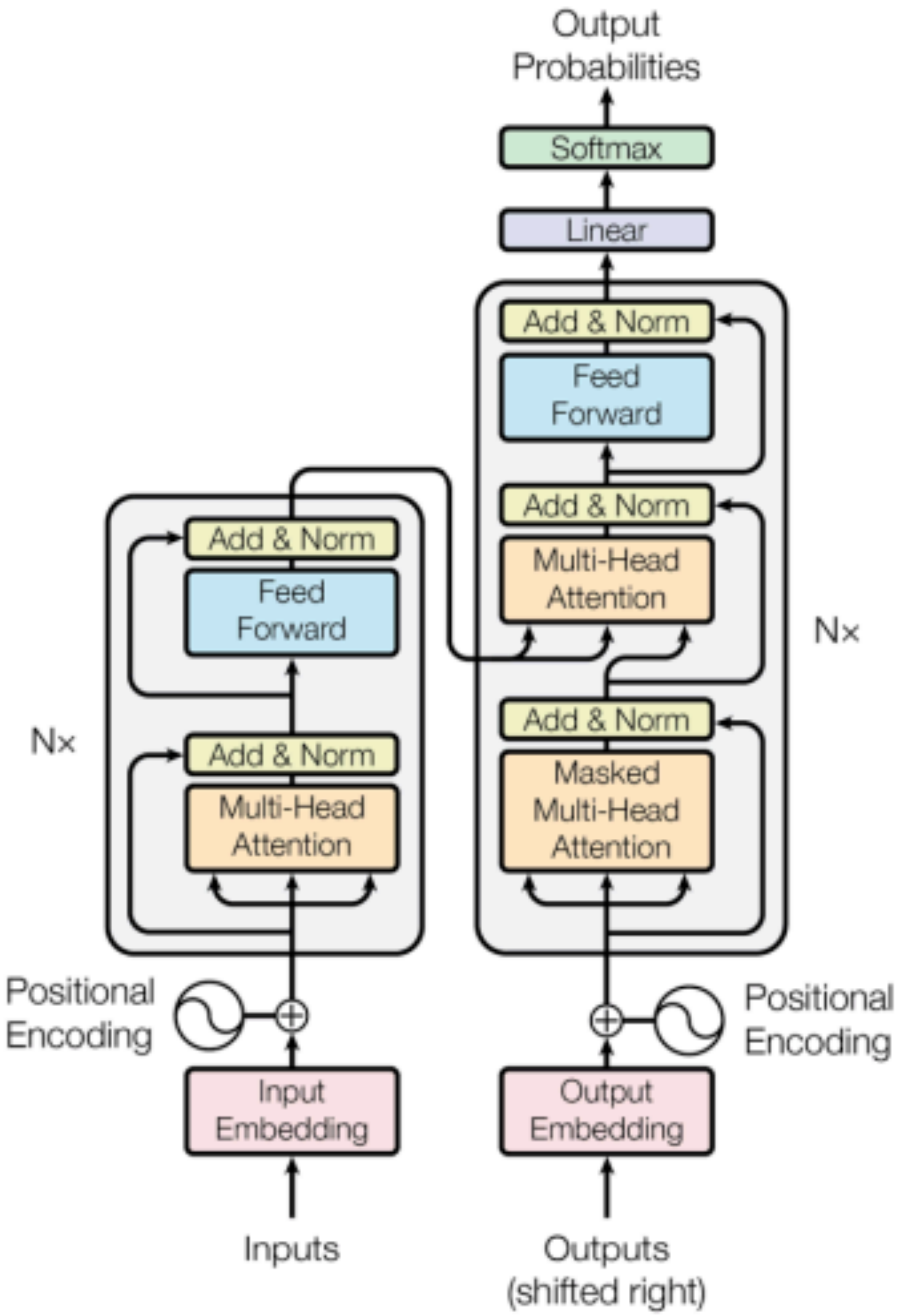
CSCI 544: Applied Natural Language Processing

Advances in Transformer & NMT

Xuezhe Ma (Max)

USC University of Southern California

Recap: Transformers

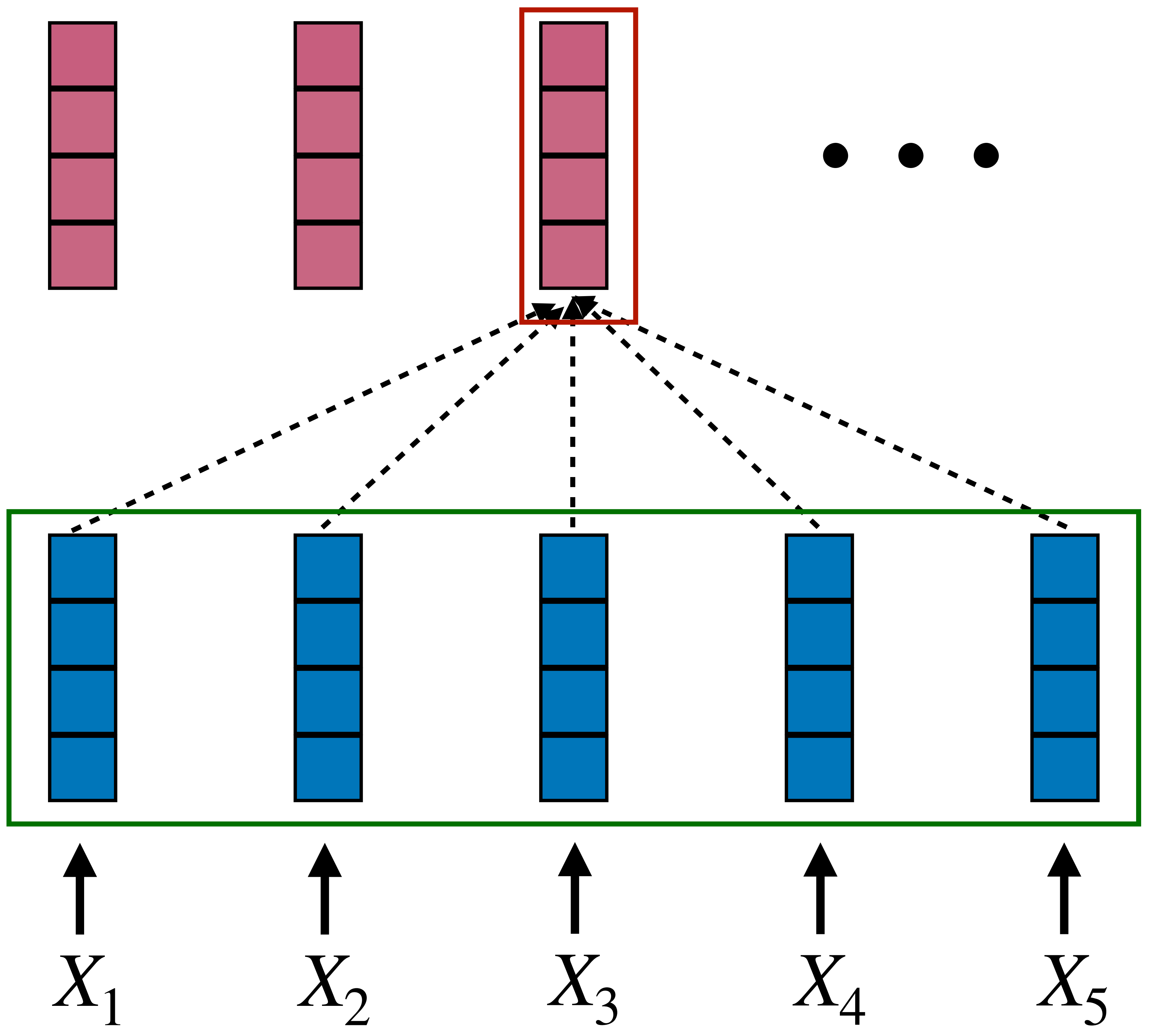


- Consists of an encoder and a decoder
- Originally proposed for neural machine translation and later adapted for almost all the NLP tasks
  - For example, BERT only uses the **encoder** of the Transformer architecture (next lecture)
- Both encoder and decoder consist of *N* layers
  - Each encoder layer has two sub-layers
  - Each decoder layer has three sublayers
  - Key innovation: **multi-head self-attention**

Vaswani et al., 2017: Attention Is All You Need

Recap: Self-Attention

- **Self-attention: attention within on single sequence**
  - Contexts and queries are drawn from the same source
- **Contextual information via self-attention**



- **Capturing long-distance dependencies**
- **No gradient vanishing**

Recap: Self-attention in equations

- A sequence of input vectors $x_1, \ldots, x_n \in \mathbb{R}^d$
- First, construct a set of queries, keys and values:

$$q_i = W_Q x_i,\ k_i = W_K x_i,\ v_i = W_V x_i$$

- Second, for each q_i, compute attention scores and attention distributions:

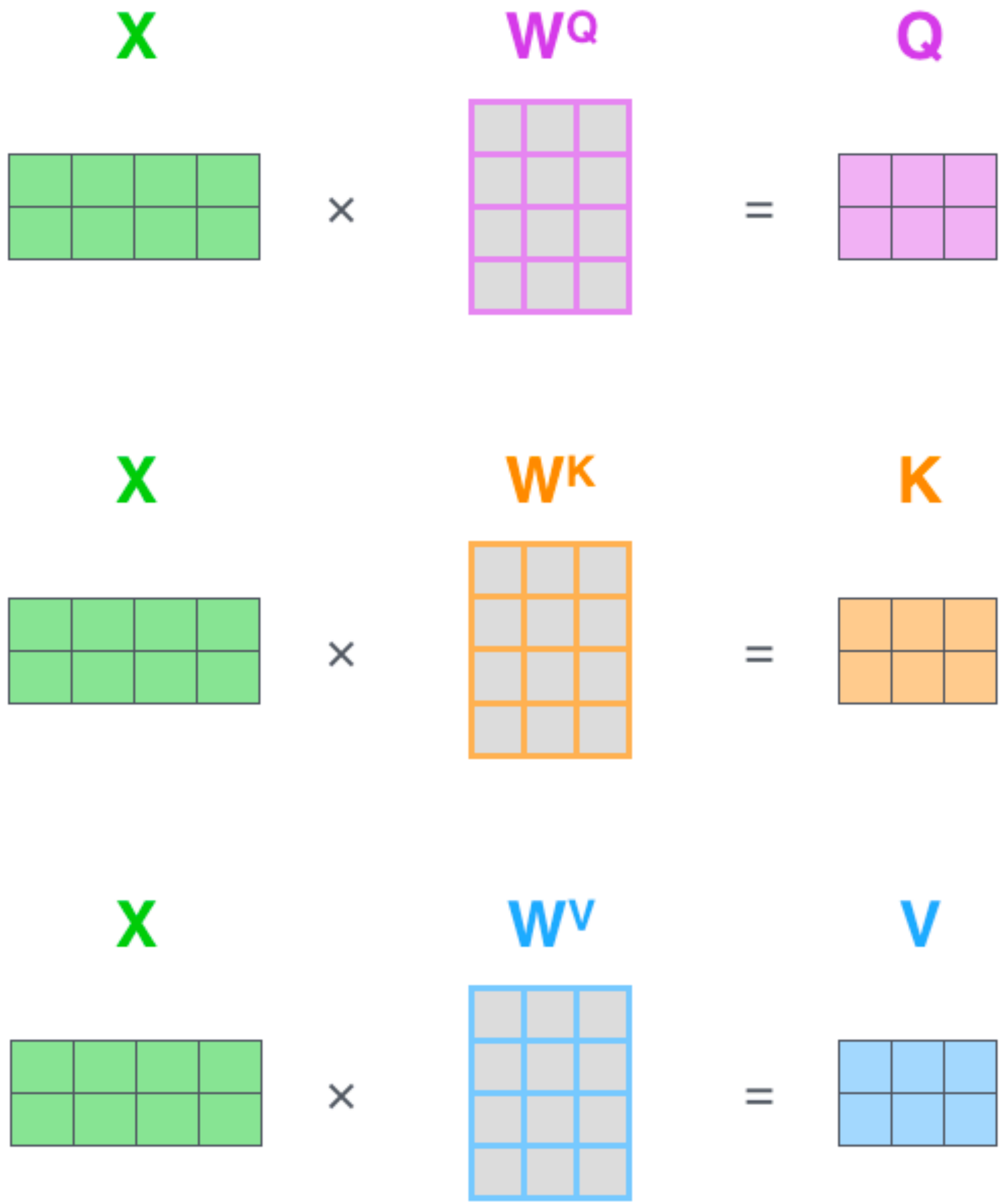
$$a_{i,j} = \text{softmax}(\frac{q_i^T k_j}{\sqrt{d}})$$

aka. "scaled dot product"

