Neural Machine Translation

Berkeley

NLP

John DeNero

CS 288, UC Berkeley

Decoding for Phrase-Based Machine Translation

Search state:

- The most recent n−1 target words (for n−gram language model)
- Coverage of source words (to ensure each word translated once)
- Most recent source position translated (for reordering)

Path score:

- Translation, language model, and reordering (distortion) scores
- Optimistic estimate of future translation & LM scores

Search strategy:

- Build target sentence left−to−right (to score language model)
- Each new state added by translating one untranslated phrase
- Extend a partial translation only if it's among the top K ways to translate N source words.

(Koehn Slides)

Neural Sequence-to-Sequence Models

Conditional Sequence Generation

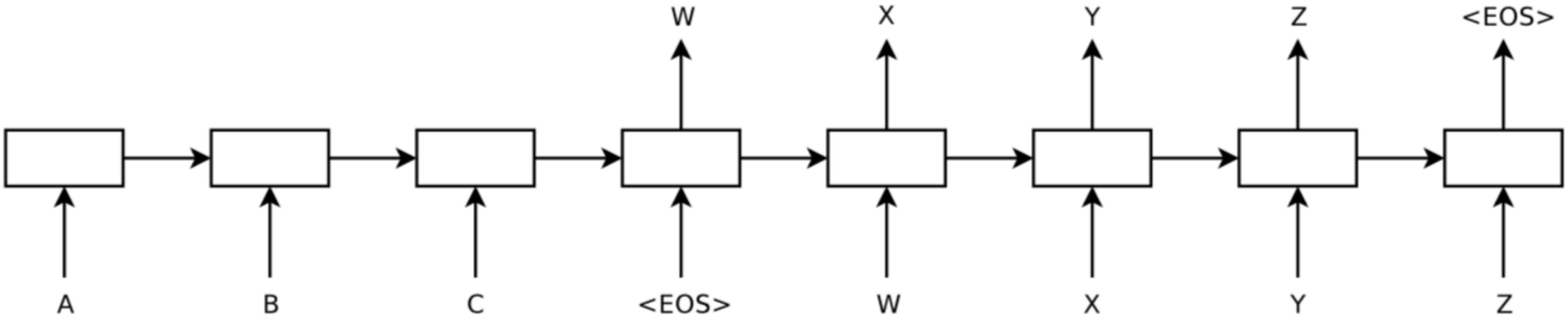
P(e|f) could just be estimated from a sequence model P(f, e)

<f> das Haus ist klein </f> the house is small </e>

Run an RNN over the whole sequence, which first computes P(f), then computes P(e, f).

Encoder-Decoder: Use different parameters or architectures encoding f and predicting e.

"Sequence to sequence" learning (Sutskever et al., 2014)



(Sutskever et al., 2014) Sequence to sequence learning with neural networks.

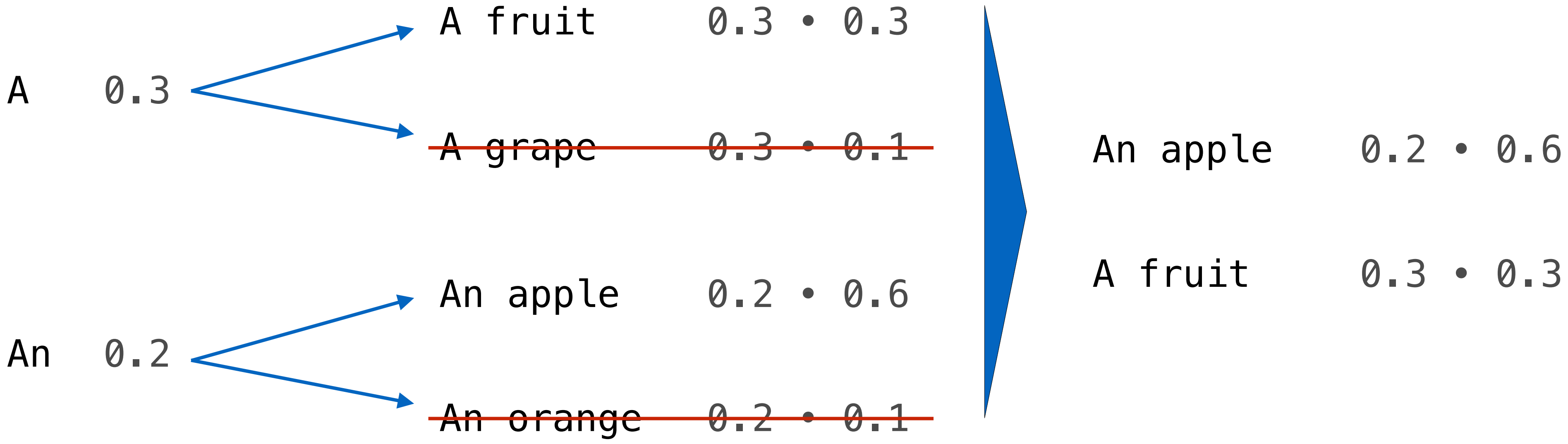
Neural Decoding

Search Strategies for Neural Machine Translation

For each target position, each word in the vocabulary is scored. (Alternatively, a restricted list of vocabulary items can be selected based on the source sentence, but quality can degrade.)

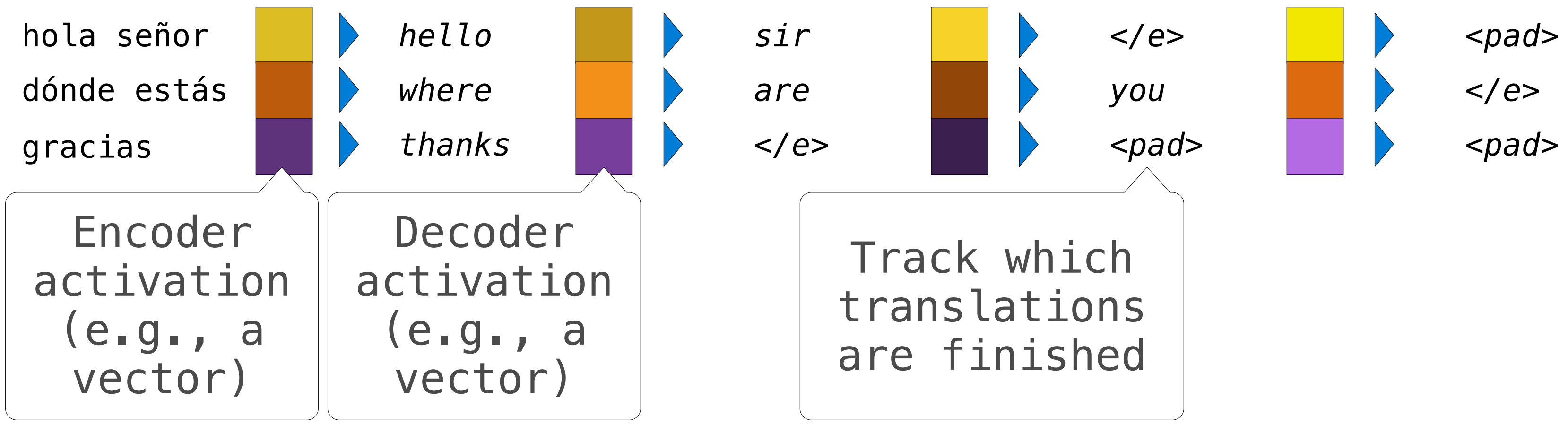
Greedy decoding: Extend a single hypothesis (partial translation) with the next word that has highest probability.

Beam search: Extend multiple hypotheses, then prune.



Implementing Beam Search for Batch Decoding

Greedy search:



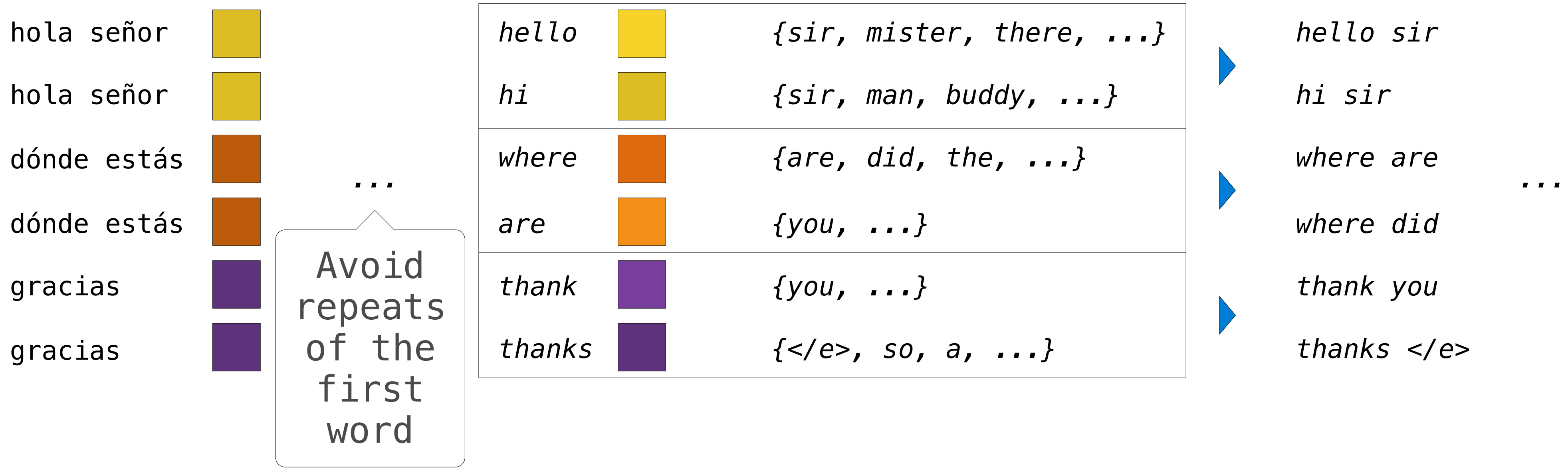
hola señor	*hello*	*sir*	*</e>*	*<pad>*
dónde estás	*where*	*are*	*you*	*</e>*
gracias	*thanks*	*</e>*	*<pad>*	*<pad>*

Encoder activation (e.g., a vector)

Decoder activation (e.g., a vector)

Track which translations are finished

Beam search (beam width of 2):



hola señor	*hello*	*{sir, mister, there, ...}*	*hello sir*
hola señor	*hi*	*{sir, man, buddy, ...}*	*hi sir*
dónde estás	*where*	*{are, did, the, ...}*	*where are*
dónde estás	*are*	*{you, ...}*	*where did*
gracias	*thank*	*{you, ...}*	*thank you*
gracias	*thanks*	*{</e>, so, a, ...}*	*thanks </e>*

...

Avoid repeats of the first word

...

