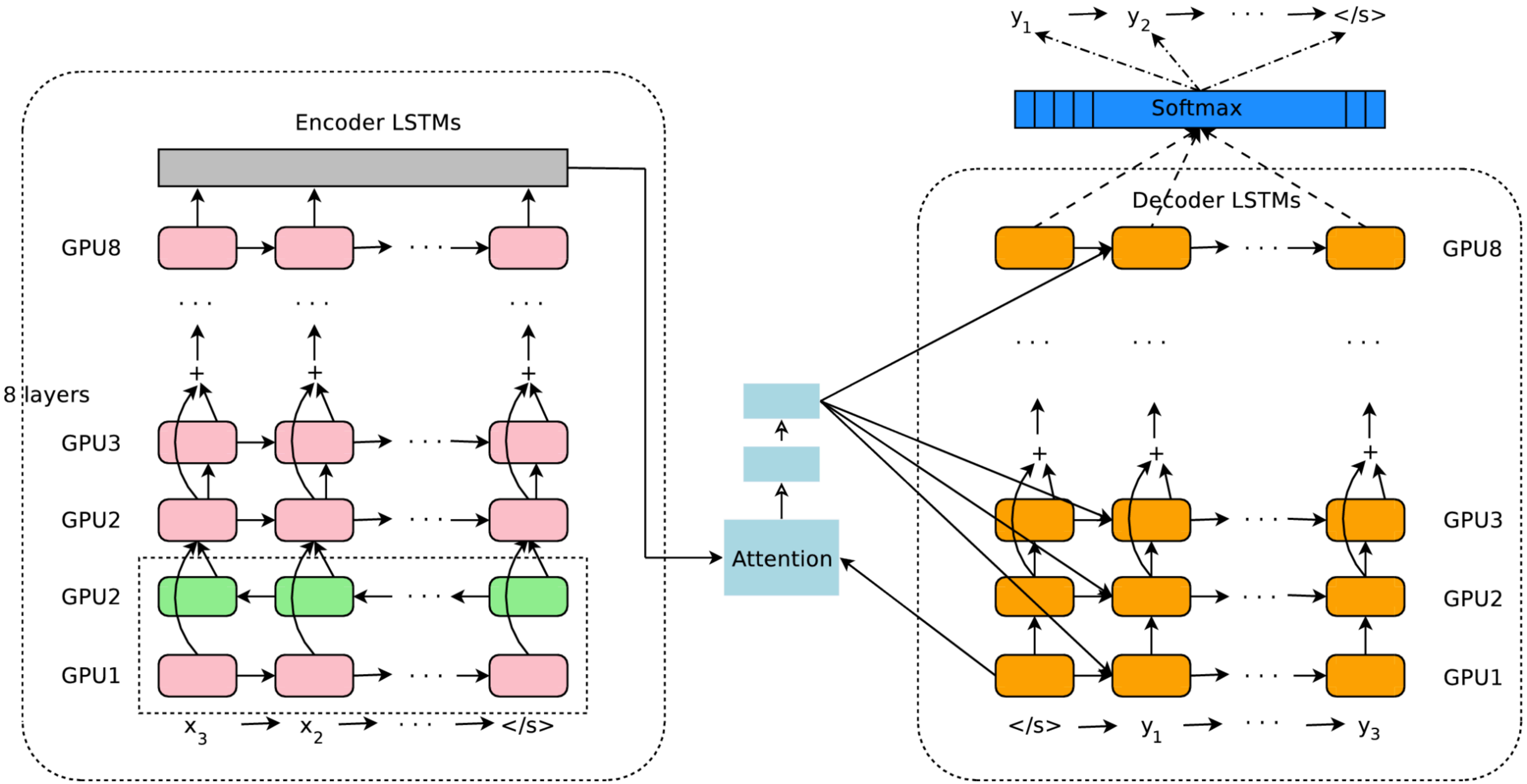
(c)	**Original:**	Completely preposterous suggestions									
	BPE:	_Comple	t	ely	_prep	ost	erous		_suggest	ions	
	Unigram LM:	_Complete	ly		_pre	post	er	ous	_suggestion	s	