- Finally, compute the weighted sum:

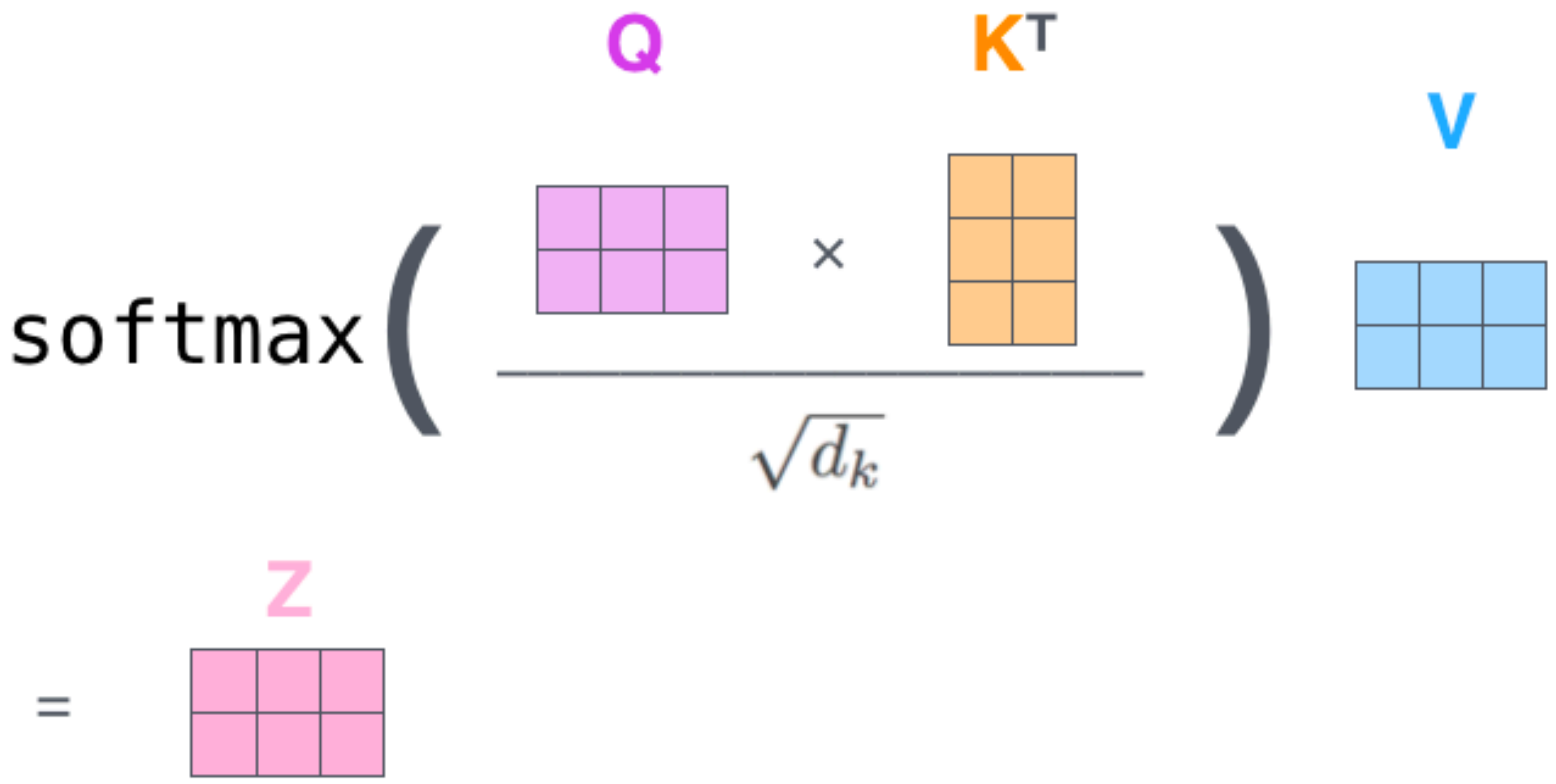
$$y_i = \sum_{j=1}^{n} a_{i,j} v_j$$

Self-attention: matrix notations

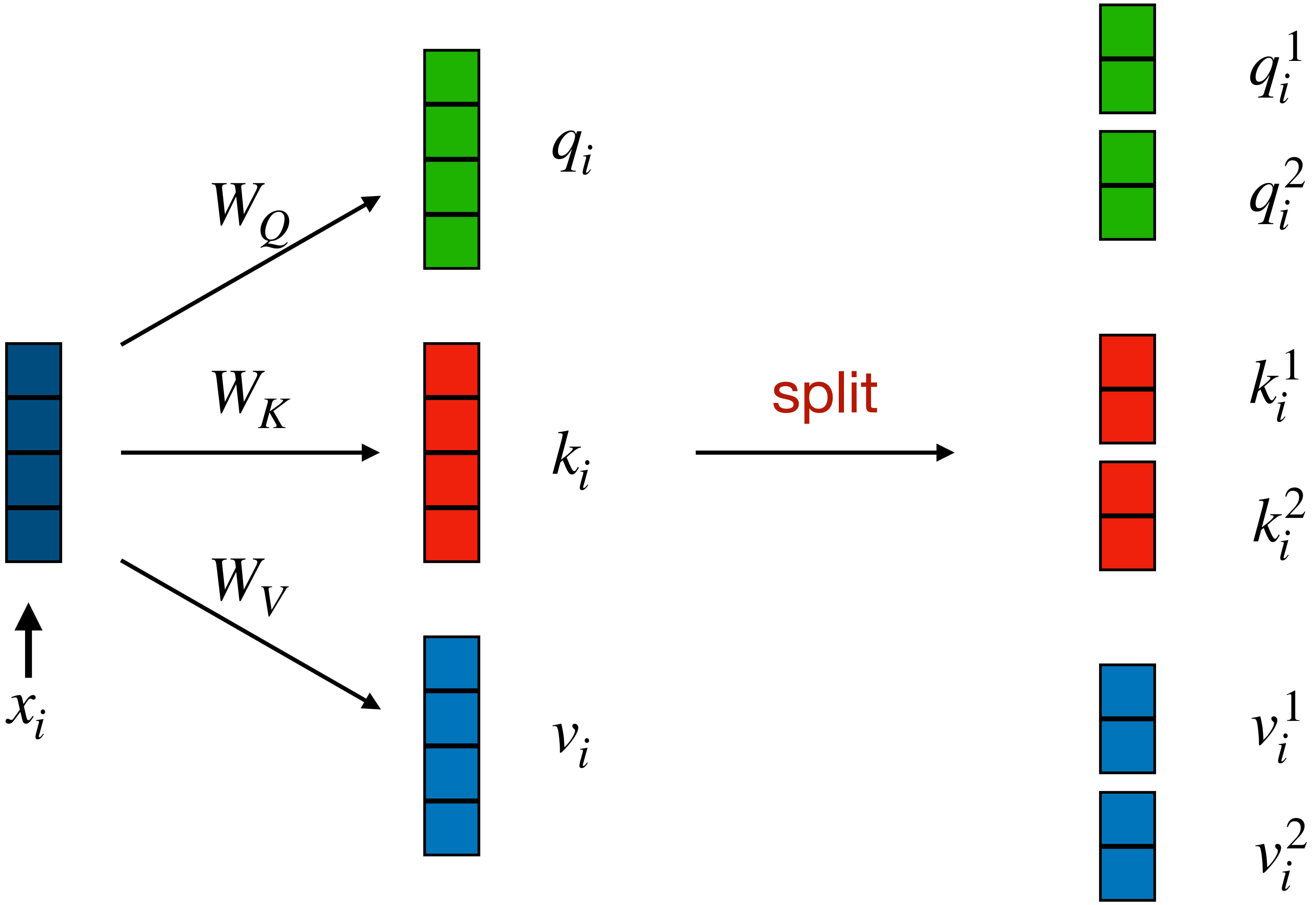


Self-attention: matrix notations

$$\boxed{\mathrm{attn}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d}})V}$$



Recap: Multi-head Attention



$$h_1 = \mathrm{attn}(Q_1, K_1, V_1) = \mathrm{softmax}\left(\frac{Q_1 K_1^T}{\sqrt{d/2}}\right)V_1$$

$$h_2 = \mathrm{attn}(Q_2, K_2, V_2) = \mathrm{softmax}\left(\frac{Q_2 K_2^T}{\sqrt{d/2}}\right)V_2$$

$$Y = \mathrm{concat}(h_1,\ h_2)W_O$$

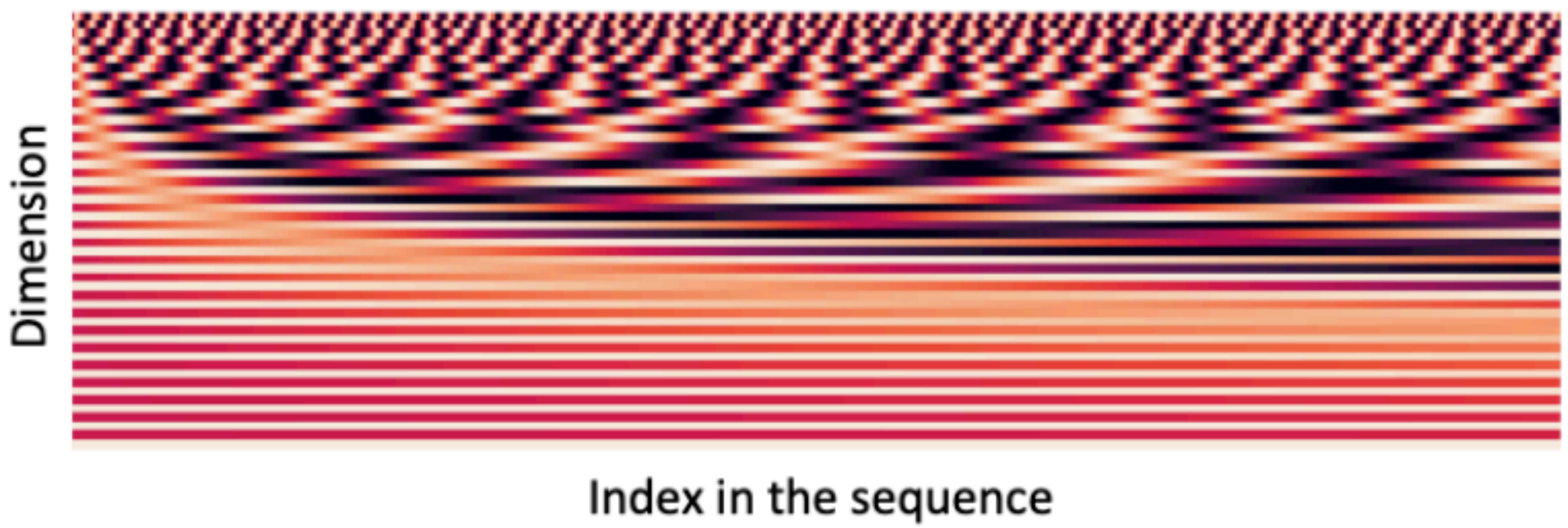
Missing Piece: Positional Information

- **Unlike RNNs, self-attention does not build in order information**
  - Encode the order of the sentence into the input $x_1, \ldots, x_n$
- **Solution: add positional encoding to the input embeddings**

$$x_i \leftarrow x_i + p_i$$

- **Use sine and cosine functions of different frequencies (not learnable)**

$$p_i = \begin{pmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{2*\frac{d}{2}/d}) \\ \cos(i/10000^{2*\frac{d}{2}/d}) \end{pmatrix}$$



Recap: Adding Nonlinearities

- There is no elementwise nonlinearities in self-attention; stacking more self-attention layers just re-averages value vectors