Beam Search Criteria to Compensate for Bad Models

NMT models often prefer translations that are too short.

$$s(e) = \sum_{i=1}^{m} \log P(e_i | e_{1:i}, f)$$

"For more than 50% of the sentences, the model in fact assigns its global best score to the empty translation"
(Stahlberg & Byrne, 2019)

Alternatives for scoring items on the beam:

Length normalization: $s(e)/m$

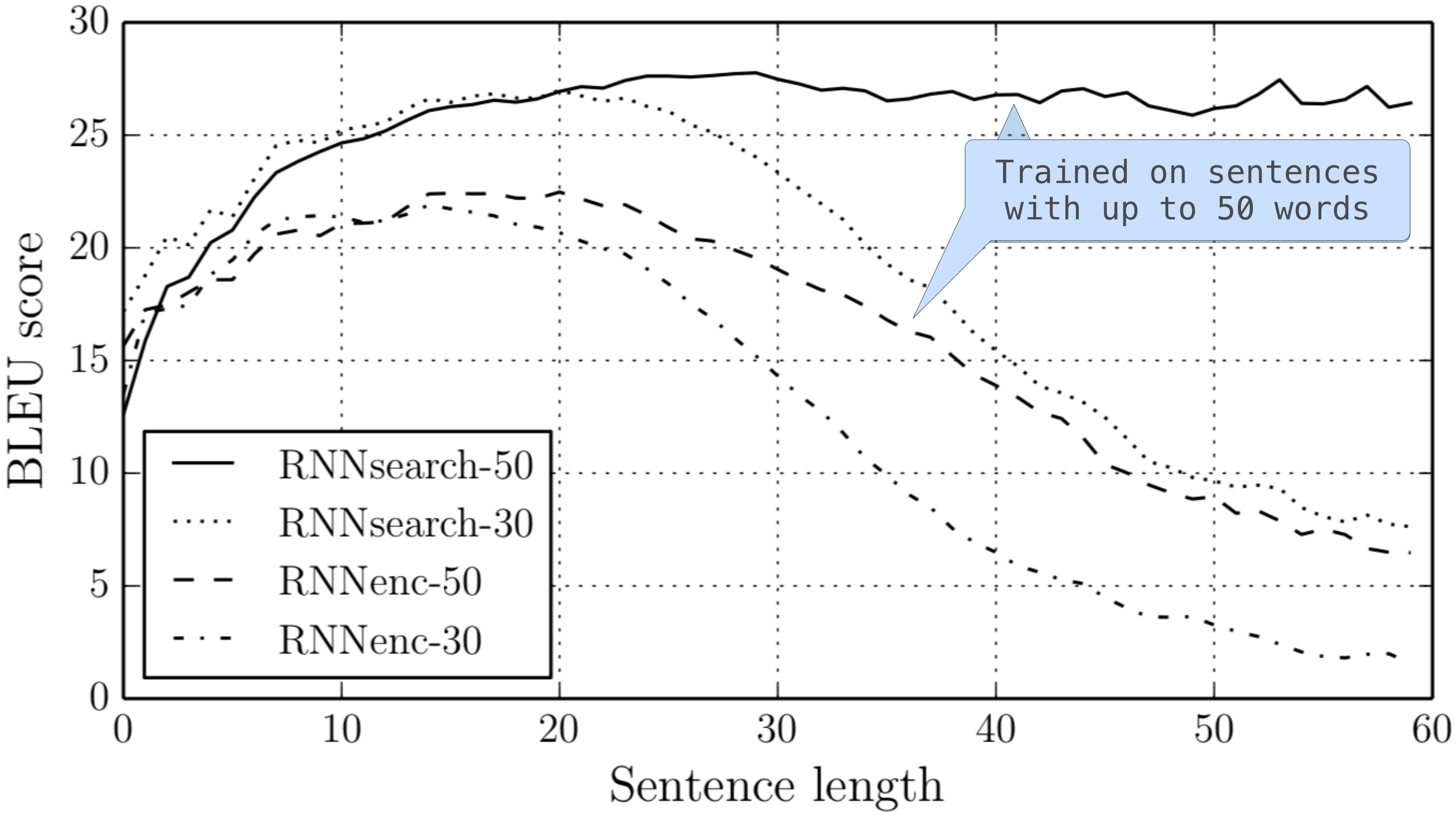
Google's correction (2016): $\dfrac{s(e)}{\frac{(5+m)^\alpha}{(5+1)^\alpha}}$

Word reward: $s(e) + \gamma m$

(Stahlberg & Byrne, 2019) On NMT Search Errors and Model Errors: Cat Got Your Tongue?
(Murray & Chiang, 2018) Correcting Length Bias in Neural Machine Translation

Attention

Impact of Attention on Long Sequence Generation



Trained on sentences with up to 50 words

(Badhanau et al., 2015) Neural Machine Translation by Jointly Learning to Align and Translate

Conditional Gated Recurrent Unit with Attention

$$\mathbf{s}_j = \mathrm{cGRU}_{\mathrm{att}}\left(\mathbf{s}_{j-1}, y_{j-1}, \mathrm{C}\right)$$

Architecture for the top research system in WMT16 and WMT17 (Univ. Edinburgh)

$$\mathbf{s}'_j = (1 - \mathbf{z}'_j) \odot \underline{\mathbf{s}}'_j + \mathbf{z}'_j \odot \mathbf{s}_{j-1}$$

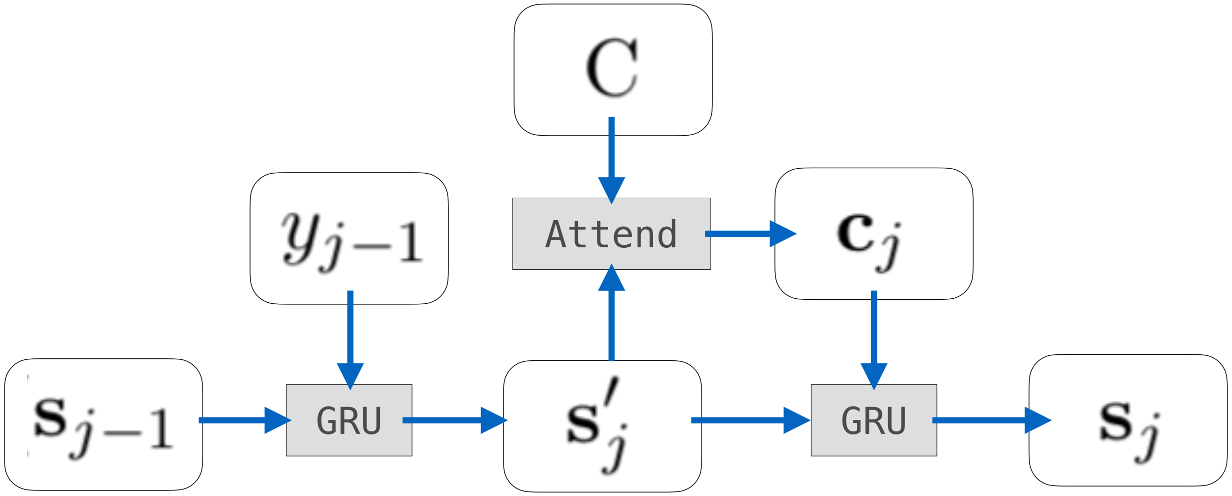
$$\underline{\mathbf{s}}'_j = \tanh\left(\mathbf{W}'\mathbf{E}[y_{j-1}] + \mathbf{r}'_j \odot (\mathbf{U}'\mathbf{s}_{j-1})\right),$$

$$\mathbf{r}'_j = \sigma\left(\mathbf{W}'_r\mathbf{E}[y_{j-1}] + \mathbf{U}'_r\mathbf{s}_{j-1}\right),$$

$$\mathbf{z}'_j = \sigma\left(\mathbf{W}'_z\mathbf{E}[y_{j-1}] + \mathbf{U}'_z\mathbf{s}_{j-1}\right),$$

Reset gate masks the previous state's projection within the nonlinear forward step

Update gate mixes the output of the forward step with the previous state



(Firat and Cho, 2016) DL4MT-Tutorial: Conditional Gated Recurrent Unit with Attention Mechanism

Conditional Gated Recurrent Unit with Attention

$$\mathbf{s}_j = \mathrm{cGRU}_{\mathrm{att}}\left(\mathbf{s}_{j-1}, y_{j-1}, \mathrm{C}\right)$$

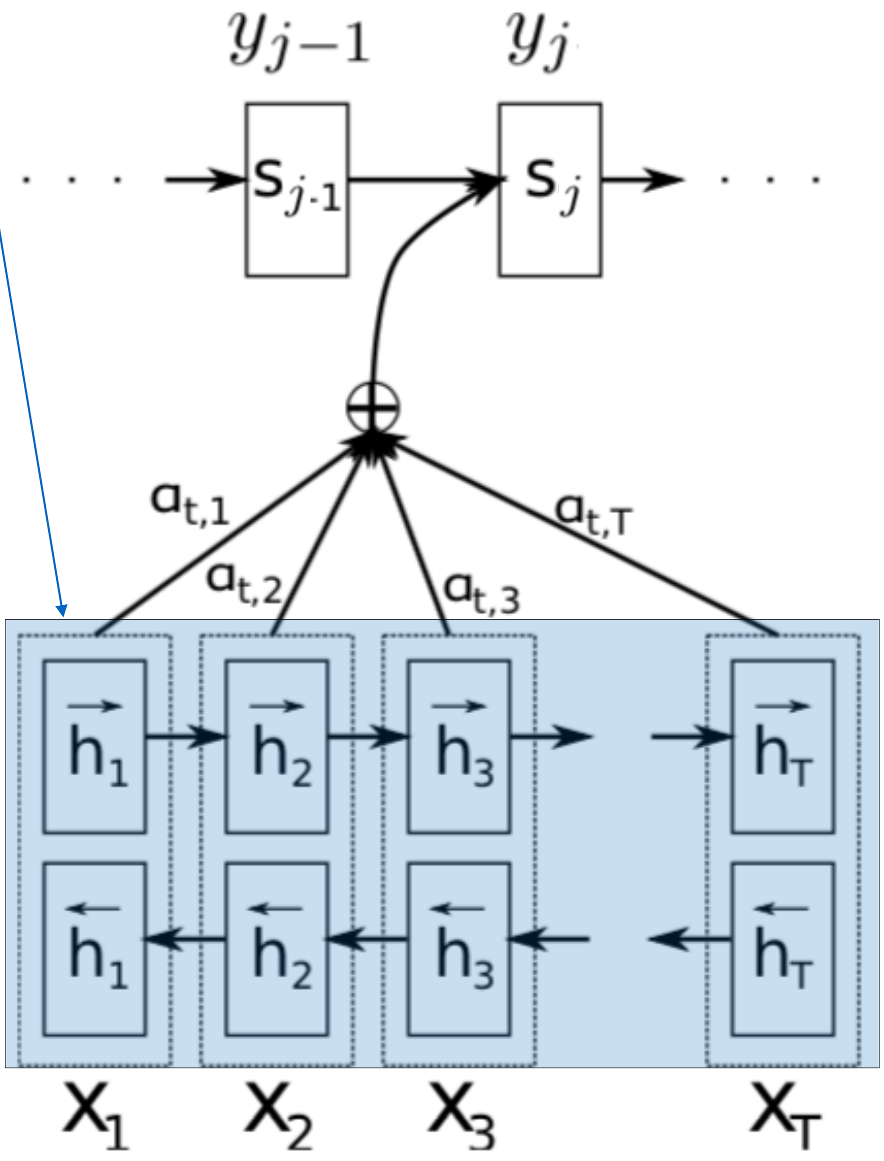
$$\mathbf{s}'_j = (1 - \mathbf{z}'_j) \odot \underset{\mathbf{E}[y_{j-1}]}{\underline{\mathbf{s}}'_j} + \mathbf{z}'_j \odot \mathbf{s}_{j-1}$$

$$\mathbf{c}_j = \mathrm{ATT}\left(\mathrm{C}, \mathbf{s}'_j\right) = \sum_i^{T_x} \alpha_{ij} \mathbf{h}_i,$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{Tx} \exp(e_{kj})},$$