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

Bostrom and Durrett (2020)

Google NMT

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)

Human Evaluation (En-Es)

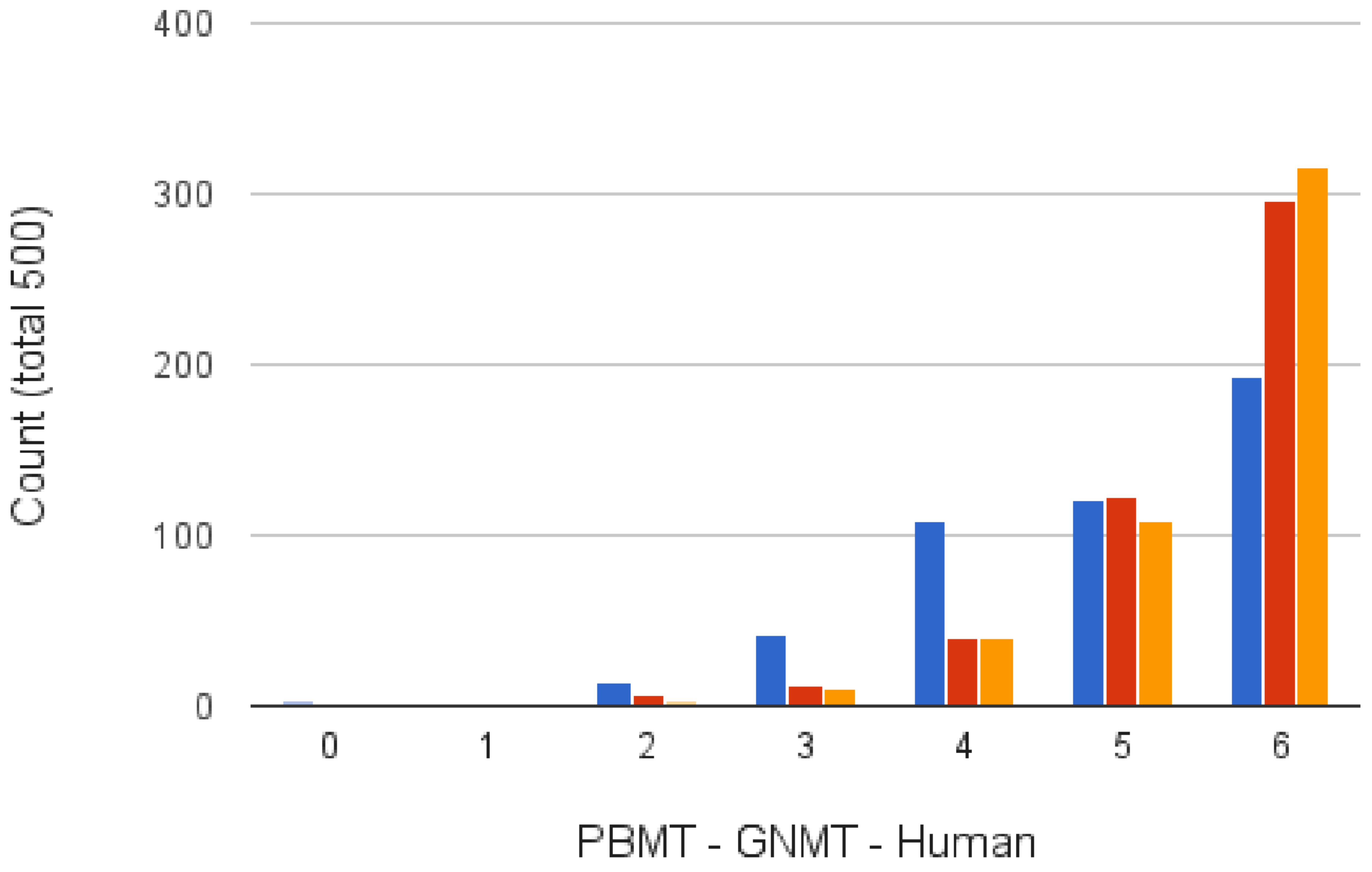


Figure 6: Histogram of side-by-side scores on 500 sampled sentences from Wikipedia and news websites for a typical language pair, here English → Spanish (PBMT blue, GNMT red, Human orange). It can be seen that there is a wide distribution in scores, even for the human translation when rated by other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