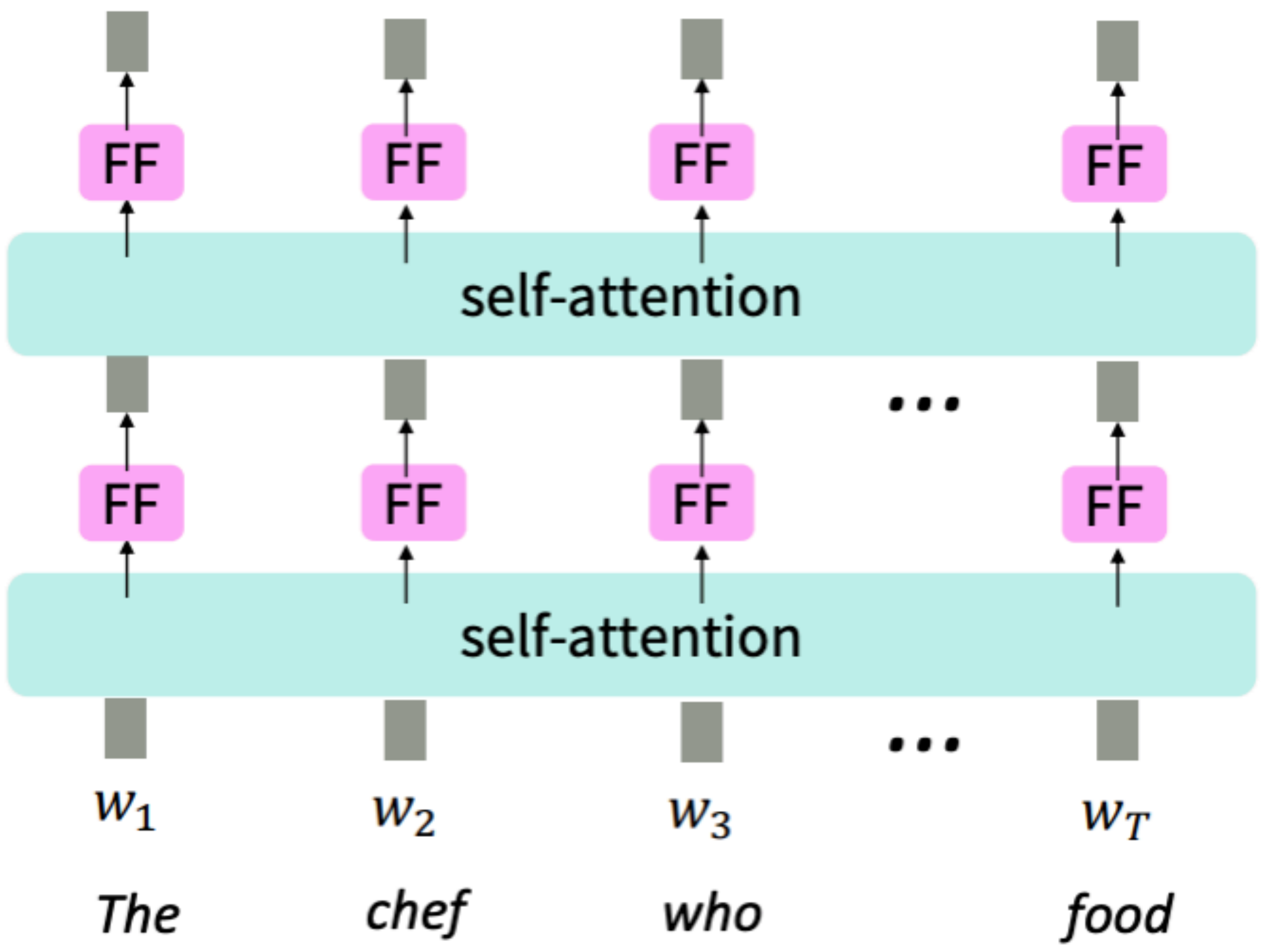
- Simple fix: add a feed-forward network to post-process each output vector

$$\mathrm{FFN}(\mathbf{x}_i) = W_2 \mathrm{ReLU}(W_1 \mathbf{x}_i + \mathbf{b}_1) + \mathbf{b}_2$$

A large number of parameters

$$\boxed{\begin{aligned} & W_1 \in \mathbb{R}^{d_{ff} \times d}, \mathbf{b}_1 \in \mathbb{R}^{d_{ff}} \\ & W_2 \in \mathbb{R}^{d \times d_{ff}}, \mathbf{b}_2 \in \mathbb{R}^{d} \end{aligned}}$$

In practice, they use $d_{ff} = 4d$



Slide credit: Danqi Chen COS 484/584

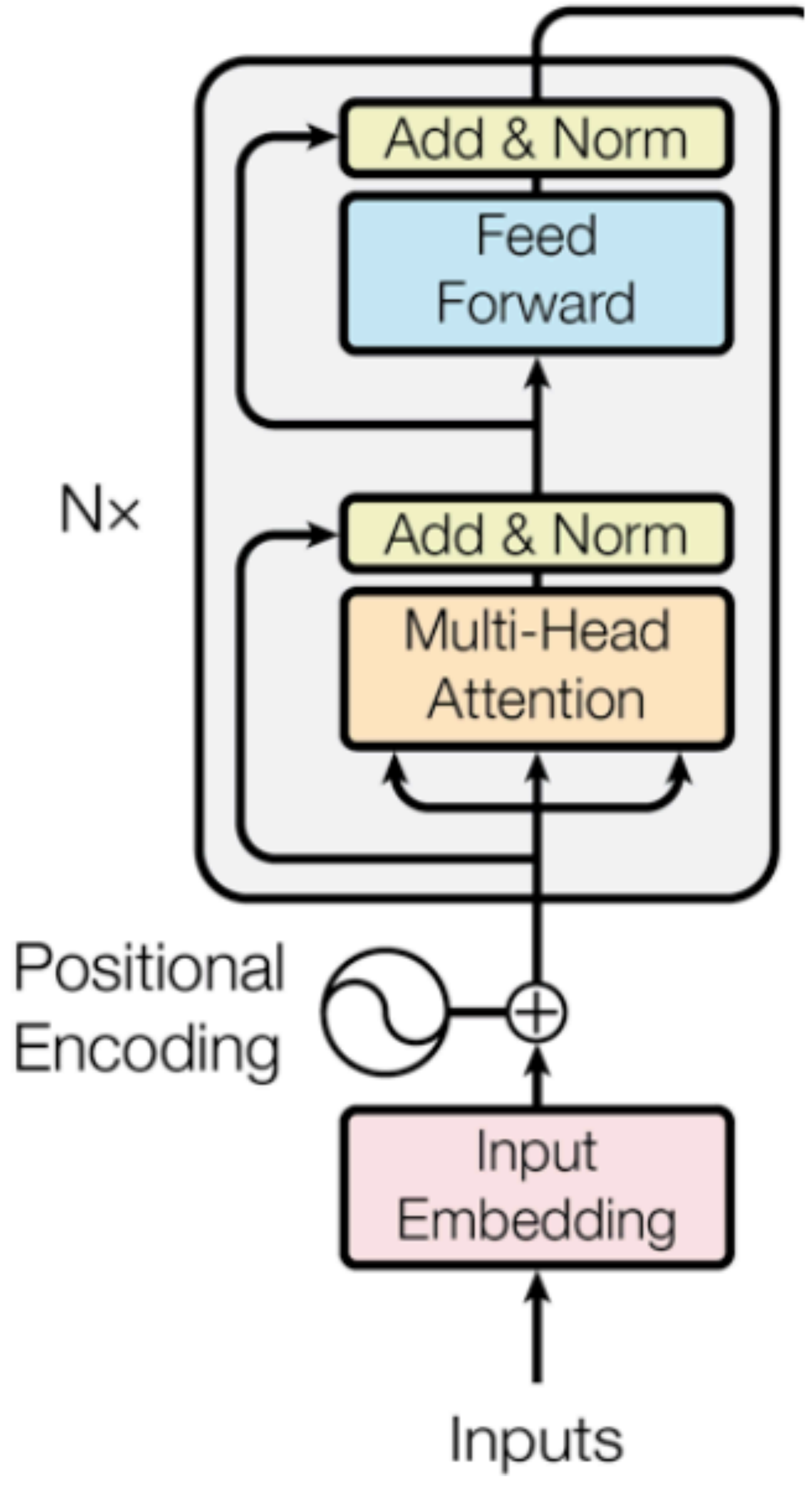
Recap: Transformer Encoder

- **Each encoder layer has two sub-layers:**
  - A multi-head self-attention layer
  - A feedforward layer
- **Add & Norm:**
  - Add: Residual connection (He et al., 2016)

$$Y \leftarrow Y + X$$

  - Norm: Layer normalization (Ba et al., 2016)

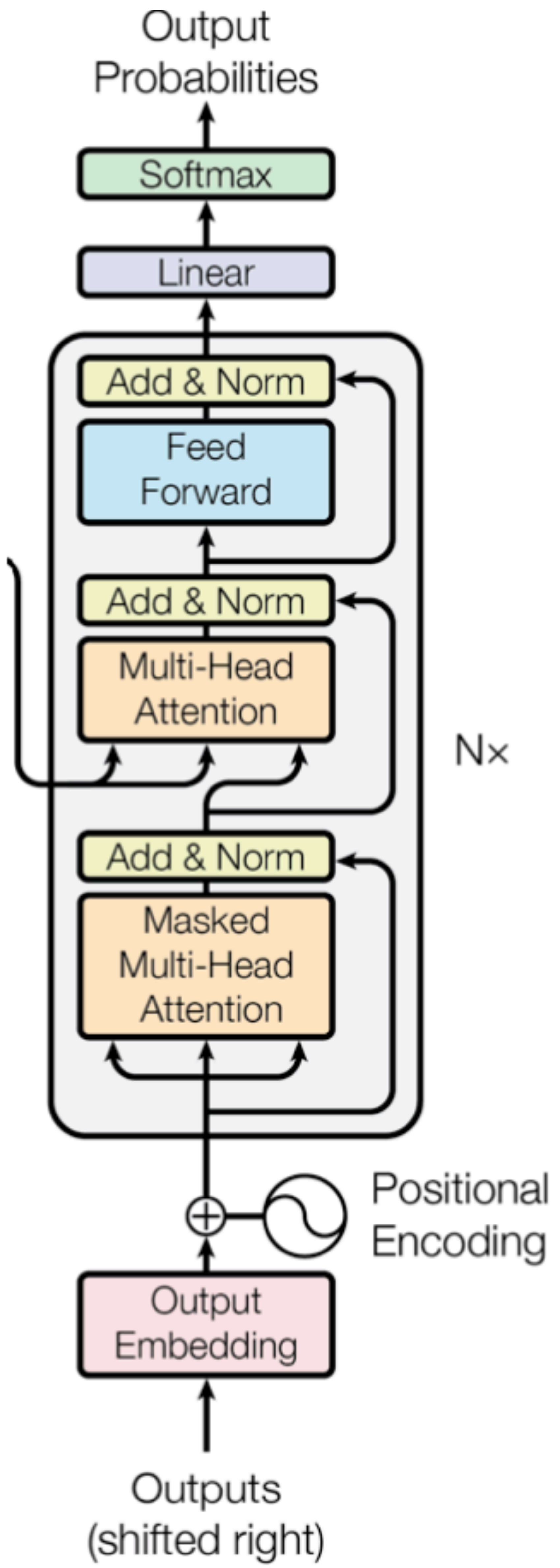
$$Y = \frac{X - \mathrm{E}[X]}{\sqrt{\mathrm{Var}[X] + \epsilon}} * \gamma + \beta$$



In (Vaswani et al., 2017), N = 6

Recap: Transformer Decoder

- **Each decoder layer has three sub-layers:**
  - A masked multi-head attention layer
  - A multi-head cross attention layer
  - A feedforward layer
- **Masked multi-head attention**
  - self-attention on the decoder states
- **Multi-head cross attention**
  - Decoder attends to encoder states
  - Encoder: keys/values
  - Decoder: queries