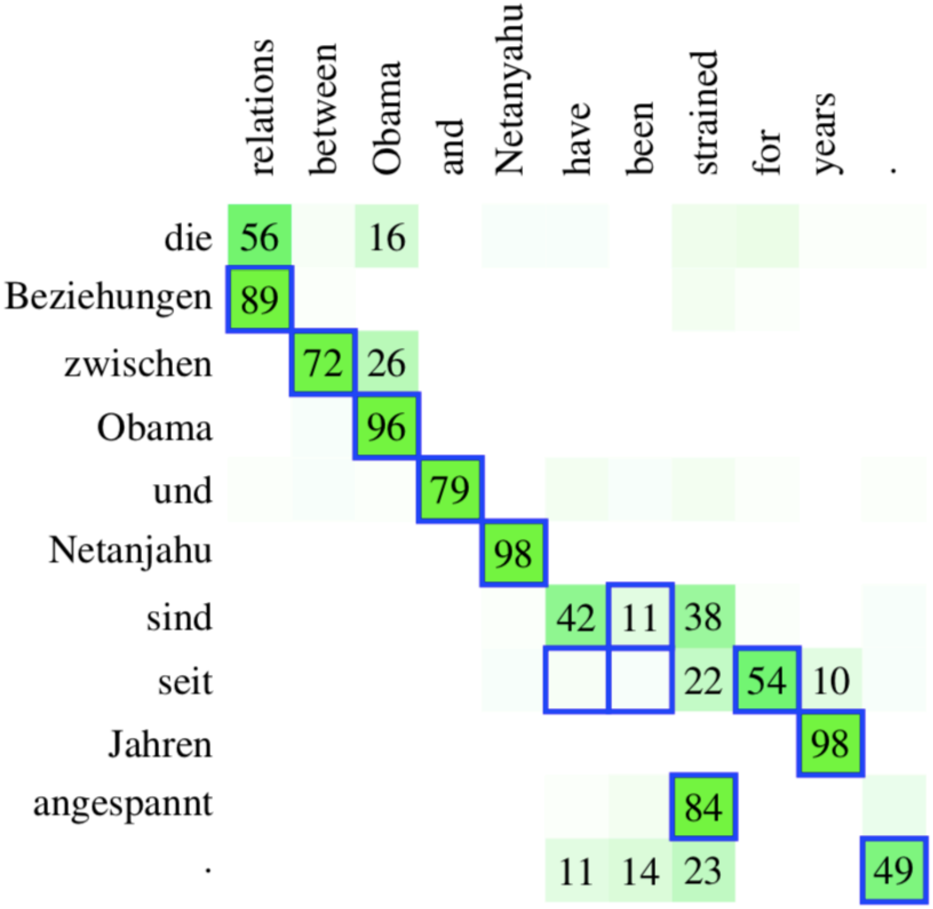
$$e_{ij} = \mathbf{v}_a^{\mathsf{T}} \tanh\left(\mathbf{U}_a \mathbf{s}'_j + \mathbf{W}_a \mathbf{h}_i\right)$$

$$\mathbf{s}_j = (1 - \mathbf{z}_j) \odot \underset{\mathbf{c}_j}{\underline{\mathbf{s}}_j} + \mathbf{z}_j \odot \mathbf{s}'_j$$



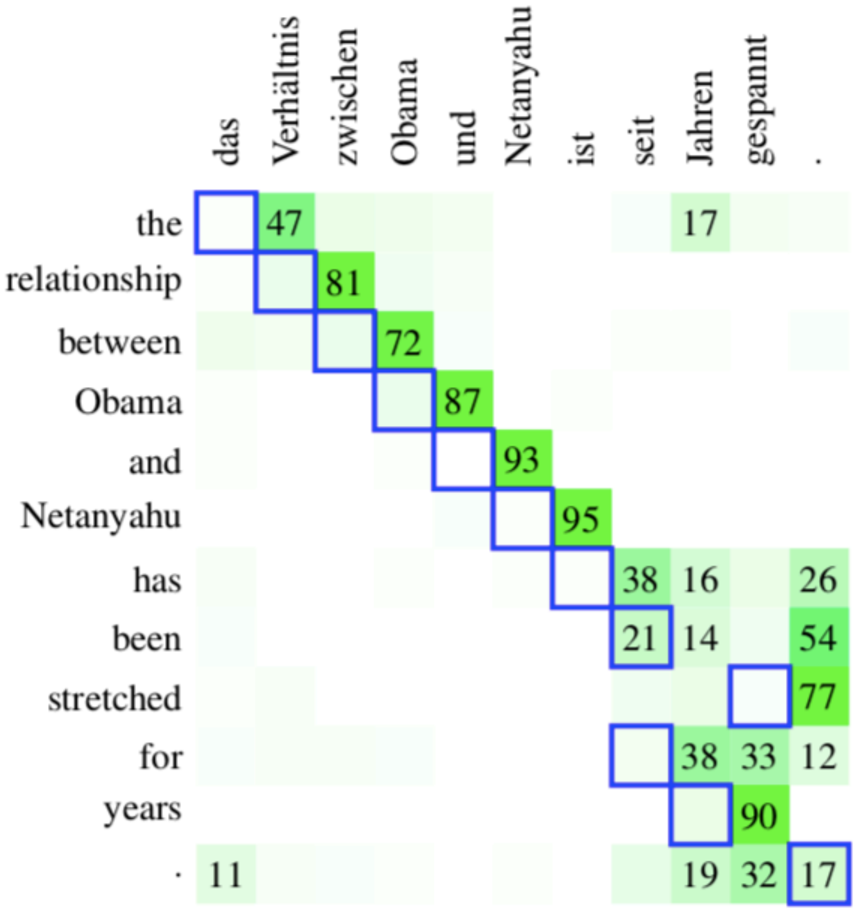
Attention Activations

Attention activations above 0.1



	relations	between	Obama	and	Netanyahu	have	been	strained	for	years	.
die	56		16								
Beziehungen	89										
zwischen		72	26								
Obama			96								
und				79							
Netanjahu					98						
sind						42	11	38			
seit								22	54	10	
Jahren										98	
angespannt								84			
.						11	14	23			49

English–German



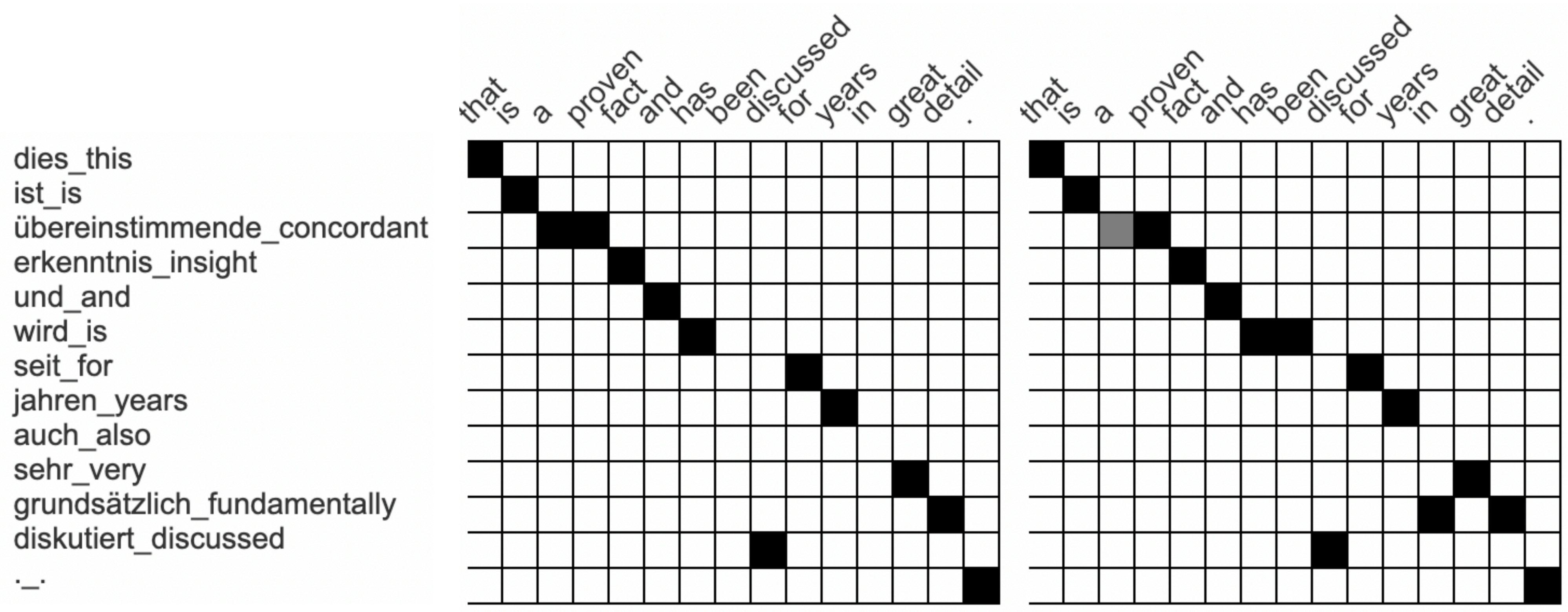
	das	Verhältnis	zwischen	Obama	und	Netanyahu	ist	seit	Jahren	gespannt	.
the		47							17		
relationship			81								
between				72							
Obama					87						
and						93					
Netanyahu							95				
has								38	16		26
been								21	14		54
stretched											77
for									38	33	12
years										90	
.	11								19	32	17

German–English

(Koehn & Knowles 2017) Six Challenges for Neural Machine Translation

Better Alignments from Attention Activations

```
Ideas:
(1) Find attention activations that would have led to correct word choice.
(2) Choose target words conditioned only on source context.
(3) Find attention activations that are good for both e->f and f->e.
```