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

"walker"

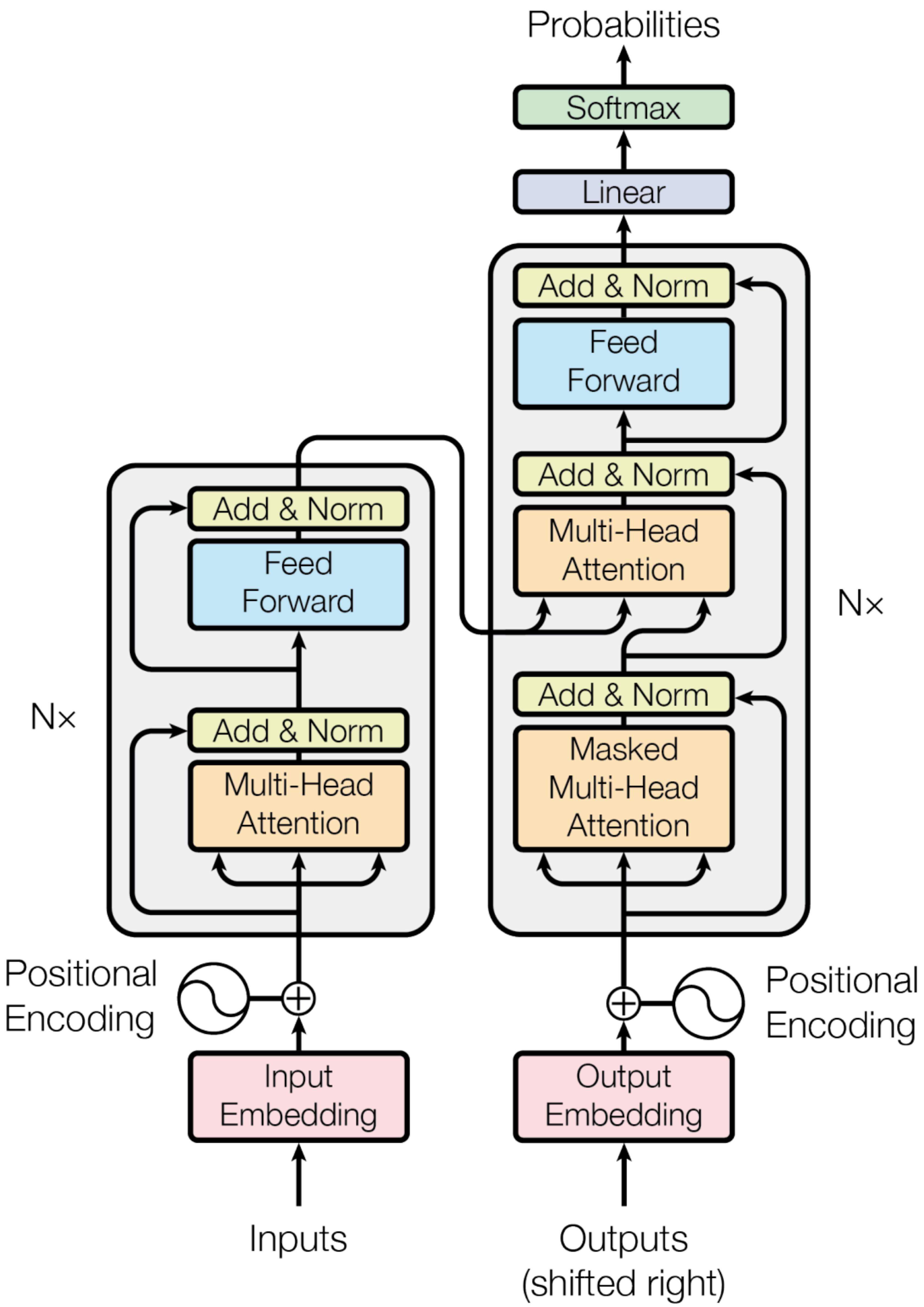
Gender is correct in GNMT but not in PBMT