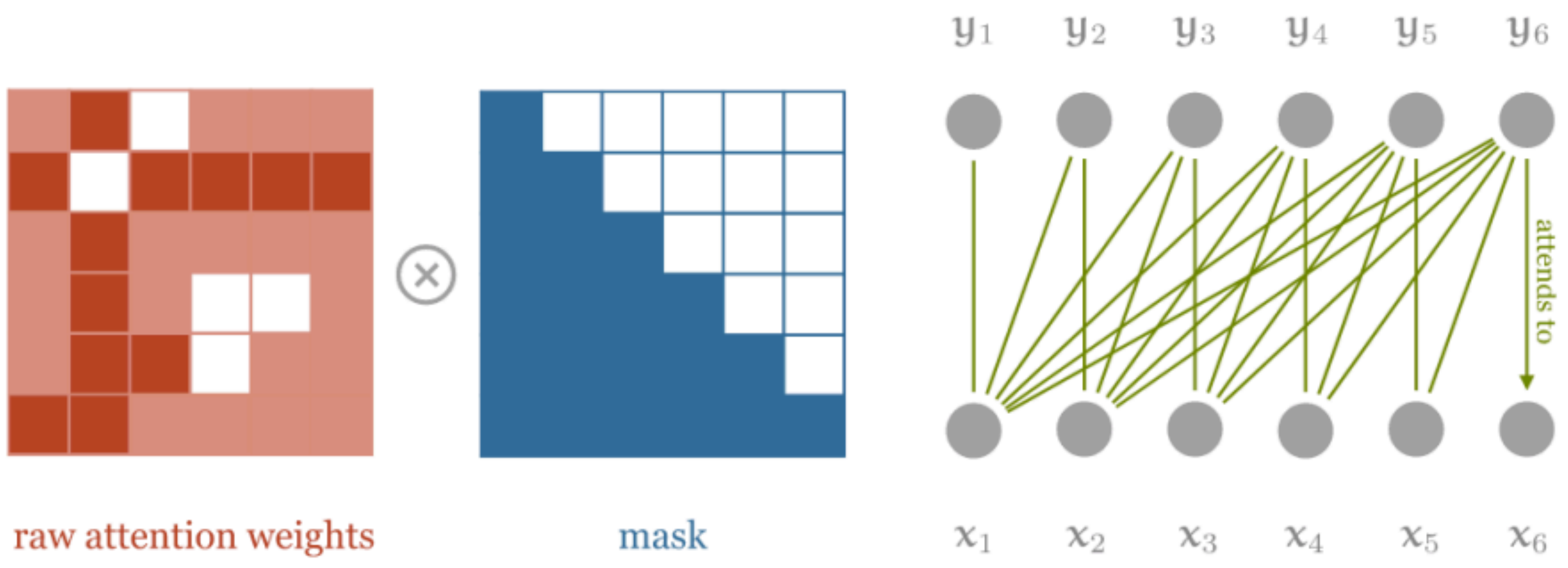


In (Vaswani et al., 2017), N = 6

Recap: Masked Multi-Head Attention

$$q_i = W_Q x_i,\ k_i = W_K x_i,\ v_i = W_V x_i$$

$$a_{i,j} = \mathrm{softmax}\left(\frac{q_i^T k_j}{\sqrt{d}}\right)$$



Efficient implementation: compute attention as we normally do, mask out attention to future words by setting attention scores to $-\infty$

```
dot = torch.bmm(queries, keys.transpose(1, 2))

indices = torch.triu_indices(t, t, offset=1)
dot[:, indices[0], indices[1]] = float('-inf')

dot = F.softmax(dot, dim=2)
```

http://peterbloem.nl/blog/transformers

Slide credit: Danqi Chen COS 484/584

Transformer: Pros and Cons

- **Easier to capture dependencies:** we draw attention between every pair of words!
- **Easier to parallelize:**

$$\boxed{\begin{gathered}\mathrm{MultiHead}(X) = \mathrm{concat}(h_1, \ldots, h_k)W_O \\ h_i = \mathrm{attn}(Q_i, K_i, V_i) \\ Q_i = (XW_Q)^i, K_i = (XW_K)^i, V_i = (XW_V)^i\end{gathered}}$$

$$\boxed{\mathrm{attn}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d}})V}$$

- **Quadratic computation in self-attention:**
  - Can be very expensive when the sequence is very long
- **Harder to model positional information**

Advances In NMT

- Semi-Supervised NMT
- Multilingual NMT
- Evaluation beyond BLEU

Semi-Supervised NMT

Semi-Supervised NMT: Motivation

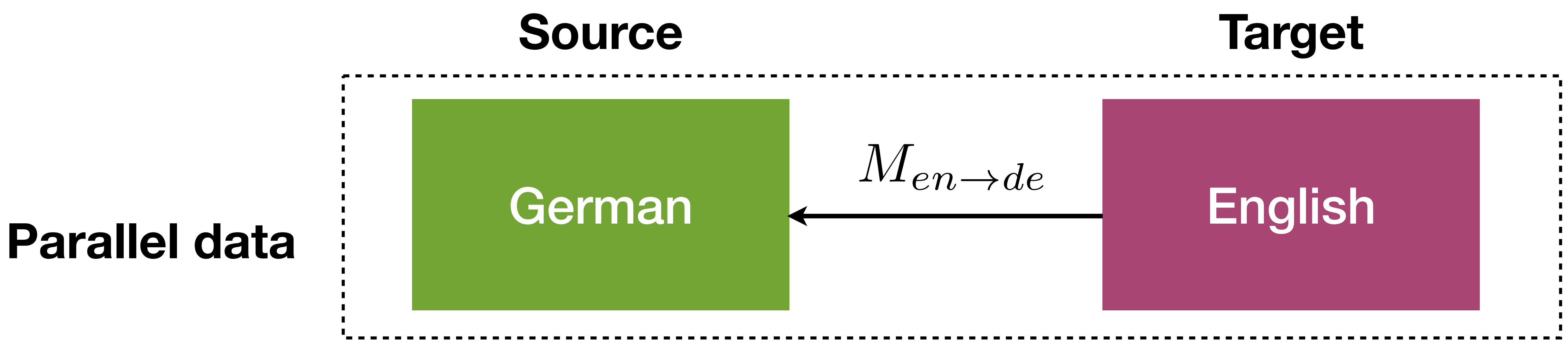
- First train a translation model with limited amount of parallel sentences
- Use this model to generate more synthetic sentence pairs with monolingual corpus

Semi-Supervised NMT: Scenarios

- **Monolingual data from target side**
  - Back-translation
- **Monolingual data from source side**
  - Self-learning

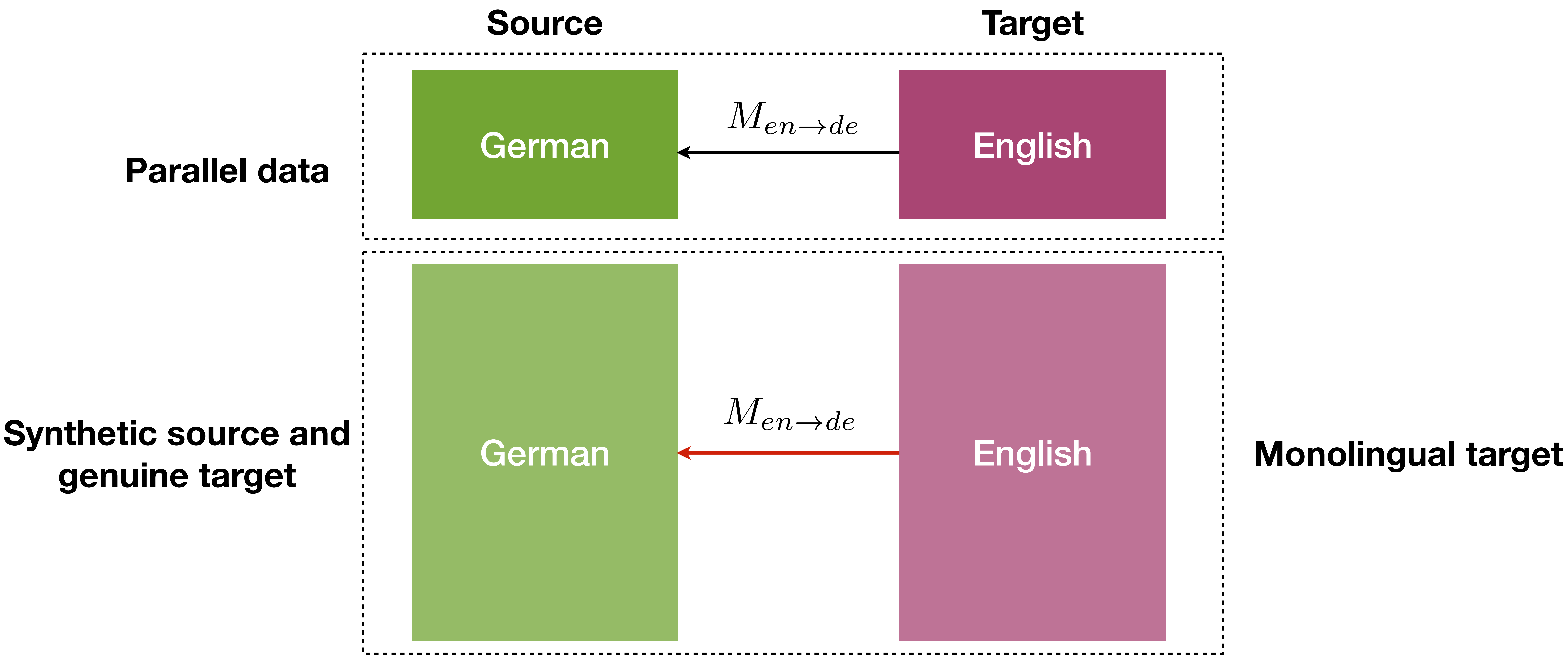
Back Translation

- **<u>Back-translation:</u> using monolingual target side data (Sennrich et al., 2016, Edunov et al., 2018)**



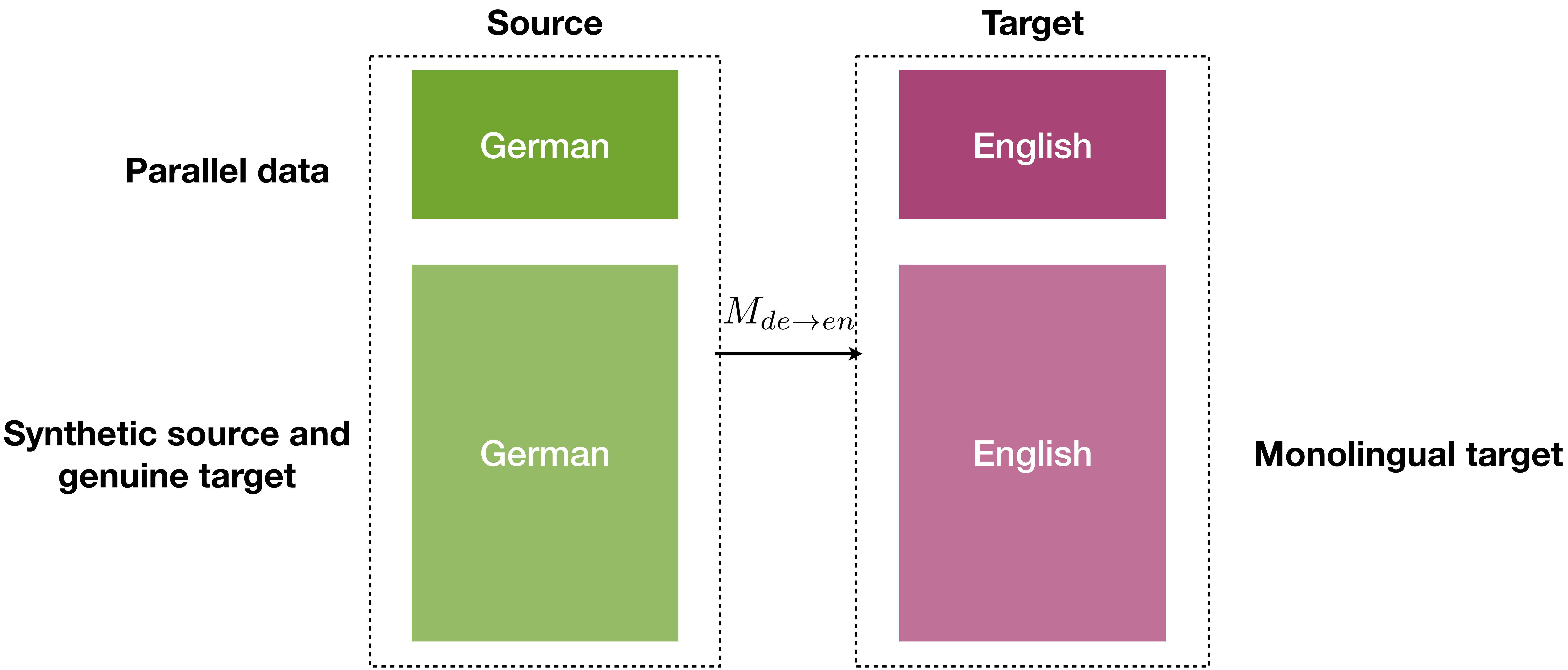
Back Translation

- <u>Back-translation:</u> using monolingual target side data (Sennrich et al., 2016, Edunov et al., 2018)