(b) Bidir. Optimization

(c) Gold Alignments

(Zenkel et al., 2020) End-to-End Neural Word Alignment Outperforms GIZA++

Transformer Architecture

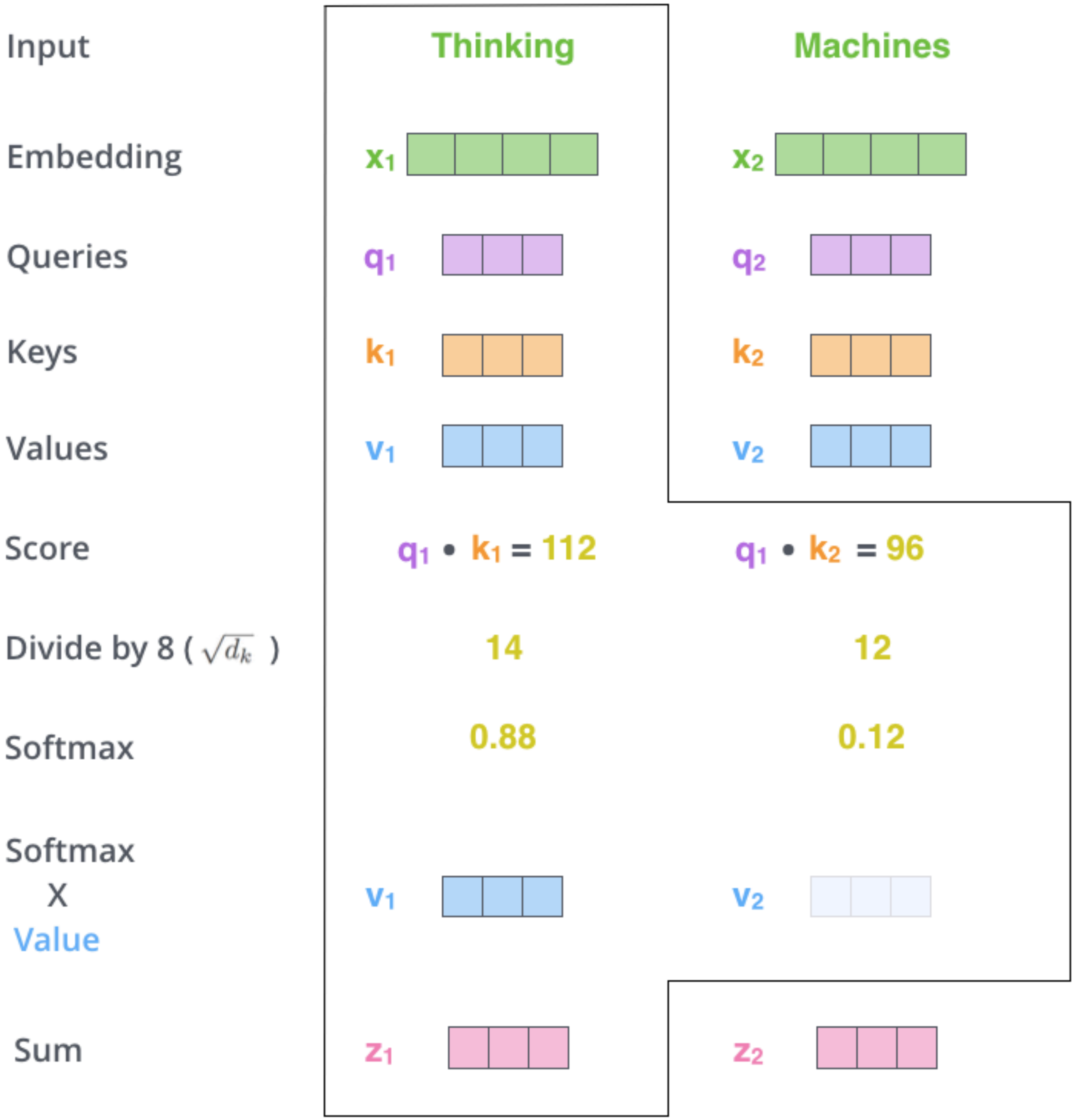
Transformer

In lieu of an RNN, use attention.

High throughput & expressivity: compute queries, keys and values as (different) linear transformations of the input.

Attention weights are queries • keys; outputs are sums of weighted values.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

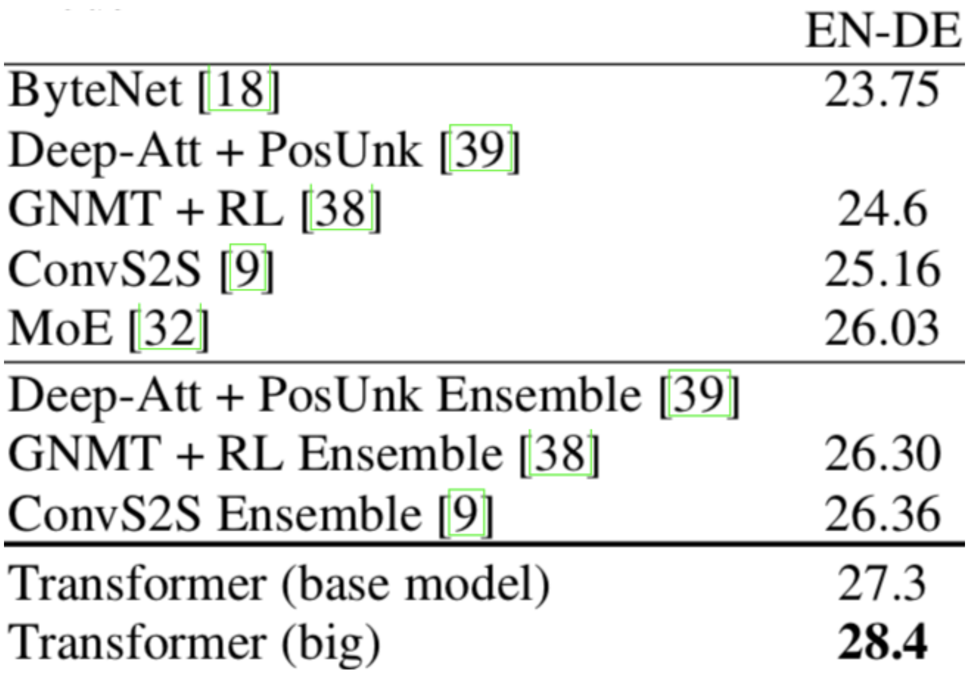


(Vaswani et al., 2017) Attention is All You Need
Figure: http://jalammar.github.io/illustrated-transformer/

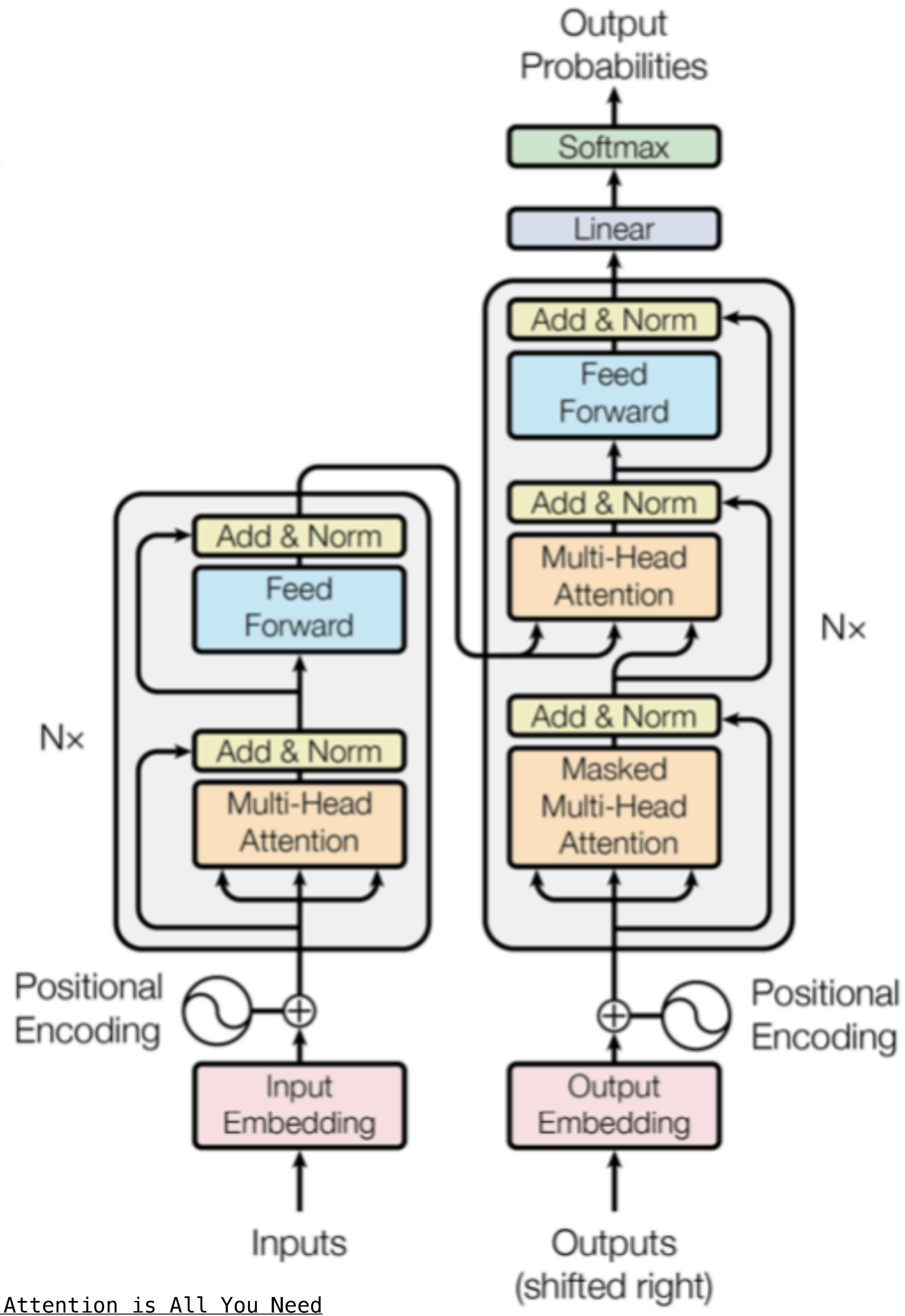
Transformer Architecture

- Layer normalization ("Add & Norm" cells) helps with RNN+attention architectures as well.

- Positional encodings can be learned or based on a formula that makes it easy to represent distance.