"sled"

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

Wu et al. (2016)

Transformer (more later)



Model	BLEU	
	EN-DE	EN-FR
ByteNet [18]	23.75	
Deep-Att + PosUnk [39]		39.2
GNMT + RL [38]	24.6	39.92
ConvS2S [9]	25.16	40.46
MoE [32]	26.03	40.56
Deep-Att + PosUnk Ensemble [39]		40.4
GNMT + RL Ensemble [38]	26.30	41.16
ConvS2S Ensemble [9]	26.36	**41.29**
Transformer (base model)	27.3	38.1
Transformer (big)	**28.4**	**41.8**

Vaswani et al. (2017)

Frontiers in MT

Low-Resource MT

- Particular interest in deploying MT systems for languages with little or no parallel data
- BPE allows us to transfer models even without training on a specific language
- Pre-trained models can help further

Burmese, Indonesian, Turkish

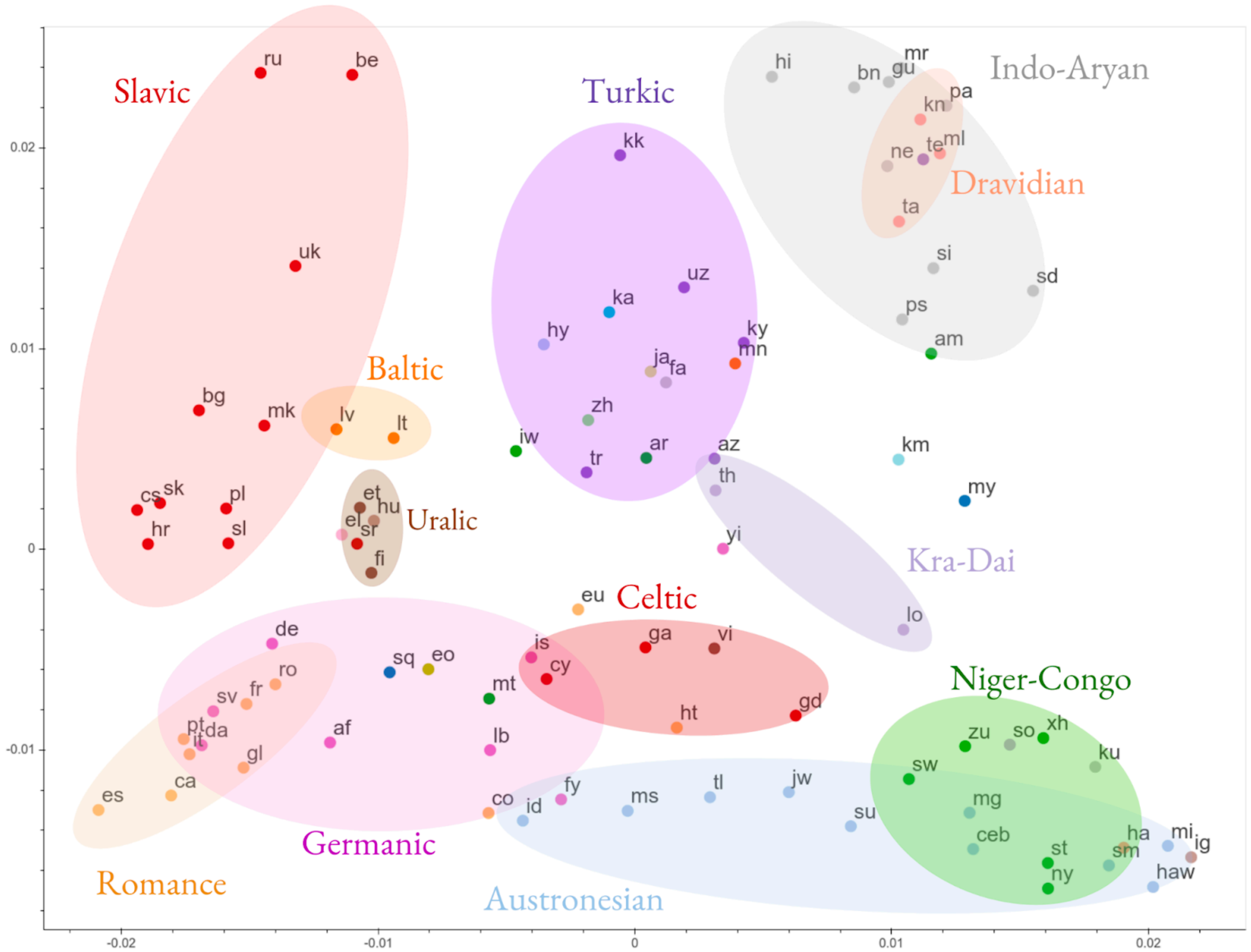
Transfer	BLEU My→En	Id→En	Tr→En
baseline (no transfer)	4.0	20.6	19.0
transfer, train	17.8	27.4	20.3
transfer, train, reset emb, train	13.3	25.0	20.0
transfer, train, reset inner, train	3.6	18.0	19.1