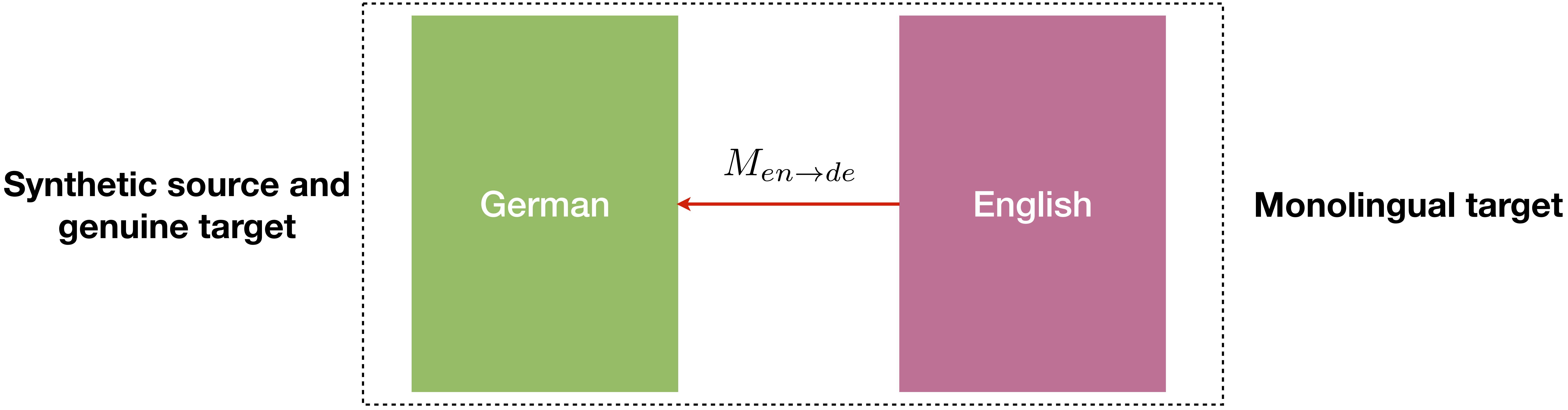
Back Translation

- <u>Back-translation:</u> using monolingual target side data (Sennrich et al., 2016, Edunov et al., 2018)



Back Translation

- **How to generation the syntactic data?**
  - Beam search
  - Greedy search
  - Top k
  - Sampling from $p_\theta(Y|X)$
  - +noise



Back Translation: Results

- **English-German**
- **Data**
  - Parallel: WMT-18 (5.2M sentence pairs)
  - Monolingual: 24M German sentences

	news2013	news2014	news2015	news2016	news2017	Average
bitext	27.84	30.88	31.82	34.98	29.46	31.00
+ beam	27.82	32.33	32.20	35.43	31.11	31.78
+ greedy	27.67	32.55	32.57	35.74	31.25	31.96
+ top10	28.25	33.94	34.00	36.45	32.08	32.94
+ sampling	28.81	34.46	34.87	37.08	32.35	33.51
+ beam+noise	29.28	33.53	33.79	37.89	32.66	33.43

Back Translation: Explanations

- **More sentences in target language improves decoder**
  - Better language model in target language
- **Synthetic sentences (with noise) improves encoder**
  - More robust against imperfect source sentences

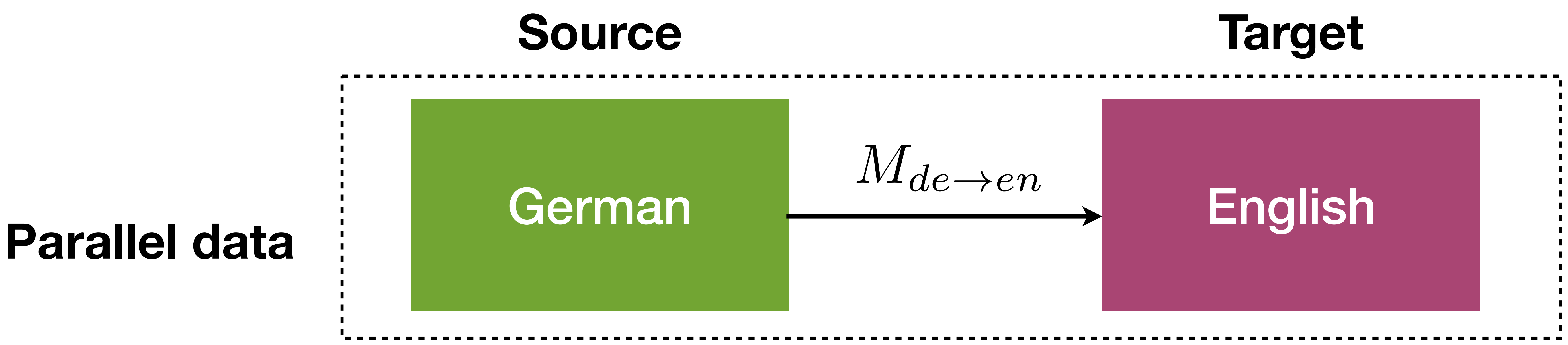
	news2013	news2014	news2015	news2016	news2017	Average
bitext	27.84	30.88	31.82	34.98	29.46	31.00
+ beam	27.82	32.33	32.20	35.43	31.11	31.78
+ greedy	27.67	32.55	32.57	35.74	31.25	31.96
+ top10	28.25	33.94	34.00	36.45	32.08	32.94
+ sampling	28.81	34.46	34.87	37.08	32.35	33.51
+ beam+noise	29.28	33.53	33.79	37.89	32.66	33.43

Semi-Supervised NMT: Scenarios

- **Monolingual data from target side**
  - Back-translation
- **Monolingual data from source side**
  - Self-learning

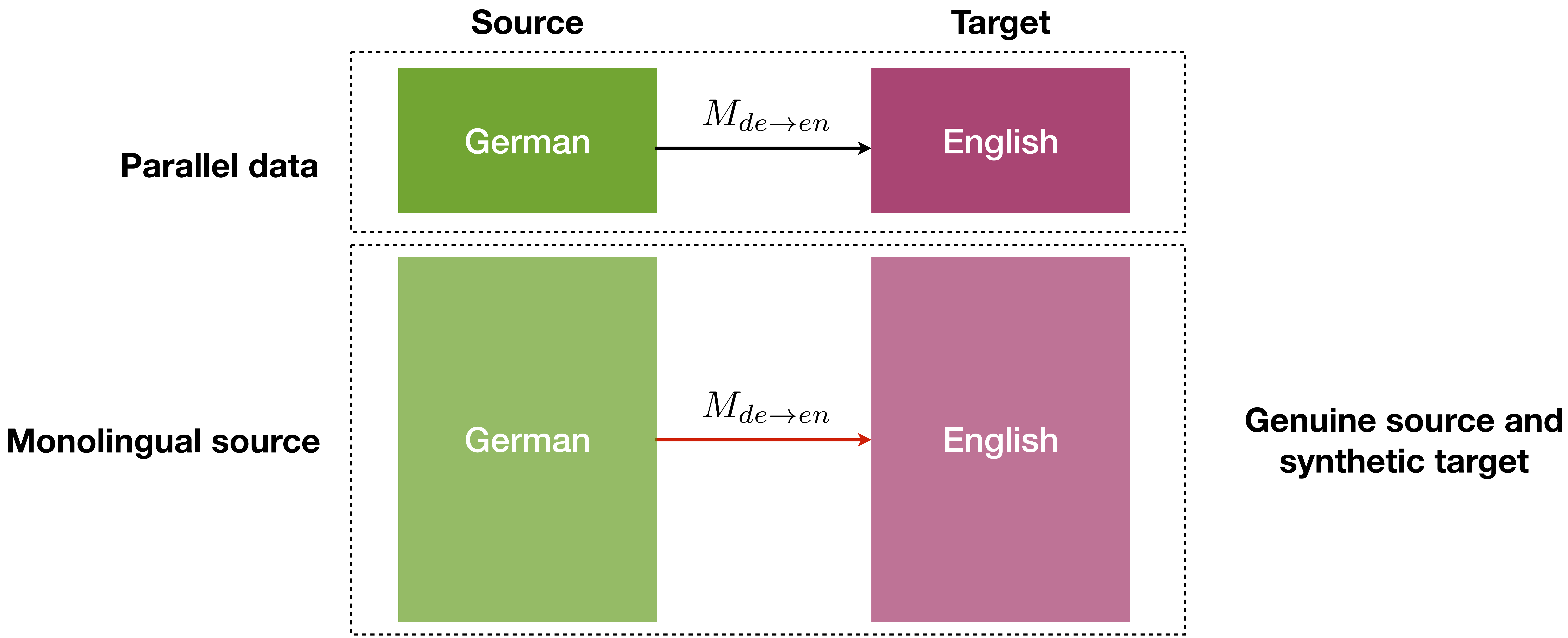
Self-Learning

- **<u>Self-training:</u> using monolingual source side data (Scudder 1965, He et al., 2020)**



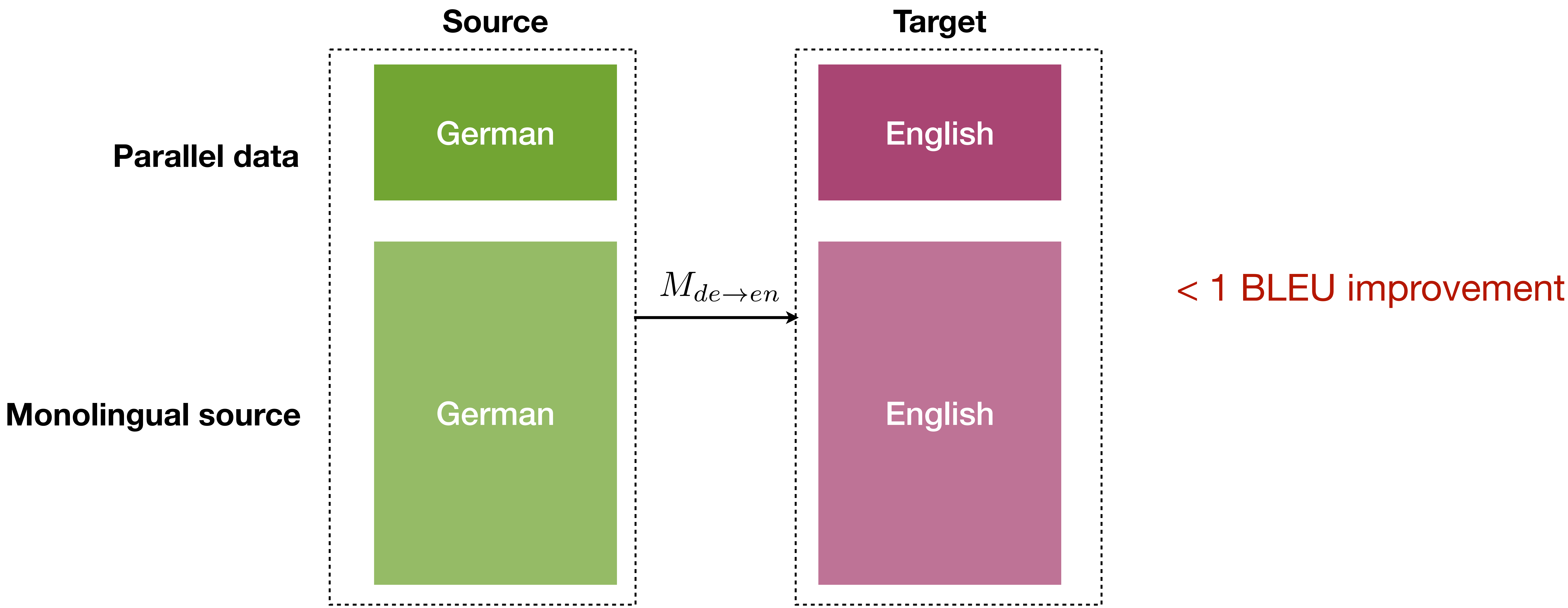
Self-Learning

- **Self-training:** using monolingual source side data (Scudder 1965, He et al., 2020)