	EN-DE
ByteNet [18]	23.75
Deep-Att + PosUnk [39]	
GNMT + RL [38]	24.6
ConvS2S [9]	25.16
MoE [32]	26.03
Deep-Att + PosUnk Ensemble [39]	
GNMT + RL Ensemble [38]	26.30
ConvS2S Ensemble [9]	26.36
Transformer (base model)	27.3
Transformer (big)	**28.4**



(Vaswani et al., 2017) Attention is All You Need

Some Transformer Concerns

Problem: Bag-of-words representation of the input.
Remedy: Position embeddings are added to the word embeddings.

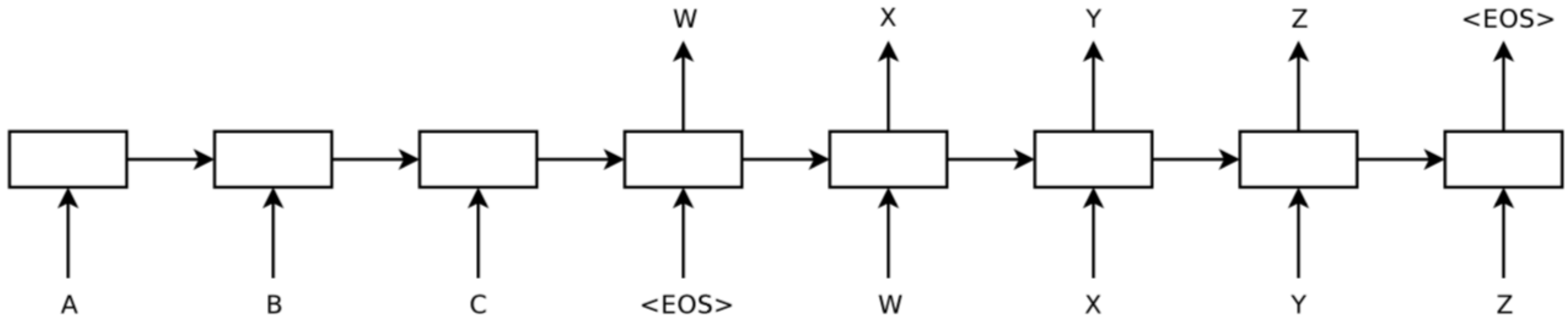
Problem: During generation, can't attend to future words.
Remedy: Masked training that zeroes attention to future words.

Problem: Deep networks needed to integrated lots of context.
Remedies: Residual connections and multi-head attention.

Problem: Optimization is hard.
Remedies: Large mini-batch sizes and layer normalization.

Training Loss Function

Teacher forcing: During training, only use the predictions of the model for the loss, not the input.



Label smoothing: Update toward a distribution in which

- 0.9 probability is assigned to the observed word, and
- 0.1 probability is divided uniformly among all other words.

Sequence–level loss has been explored, but (so far) abandoned.

Training Data

Subwords

The sequence of symbols that are embedded should be common enough that an embedding can be estimated robustly for each, and all symbols have been observed during training.

Solution 1: Symbols are words with rare words replaced by UNK.

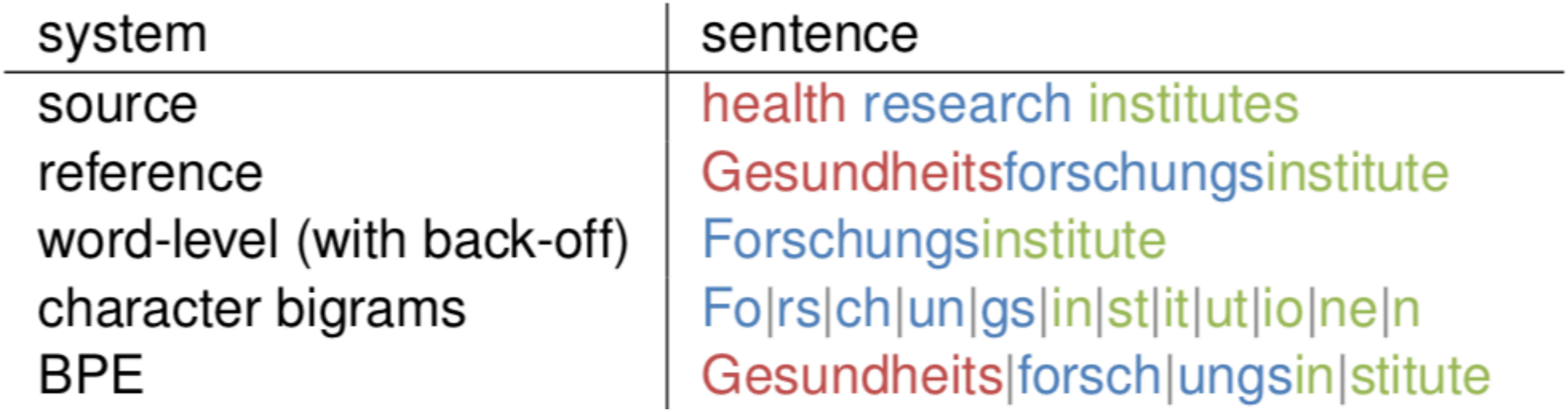
- Replacing UNK in the output is a new problem (like alignment).
- UNK in the input loses all information that might have been relevant from the rare input word (e.g., tense, length, POS).

Solution 2: Symbols are subwords.

- Byte-Pair Encoding is the most common approach.
- Other techniques that find common subwords aren't reliably much better (but are somewhat more complicated).
- Training on many sampled subword decompositions can improve out-of-domain translations.

(Sennrich et al., 2016) Neural Machine Translation of Rare Words with Subword Units
(Kudo, 2018) Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates

BPE Example



system	sentence
source	health research institutes
reference	Gesundheitsforschungsinstitute
word-level (with back-off)	Forschungsinstitute
character bigrams	Fo|rs|ch|un|gs|in|st|it|ut|io|ne|n
BPE	Gesundheits|forsch|ungsin|stitute

Example from Rico Sennrich

Initialize: Split each word into symbols that are individual characters

Repeat: Convert the most frequent symbol bigram into a new symbol

```
vocab = {'l o w </w>' : 5,
         'l o w e r </w>' : 2,
         'n e w e s t </w>': 6,
         'w i d e s t </w>': 3}
```

```
('e', 's') appears 9 times and is now 'es'
('es', 't') appears 9 times and is now 'est'
('est', '</w>') appears 9 times and is now 'est</w>'
('l', 'o') appears 7 times and is now 'lo'
('lo', 'w') appears 7 times and is now 'low'
('n', 'e') appears 6 times and is now 'ne'
('ne', 'w') appears 6 times and is now 'new'
('new', 'est</w>') appears 6 times and is now 'newest</w>'
('low', '</w>') appears 5 times and is now 'low</w>'
('w', 'i') appears 3 times and is now 'wi'
```

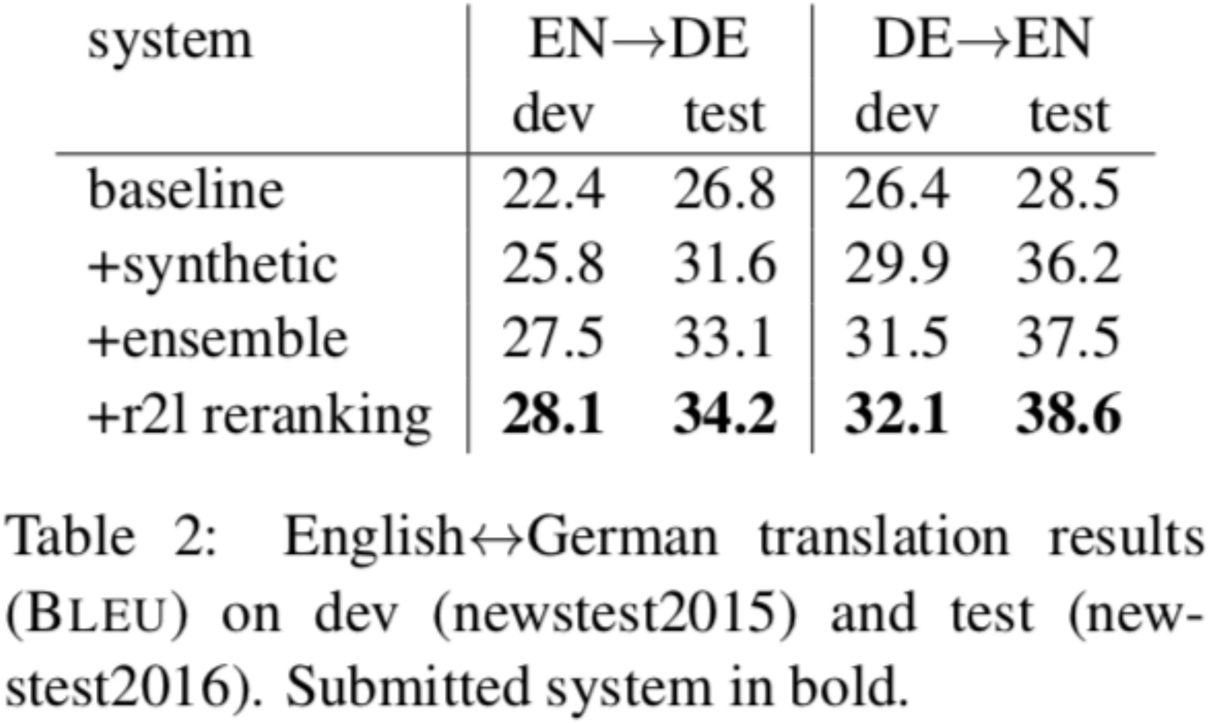
```
{'low</w>': 5, 'low e r </w>': 2, 'newest</w>': 6, 'wi d est</w>': 3}
```

(Sennrich et al., 2016) Neural Machine Translation of Rare Words with Subword Units

Back Translations

Synthesize an *en–de* parallel corpus by using a *de–en* system to translate monolingual *de* sentences.

- Better generating systems don't seem to matter much.
- Can help even if the *de* sentences are already in an existing *en–de* parallel corpus!