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

Aji et al. (2020)

Massively Multilingual MT

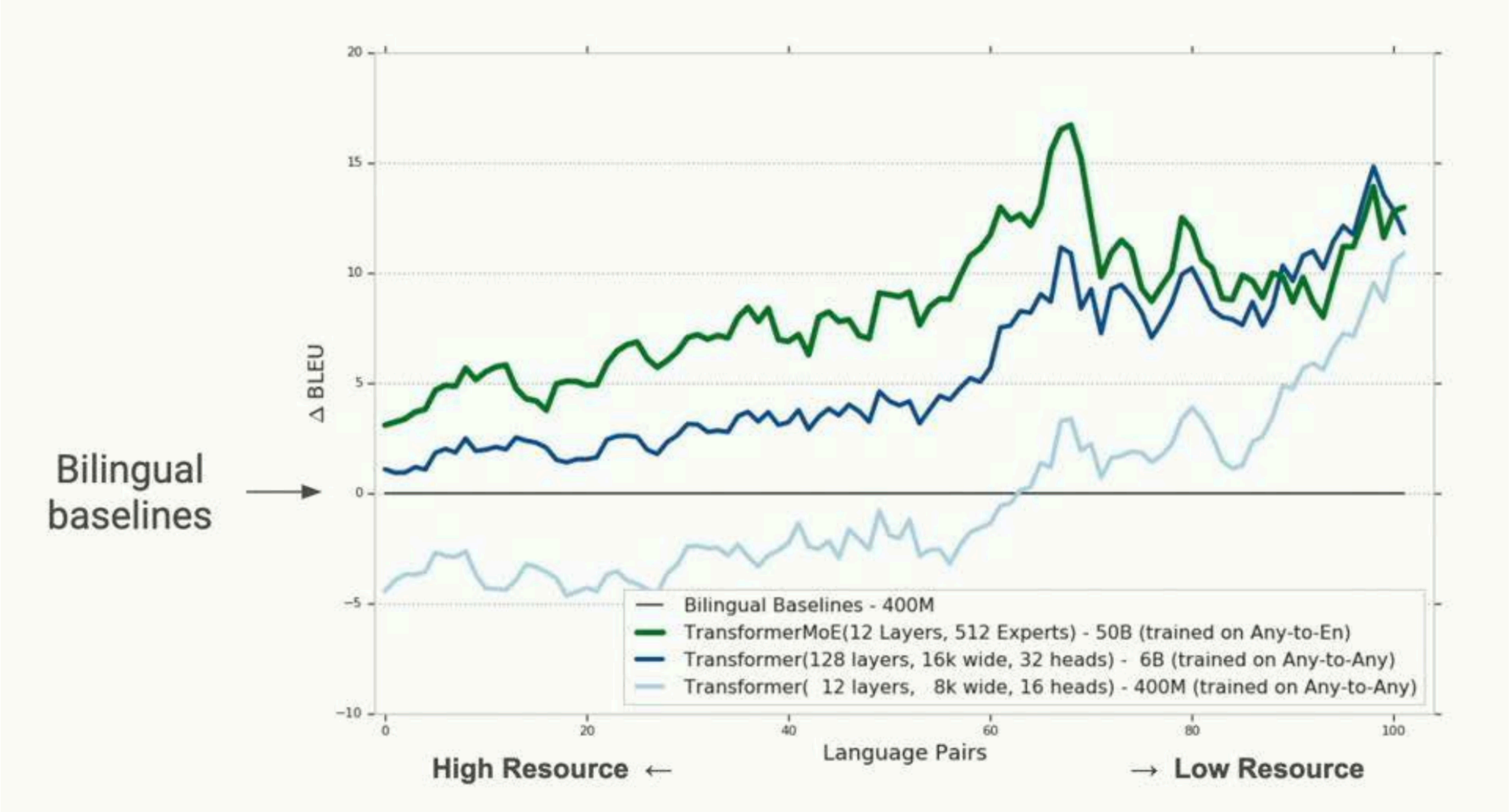
- For 103 languages



Arivazhagan et al. (2019), Kudugunta et al. (2019)

Massively Multilingual MT

- For 103 languages



Arivazhagan et al. (2019), Kudugunta et al. (2019)

Massively Multilingual MT

- For 200 languages (54B parameters)
  - Mixture of Expert (BOE) model. With more low-resource language pairs in the training data, the multilingual systems start to overfit.
  - Solutions: regularization, curriculum learning, self-supervised learning, and diversifying back-translation.

	eng_Latn-xx			xx-eng_Latn		
	MMTAfrica	M2M-100*	NLLB-200	MMTAfrica	M2M-100*	NLLB-200