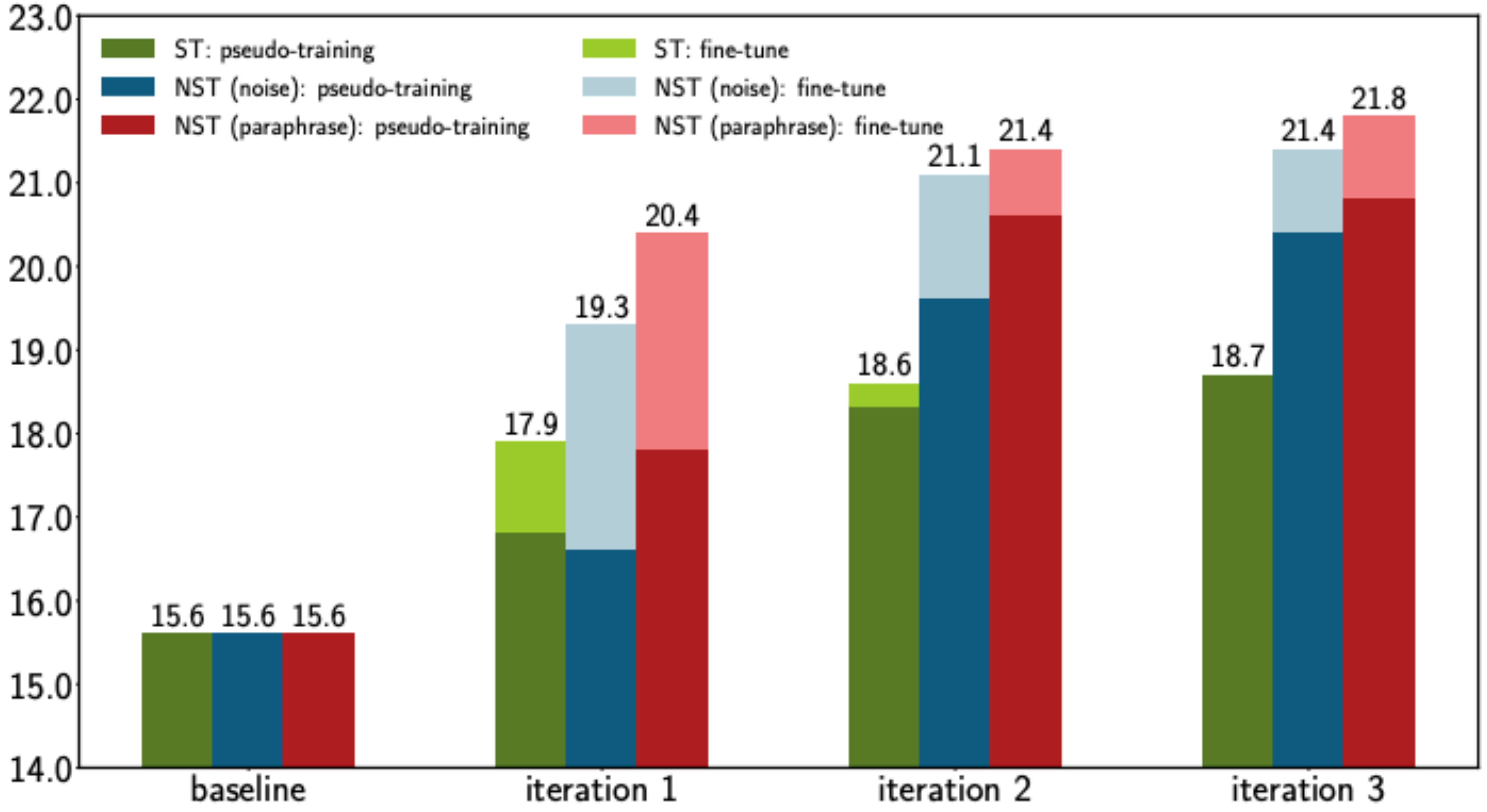
Self-Learning

- **Self-training:** using monolingual source side data (Scudder 1965, He et al., 2020)



Noisy Self-Learning

- **We should add noise to encoder, but genuine sentences to decoder!**
- **Adding noises to source inputs** (He et al., 2020)
  - Word dropout (masks)
  - Permutations
  - Paraphrasing



Semi-Supervised NMT: Summary

- **Motivation**
  - Leveraging large-scale monolingual data to improve MT models
  - Monolingual target sentences: back-translation
  - Monolingual source sentences: self-learning
- **Empirical Evidences**
  - Genuine sentences helps decoder: better language model
  - Noisy sentences helps encoder: robust against noise

Multilingual NMT

Multilingual NMT

- **Many languages are left behind**
  - There are not enough monolingual data for many languages
  - Even less annotated data for NMT

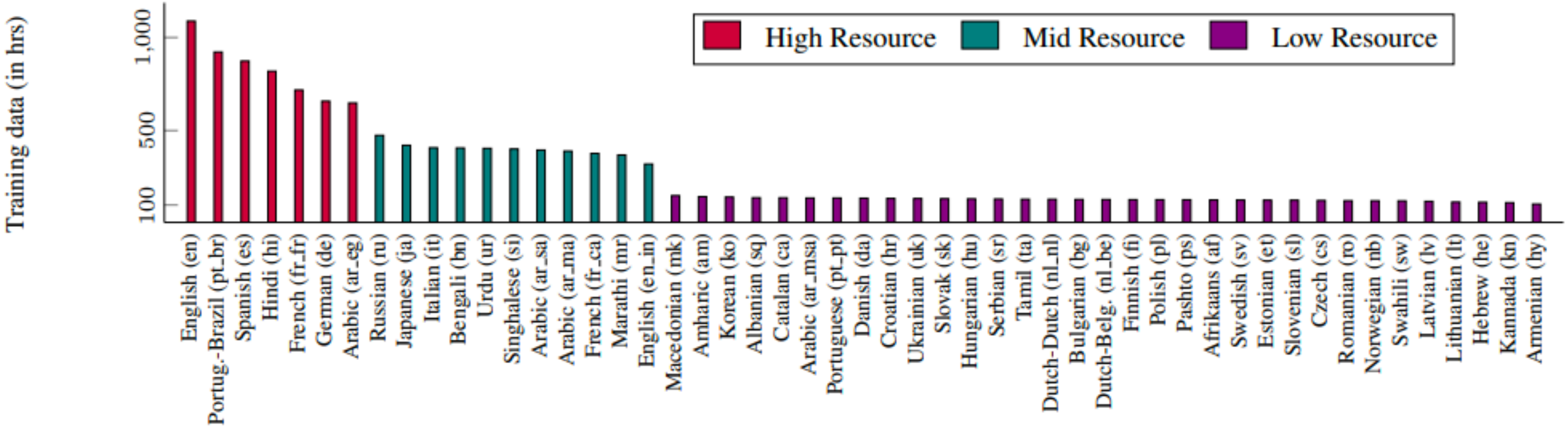
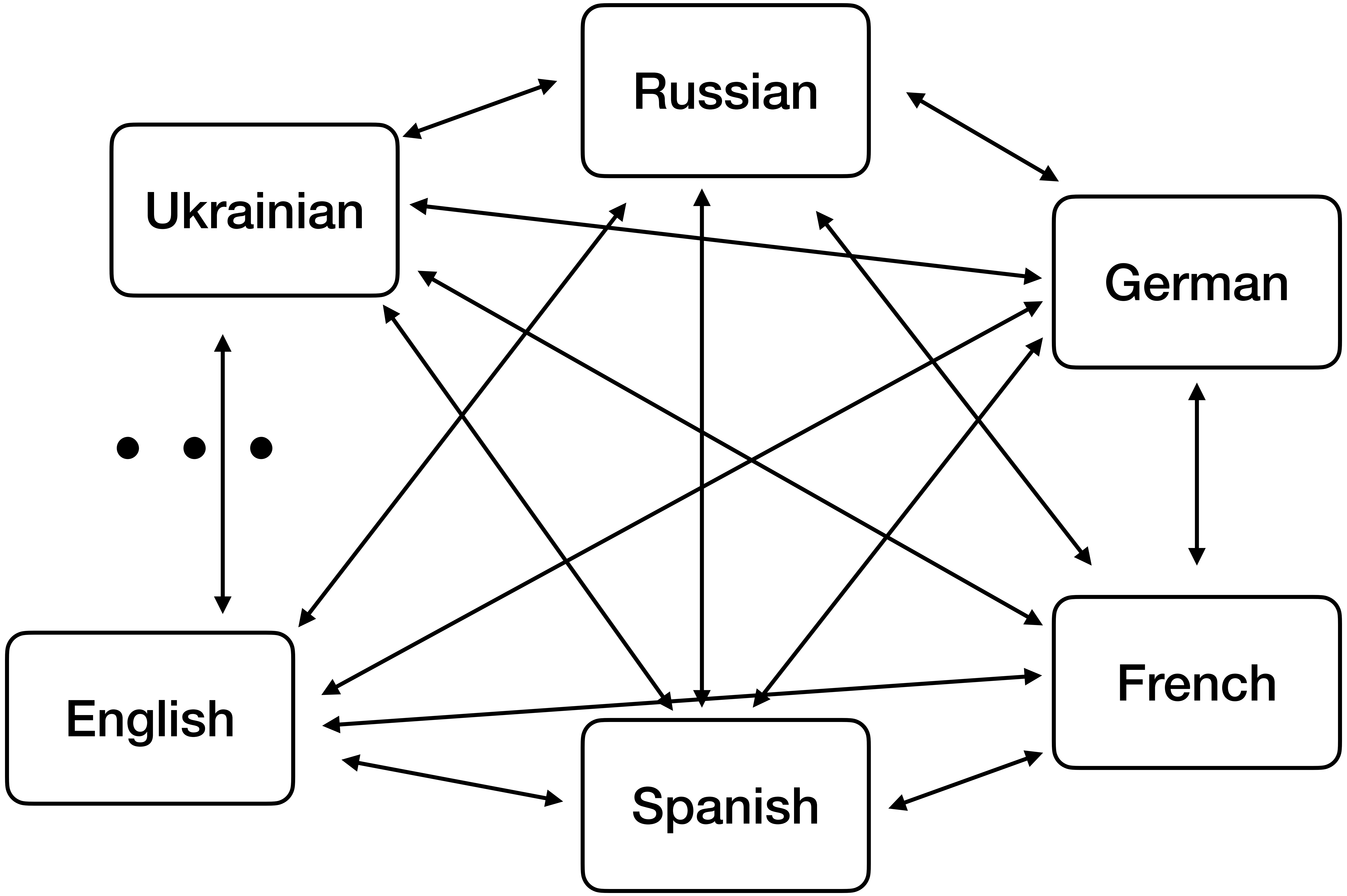