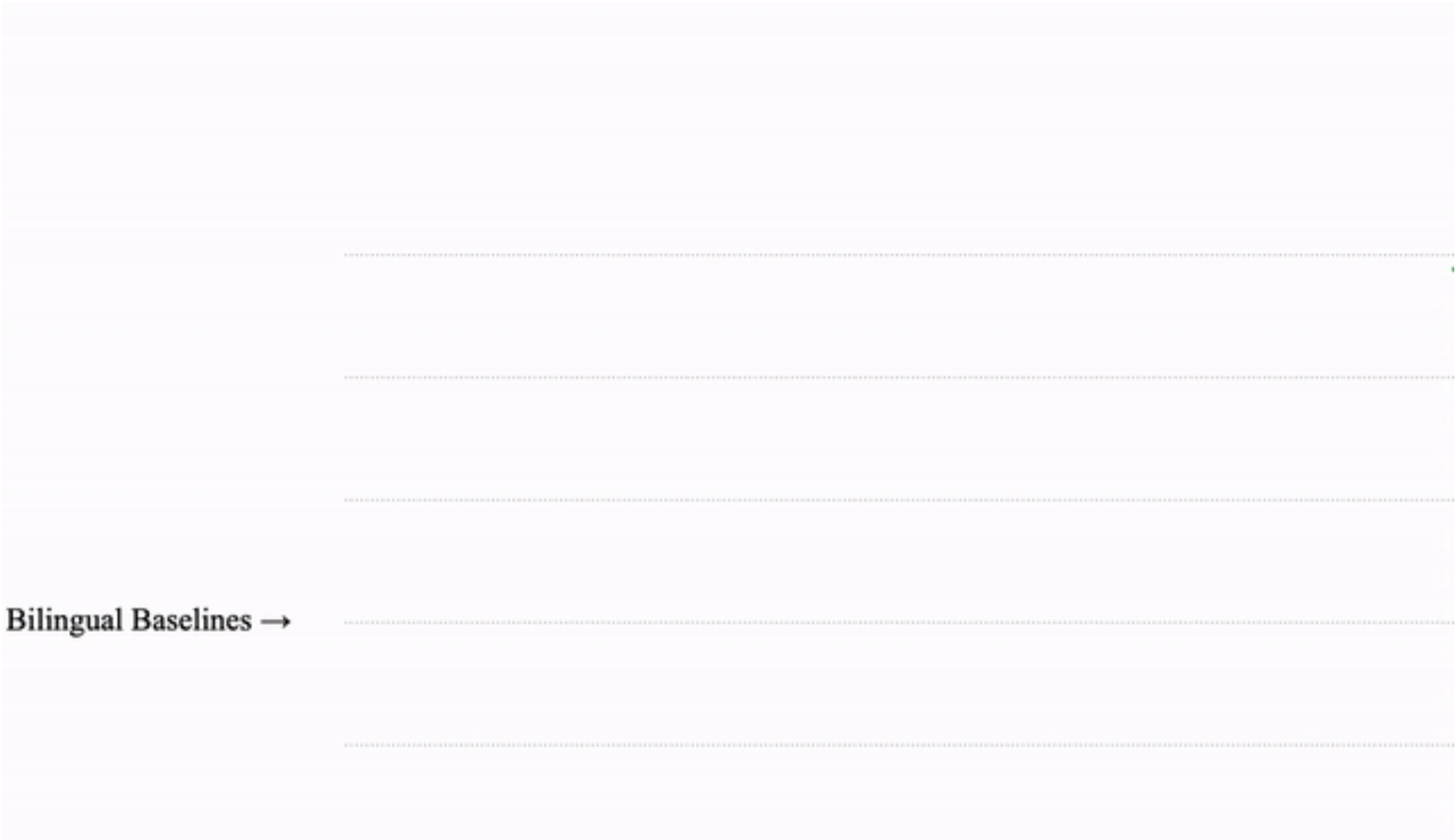


system	EN→DE		DE→EN	
	dev	test	dev	test
baseline	22.4	26.8	26.4	28.5
+synthetic	25.8	31.6	29.9	36.2
+ensemble	27.5	33.1	31.5	37.5
+r2l reranking	**28.1**	**34.2**	**32.1**	**38.6**

Table 2: English↔German translation results (BLEU) on dev (newstest2015) and test (newstest2016). Submitted system in bold.

(Sennrich et al., 2015) Improving Neural Machine Translation Models with Monolingual Data
(Sennrich et al., 2016) Edinburgh Neural Machine Translation Systems for WMT 16

Multilingual Neural Machine Translations



Translation quality improvement of a single massively multilingual model as we increase the capacity (number of parameters) compared to 103 individual bilingual baselines.

https://ai.googleblog.com/2019/10/exploring-massively-multilingual.html

First Large-Scale Massively Multilingual Experiment

Trained on Google-internal corpora for 103 languages.

1M or fewer sentence pairs per language; 95M examples total.

Evaluated on "10 languages from different typological families: Semitic – Arabic (Ar), Hebrew (He), Romance – Galician (Gl), Italian (It), Romanian (Ro), Germanic – German (De), Dutch (Nl), Slavic – Belarusian (Be), Slovak (Sk) and Turkic – Azerbaijani (Az) and Turkish (Tr)."

Model architecture: Sequence-to-sequence Transformer with a target-language indicator token prepended to each source sentence to enable multiple output languages.

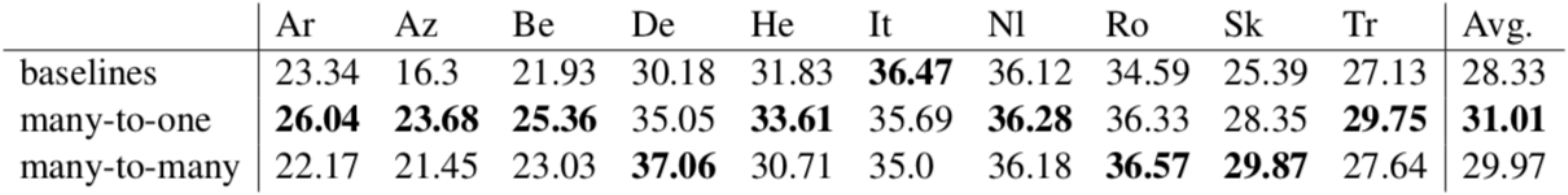
- 6 layer encoder & decoder; 1024/8192 layer sizes; 16 heads
- 473 million trainable model parameters
- 64k subwords shared across 103 languages

Baseline: Same model architecture trained on bilingual examples.

Roee Aharoni, Melvin Johnson, Orhan Firat, 2019, "Massively Multilingual Neural Machine Translation"

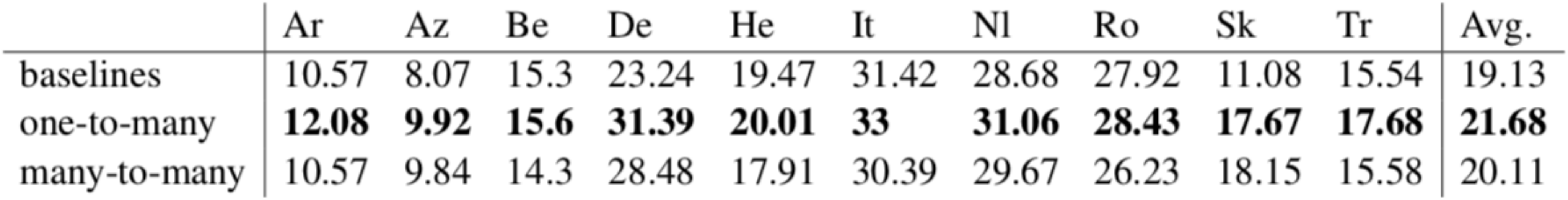
First Large-Scale Massively Multilingual Experiment

Evaluated on "10 languages from different typological families: Semitic – Arabic (Ar), Hebrew (He), Romance – Galician (Gl), Italian (It), Romanian (Ro), Germanic – German (De), Dutch (Nl), Slavic – Belarusian (Be), Slovak (Sk) and Turkic – Azerbaijani (Az) and Turkish (Tr)."



	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	25.39	27.13	28.33
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

Table 5: X→En test BLEU on the 103-language corpus



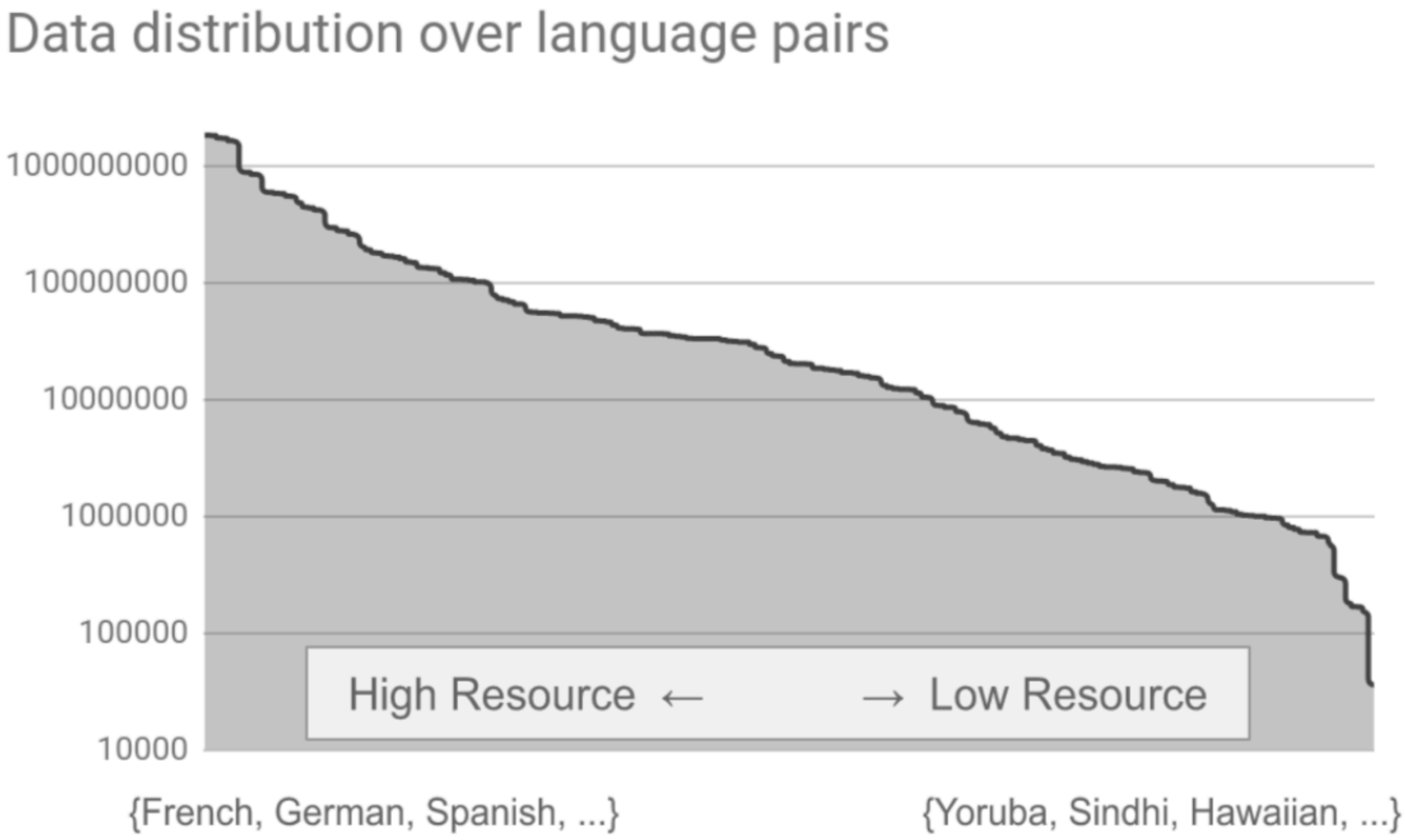
	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

Table 6: En→X test BLEU on the 103-language corpus

Roee Aharoni, Melvin Johnson, Orhan Firat, 2019, "Massively Multilingual Neural Machine Translation"

Full-Scale Massively Multilingual Experiment

25 billion parallel sentences in 103 languages.