hau_Latn	-/-	4.0/-	**33.6**/**53.5**	-/-	16.3/-	**38.5**/**57.3**
ibo_Latn	21.4/-	19.9/-	**25.8**/**41.4**	15.4/-	12.0/-	**35.5**/**54.4**
lug_Latn	-/-	7.6/-	**16.8**/**39.8**	-/-	7.7/-	**27.4**/**46.7**
luo_Latn	-/-	13.7/-	**18.0**/**38.5**	-/-	11.8/-	**24.5**/**43.7**
swh_Latn	**40.1**/-	27.1/-	37.9/**58.6**	28.4/-	25.8/-	**48.1**/**66.1**
wol_Latn	-/-	8.2/-	**11.5**/**29.7**	-/-	7.5/-	**22.4**/**41.2**
xho_Latn	27.1/-	-/-	**29.5**/**48.6**	21.7/-	-/-	**41.9**/**59.9**
yor_Latn	12.0/-	13.4/-	**13.8**/**25.5**	9.0/-	9.3/-	**26.6**/**46.3**
zul_Latn	-/-	19.2/-	**36.3**/**53.3**	-/-	19.2/-	**43.4**/**61.5**

Table 31: **Comparison on FLORES-101 devtest on African Languages.** We compare to two

Fan et al. (2022), NLLB Team (2022)