Figure : *Training data distribution across different languages*

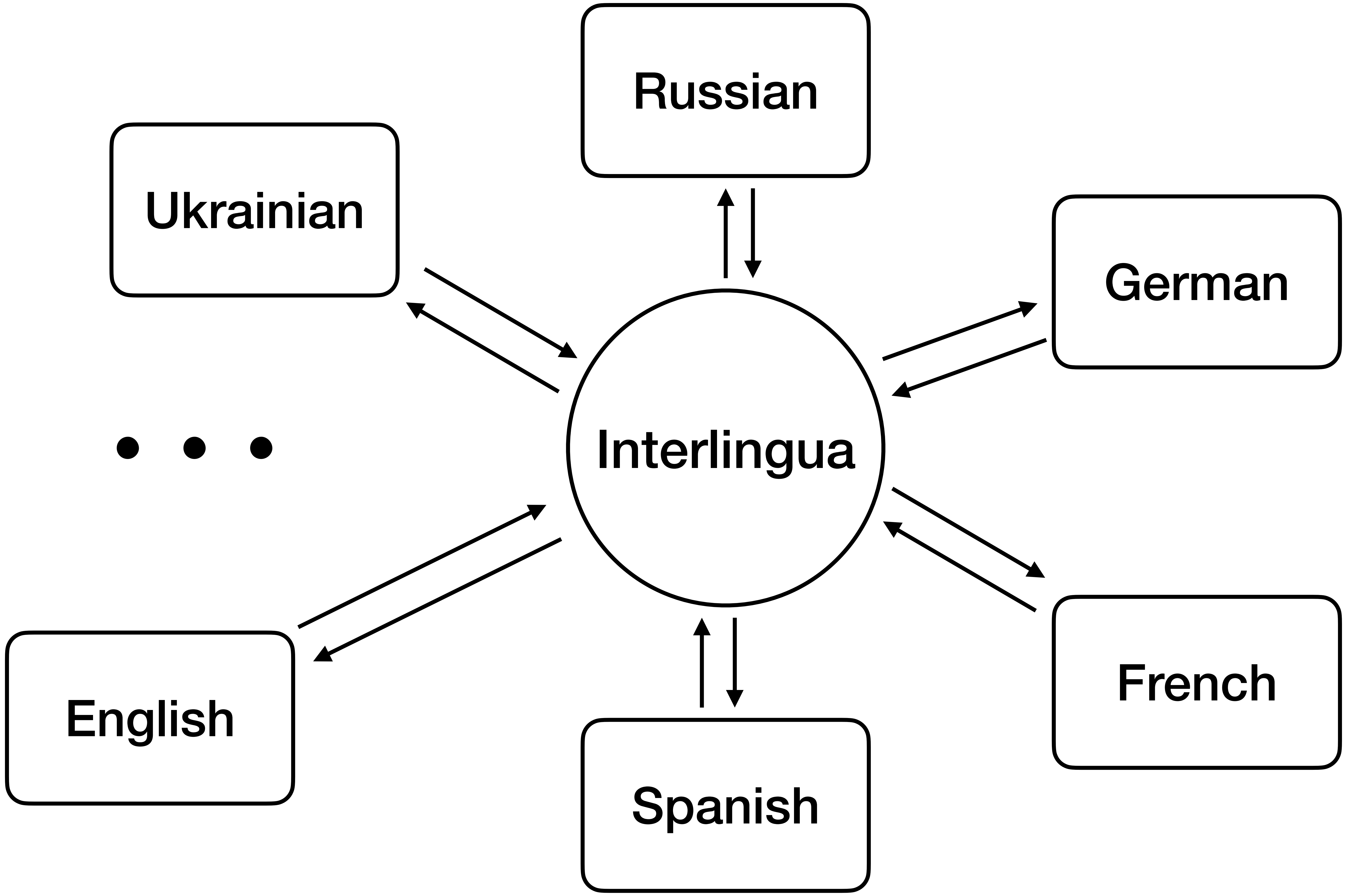
Multilingual NMT

- **Supporting multiple languages could be tedious**
  - Supporting translating from n languages requires $n \times (n-1)$ NMT models



Multilingual NMT

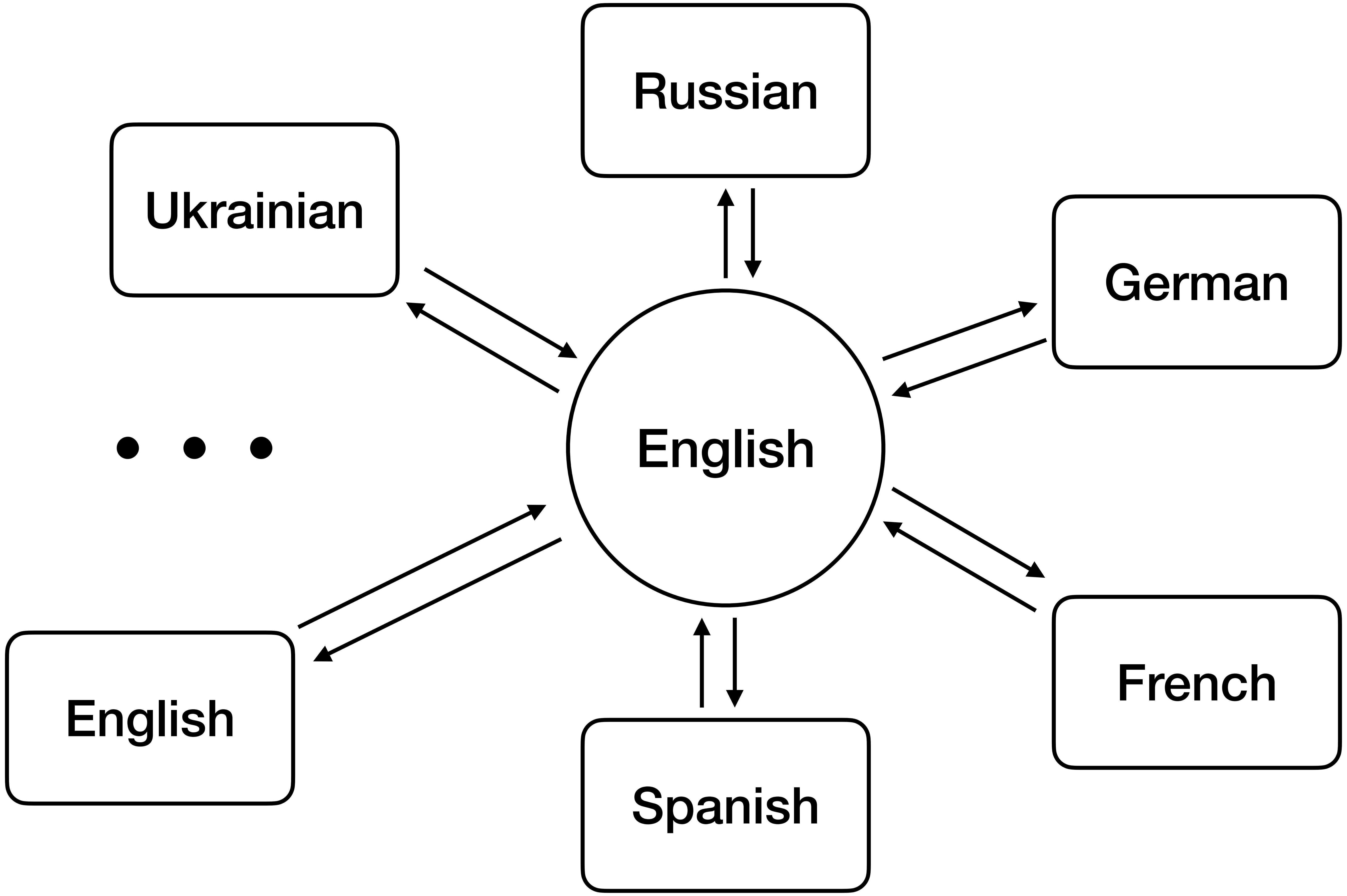
- **Interlingual representation for NMT**



Small languages benefit from big ones that are in the same language family

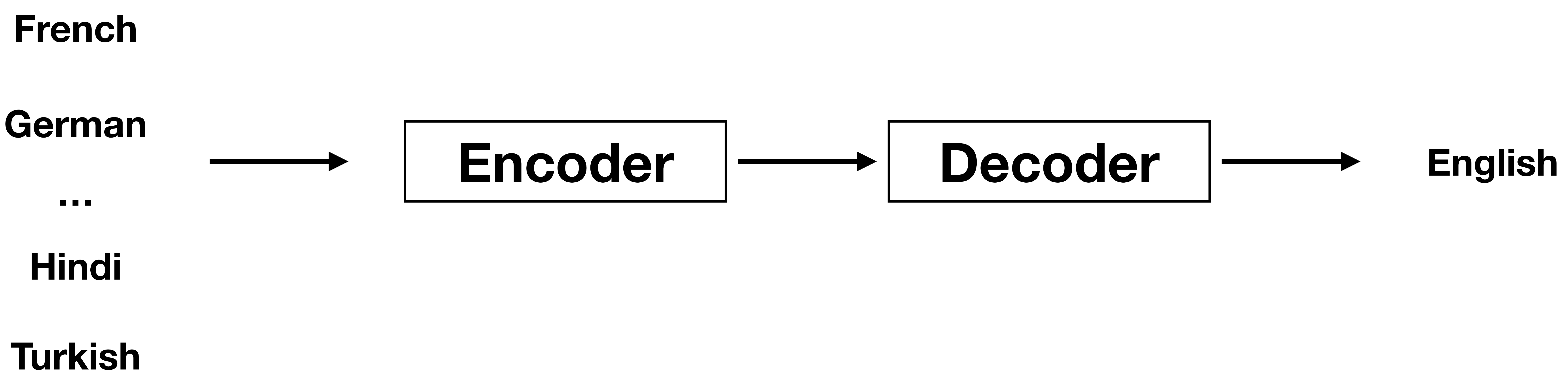
Many-to-One NMT

- **English as a pivot language**



Many-to-One NMT

- Training a single Encoder-Decoder model from multiple languages to English



We need a shared vocabulary across multiple languages!

Vocabulary across Multiple Languages

- **Lexical Divergences**
  - Wall in English corresponds to two words in German, *Wand* (walls inside a building) and *Mauer* (walls outside a building)
- **Morphological Divergences**
  - Number of morphemes per word
    - Isolating languages: Chinese and Vietnamese
    - Polysynthetic languages: Eskimo
  - Morphological boundary
    - Agglutinative languages: relatively clean boundaries
      - Turkish
    - Fusion languages: no clean boundaries
      - Russian: *stolom* (table-SG-INSTR-DECL1)
      - -om: singular (SG), instrumental (INSTR) and first declension (DECL1)

Vocabulary across Multiple Languages

- **Combination of individual vocabularies**
  - Too many different words
  - No shared information
- **Character- or Byte- level vocabulary**
  - Too long sentences
  - Too difficult contextual information

Trade-off between these two ideas?

Byte Pair Encoding

- **First split each word into characters (bytes)**
- **Count the frequency of each consecutive byte pair, find out the most frequent one and merge the two byte pair tokens to one item**

	V={all chars/bytes}	V=V + {es}	V=V + {est}
low: 5	l o w </w>: 5	l o w </w>: 5	l o w </w>: 5
lower: 2	l o w e r </w>: 2	l o w e r </w>: 2	l o w e r </w>: 2
newest: 6	n e w e s t </w>: 6	n e w es t </w>: 6	n e w est </w>: 6
widest: 3	w l d e s t </w>: 3	w l d es t </w>: 3	w l d est </w>: 3

......

Byte Pair Encoding

- First split each word into characters (bytes)
- Count the frequency of each consecutive byte pair, find out the most frequent one and merge the two byte pair tokens to one item
- Iterate from the longest token from learned vocabulary to the shortest one, trying to replace the substring in each of the word to tokens.

V={$\boxed{\text{est</w>}}$, gh, *chars/bytes*}

highest ⟶ h i g h $\boxed{\text{e s t </w>}}$ ⟶ h i g h est</w>

Byte Pair Encoding

- **First split each word into characters (bytes)**
- **Count the frequency of each consecutive byte pair, find out the most frequent one and merge the two byte pair tokens to one item**
- **Iterate from the longest token from learned vocabulary to the shortest one, trying to replace the substring in each of the word to tokens.**

V={est</w>, $\boxed{\text{gh}}$, *chars/bytes*}

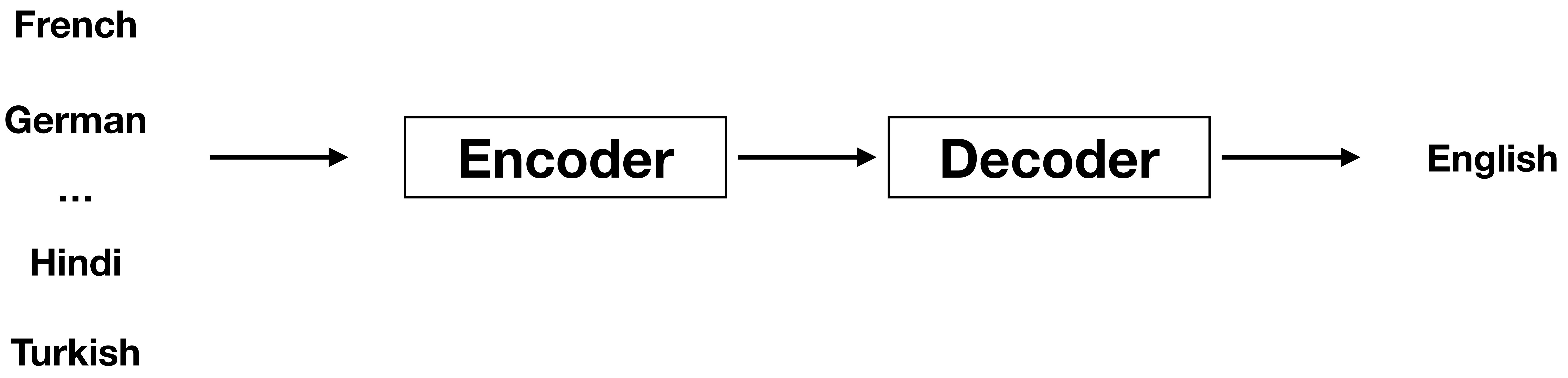
highest → h i g h e s t </w> → h i $\boxed{\text{g h}}$ est</w> → h i gh est</w>