Arivazhagan, Bapna, Firat, et al. (2019) "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges"

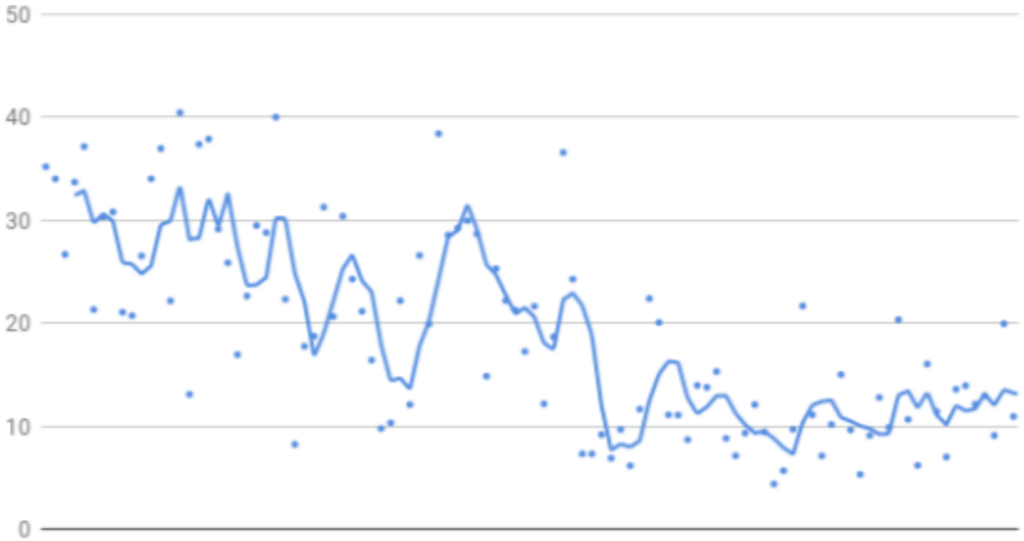
Full-Scale Massively Multilingual Experiment

25 billion parallel sentences in 103 languages.

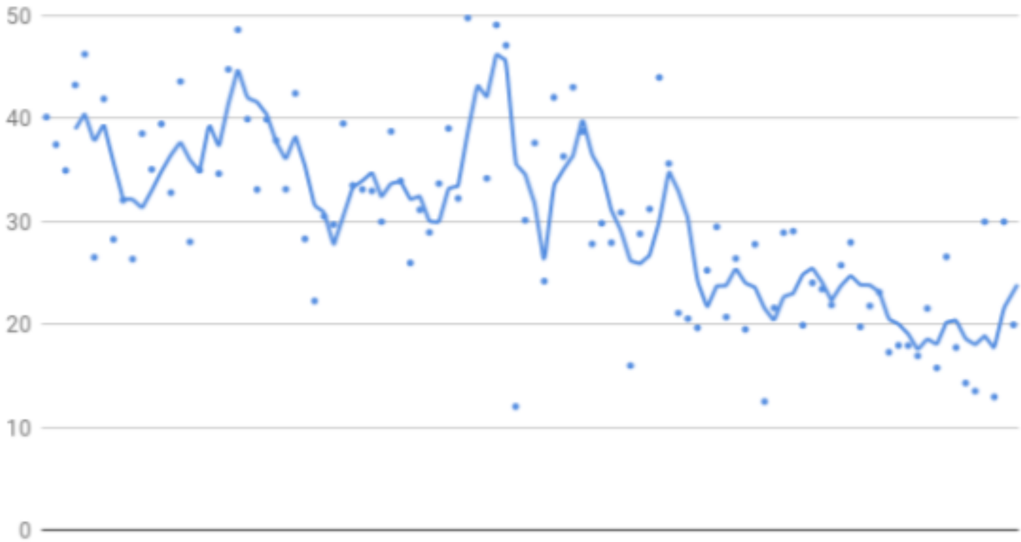
Baselines: Bilingual Transformer Big w/ 32k Vocab (~375M params) for most languages; Transformer Base for low-resource languages.

Evaluation: Constructed multi-way dataset of 3k–5k translated English sentences.

Bilingual En→Any translation performance vs dataset size



Bilingual Any→En translation performance vs dataset size



"Performance on individual language pairs is reported using dots and a trailing average is used to show the trend."

Arivazhagan, Bapna, Firat, et al. (2019) "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges"

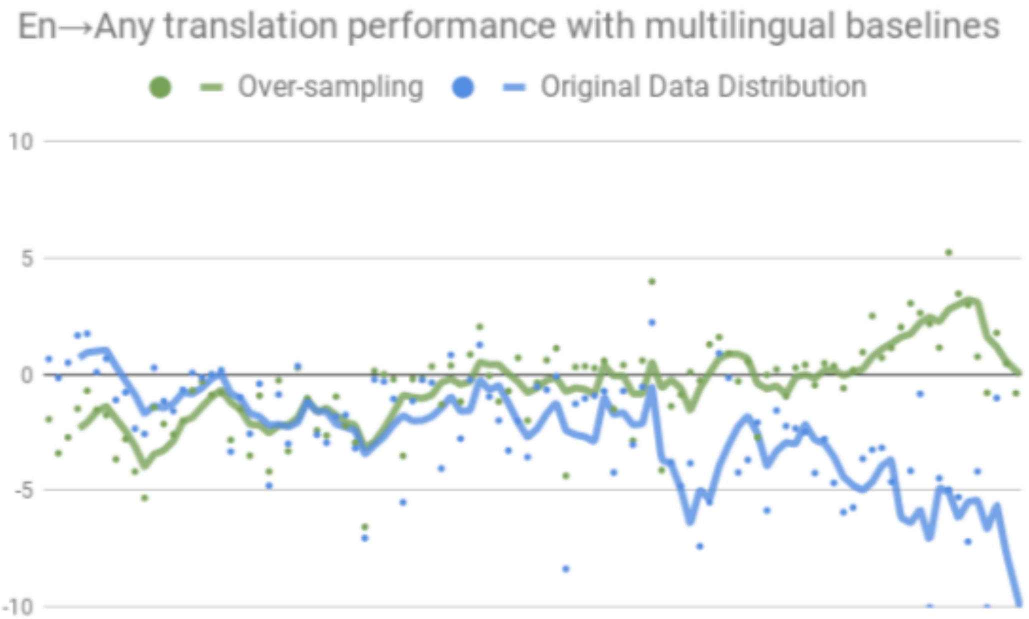
Full-Scale Massively Multilingual Experiment

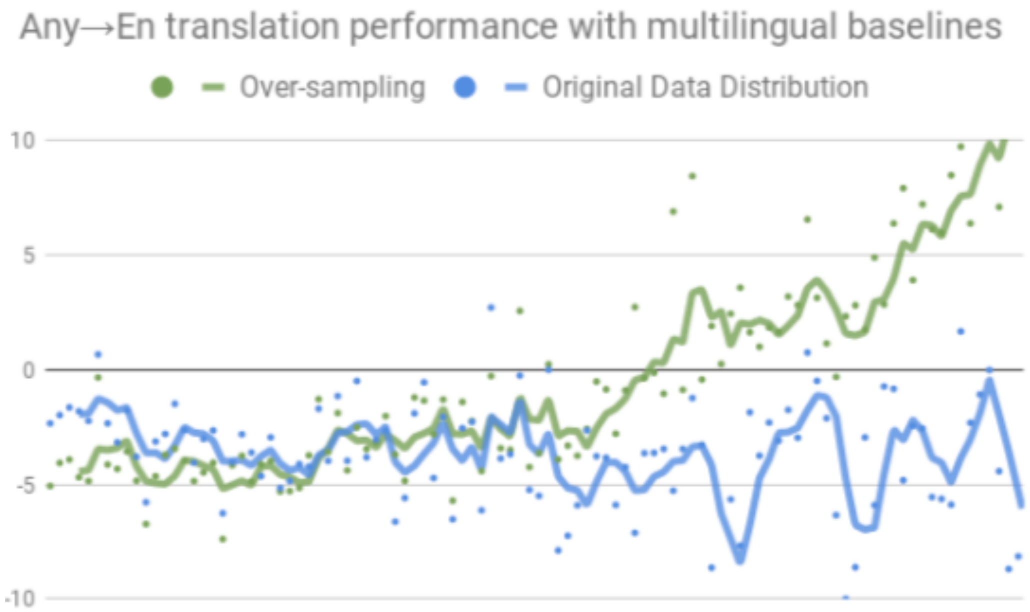
25 billion parallel sentences in 103 languages.

Baselines: Bilingual Transformer Big w/ 32k Vocab (~375M params) for most languages; Transformer Base for low-resource languages.

Multilingual system: Transformer Big w/ 64k Vocab trained 2 ways:

- "All the available training data is combined as it is."
- "We over-sample (up-sample) low-resource languages so that they appear with equal probability in the combined dataset."