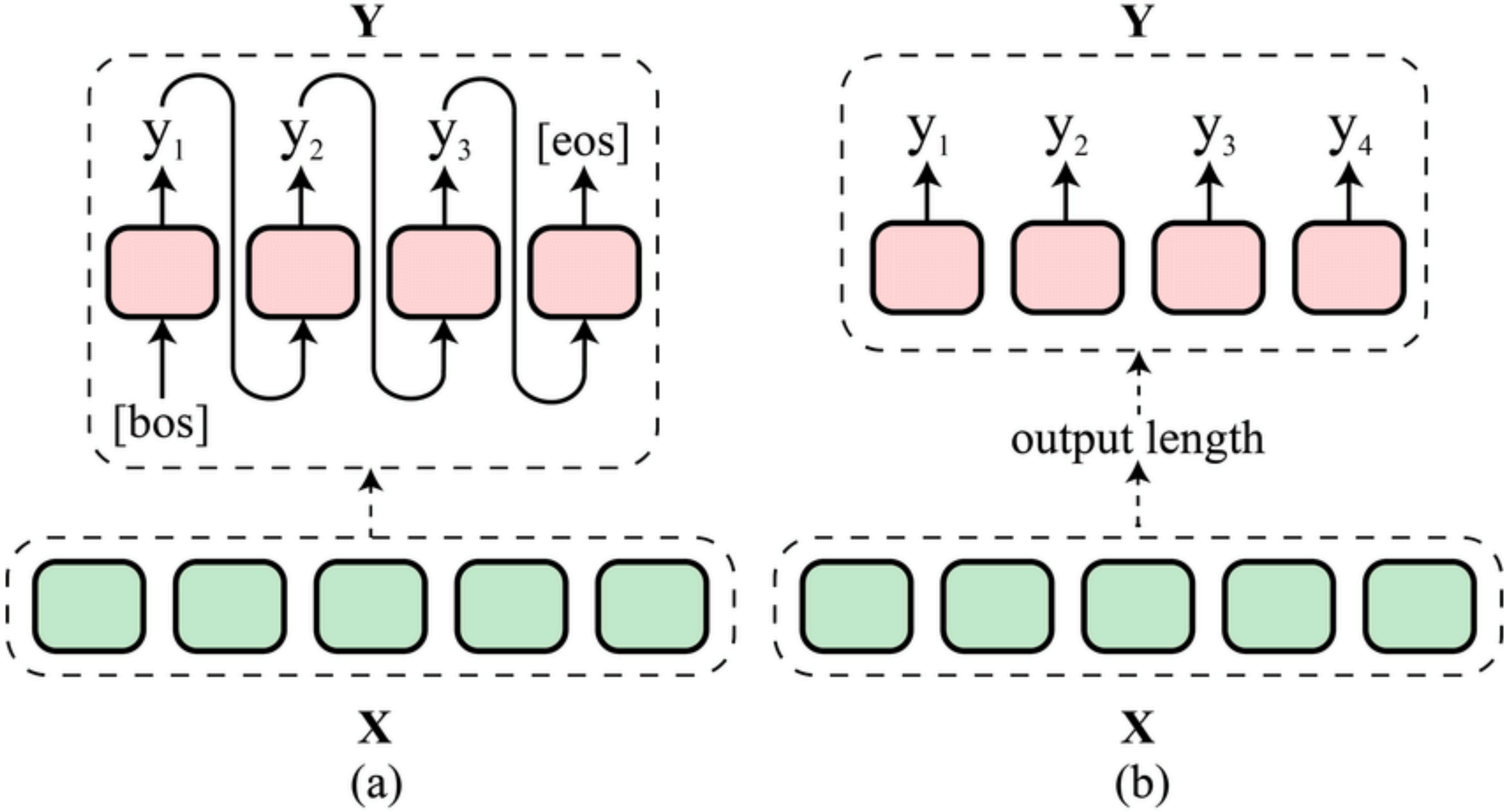
Unsupervised MT

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

- Common principles of unsupervised MT
  - Language models
  - (Iterative) Back-translation! - The goal of this model is to generate a source sentence for each target sentence in the monolingual corpus.

Lample et al. (2018)

Non-Autoregressive NMT



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

Efficiency of NMT

EMNLP 2022
SEVENTH CONFERENCE ON
MACHINE TRANSLATION (WMT22)

December 7-8, 2022
Abu Dhabi

Shared Task: Efficiency

[HOME]
TRANSLATION TASKS: [GENERAL MT (NEWS)] [BIOMEDICAL] [LARGE-SCALE MULTILINGUAL AFRICAN] [EFFICIENCY] [SIGN LANGUAGE] [CODE MIXED] [CHAT] [UNSUP AND VERY LOW RES]
EVALUATION TASKS: [METRICS] [QUALITY ESTIMATION]
OTHER TASKS: [WORD-LEVEL AUTOCOMPLETION] [TRANSLATION SUGGESTION] [AUTOMATIC POST-EDITING]

Efficiency Task

The efficiency task measures latency, throughput, memory consumption, and size of machine translation on CPUs and GPUs. Participants provide their own code and models using standardized data and hardware. This is a continuation of the WMT 2021 Efficiency Shared Task.

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