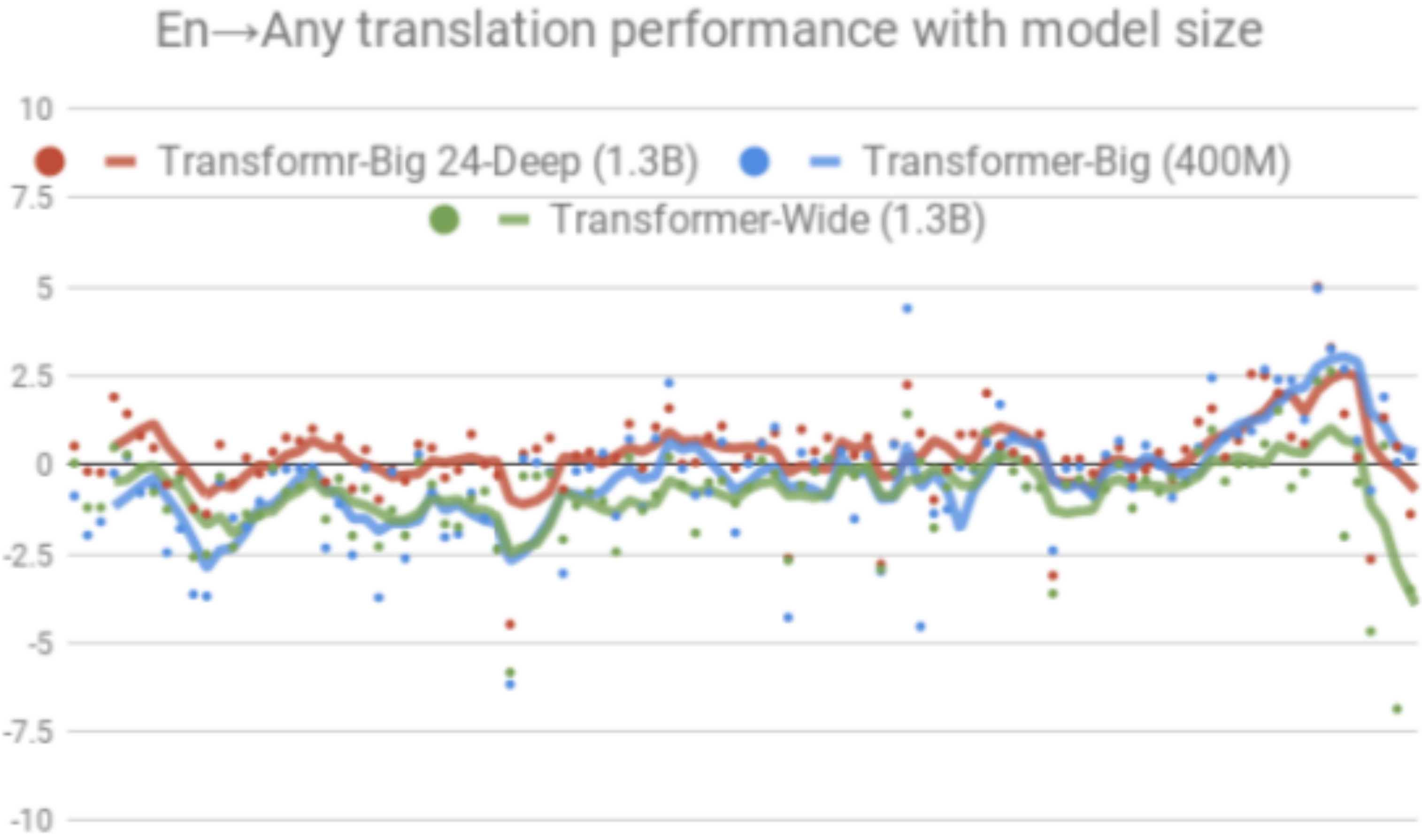
Arivazhagan, Bapna, Firat, et al. (2019) "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges"

Full-Scale Massively Multilingual Experiment

25 billion parallel sentences in 103 languages.

Baselines: Bilingual Transformer Big w/ 32k Vocab (~375M params) for most languages; Transformer Base for low-resource languages.

Multilingual systems: Transformers of varying sizes.



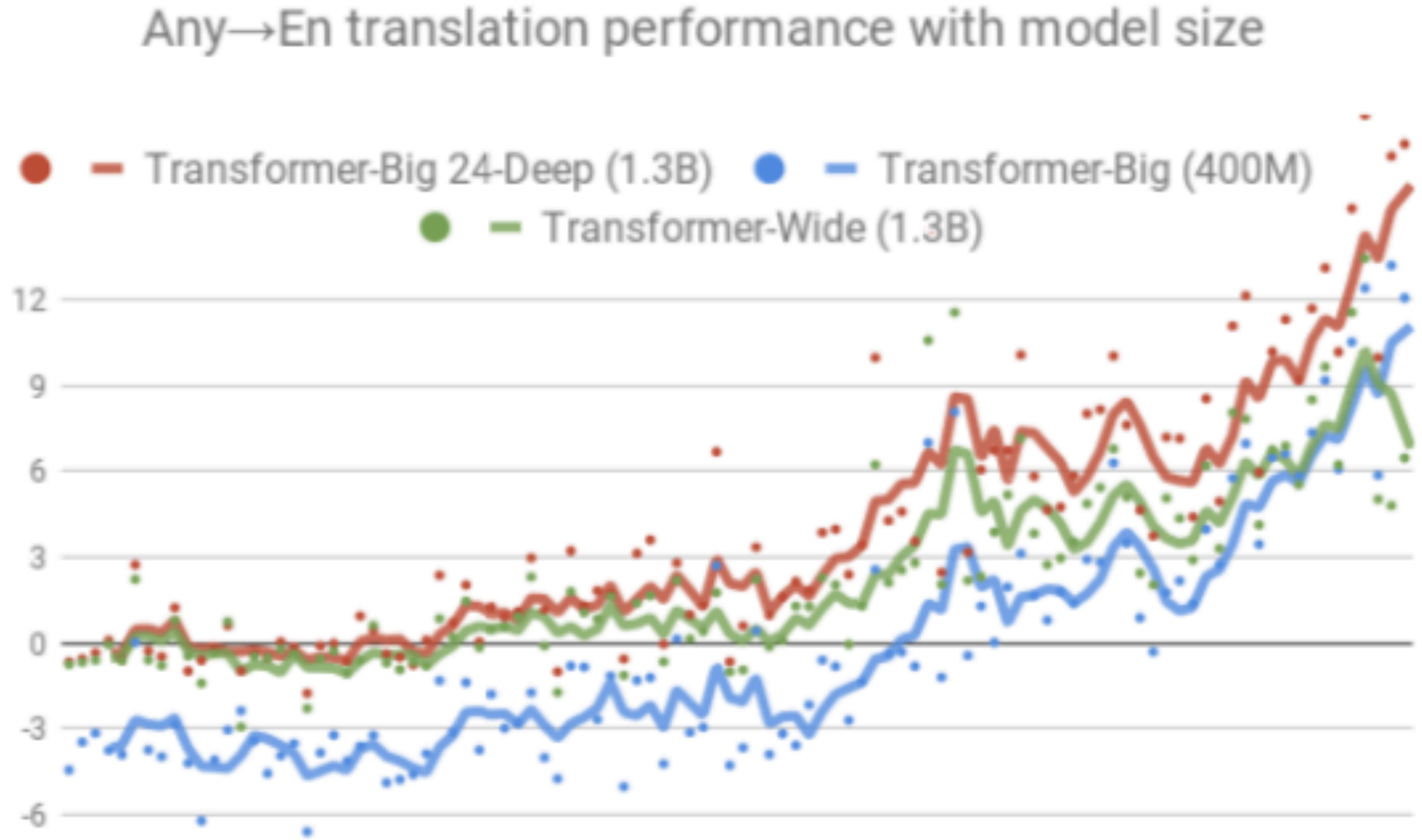
Arivazhagan, Bapna, Firat, et al. (2019) "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges"

Full-Scale Massively Multilingual Experiment

25 billion parallel sentences in 103 languages.

Baselines: Bilingual Transformer Big w/ 32k Vocab (~375M params) for most languages; Transformer Base for low-resource languages.

Multilingual systems: Transformers of varying sizes.



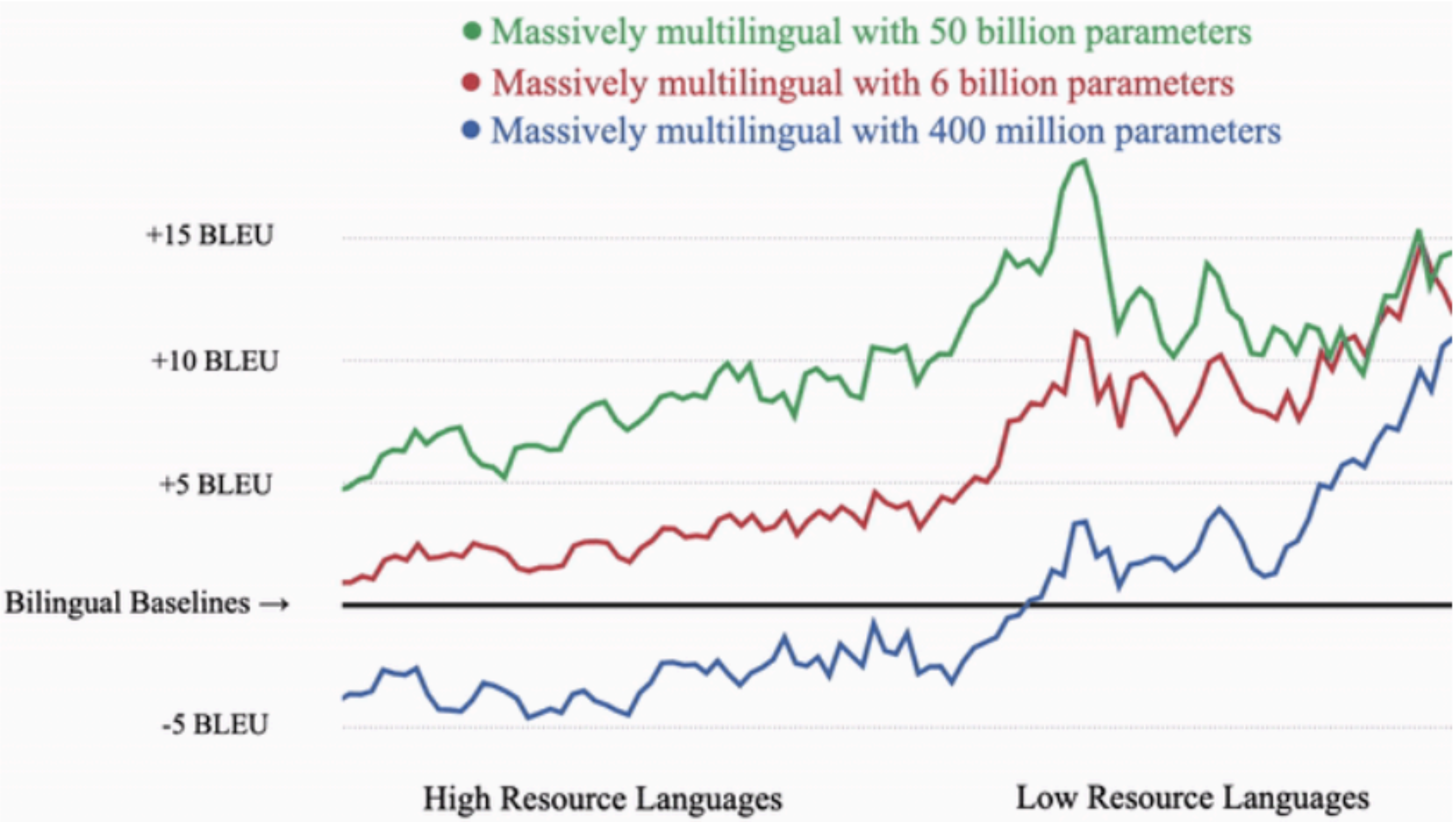
Arivazhagan, Bapna, Firat, et al. (2019) "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges"

Full-Scale Massively Multilingual Experiment

25 billion parallel sentences in 103 languages.

Baselines: Bilingual Transformer Big w/ 32k Vocab (~375M params) for most languages; Transformer Base for low-resource languages.

Multilingual systems: Transformers of varying sizes.



https://ai.googleblog.com/2019/10/exploring-massively-multilingual.html