Byte Pair Encoding: Pros and Cons

- **Pros**
  - Trade-off between char/byte -level tokens and original words
  - Capturing shared morphemes/sub-words across similar languages
  - Usually no *unknown* words, unless meeting special/uncommon characters
  - Not only for multilingual tasks, but also for monolingual ones
- **Cons**
  - Shallow similarity, working well only on similar languages
  - Over-segment low-resource or morphologically rich languages

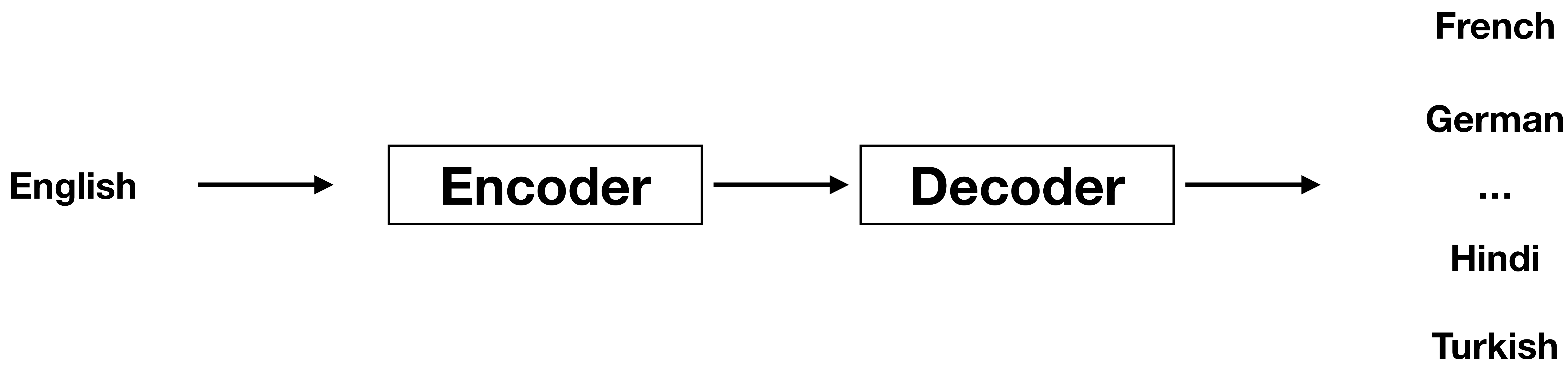
Many-to-One NMT

- Training a single Encoder-Decoder model from multiple languages to English



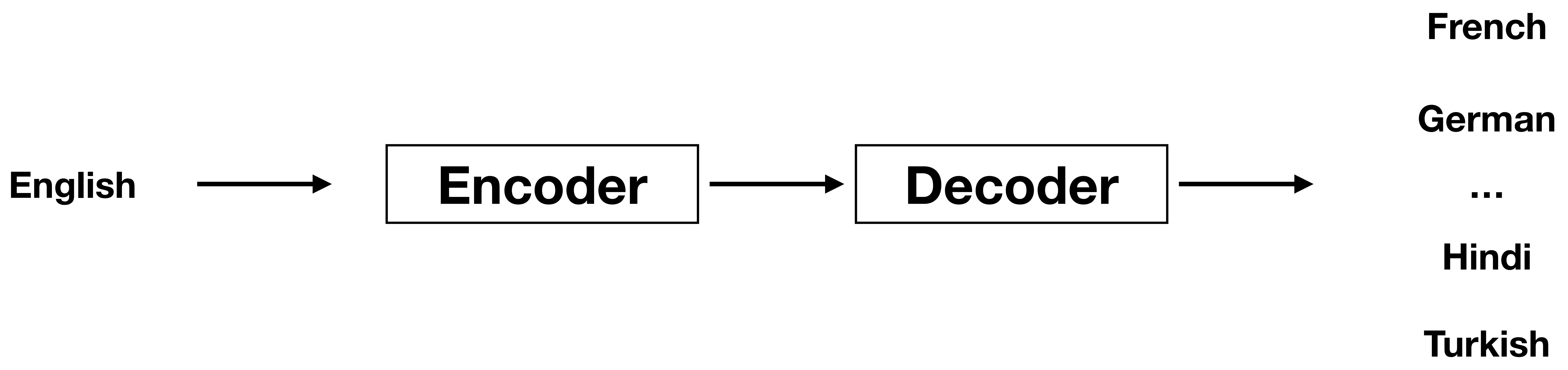
One-to-many?

One-to-Many NMT



Given an English sentence, how could we know which language we want to translate to?

One-to-Many NMT



	French	Comment ça va?
How are you?	**Spanish**	cómo estás?
	Turkish	nasılsın?

One-to-Many NMT

English → Encoder → Decoder →

French

German

...

Hindi

Turkish

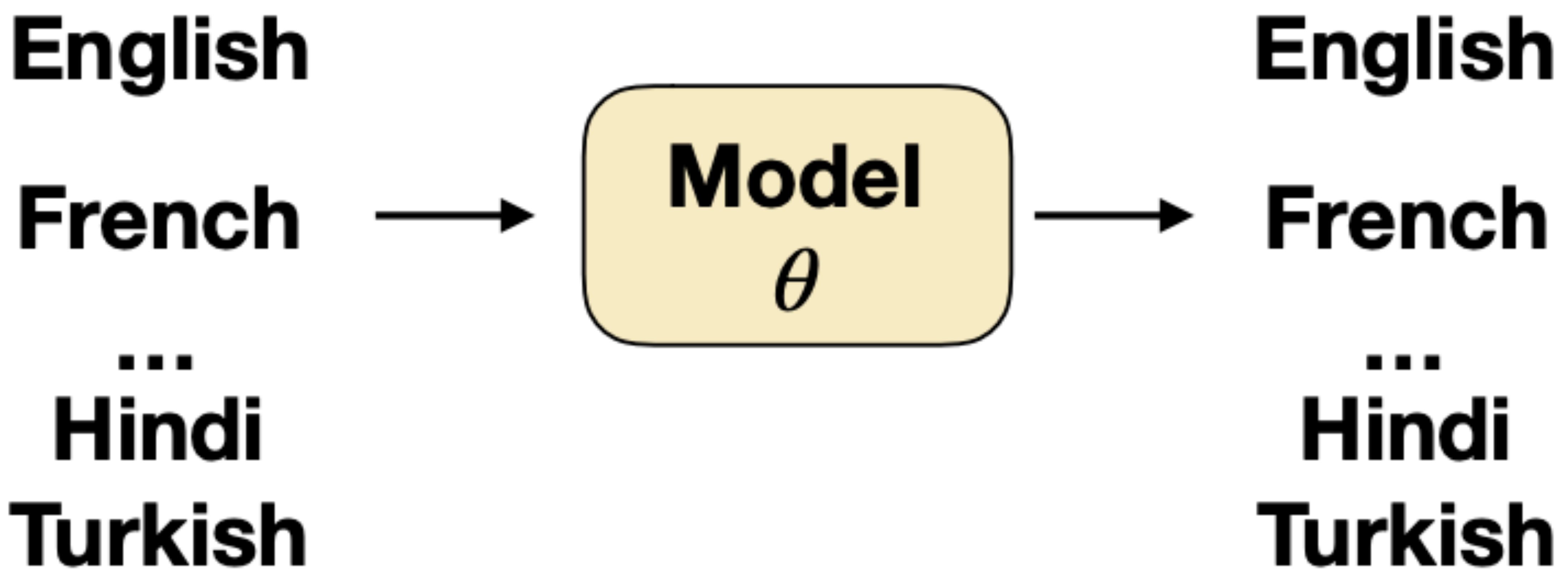
<2fr> How are you? — French: Comment ça va?

<2es> How are you? — Spanish: cómo estás?

<2tr> How are you? — Turkish: nasılsın?

Many-to-Many NMT

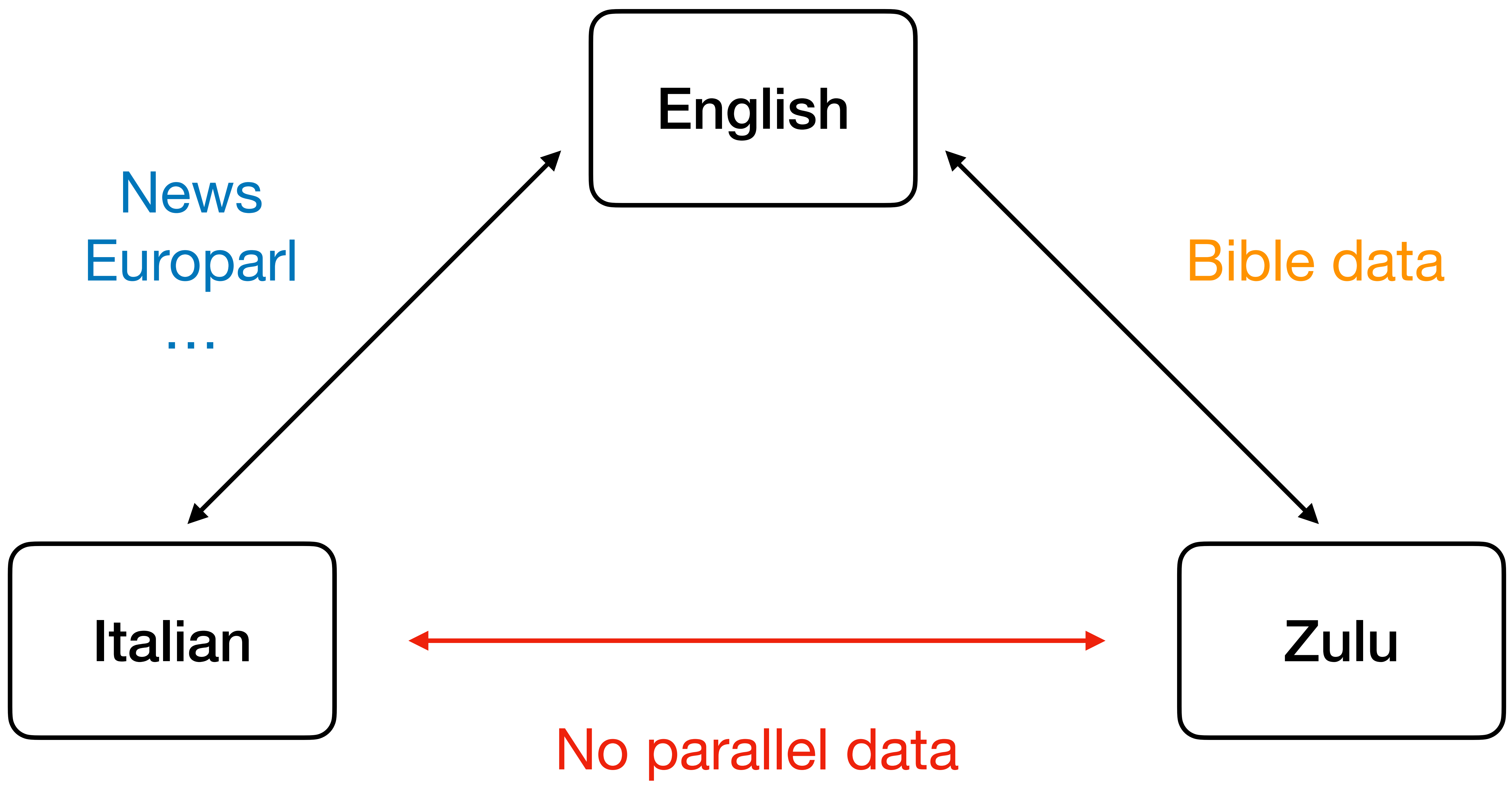
- **Is English always a good pivot language?**
  - Chinese-Japanese
  - Spanish-Portuguese
- **Can we do many-to-many translation?**
  - Training a single model on a mixed dataset from multiple language pairs



Google's multilingual neural machine translation system. (Johnson et al., 2016)

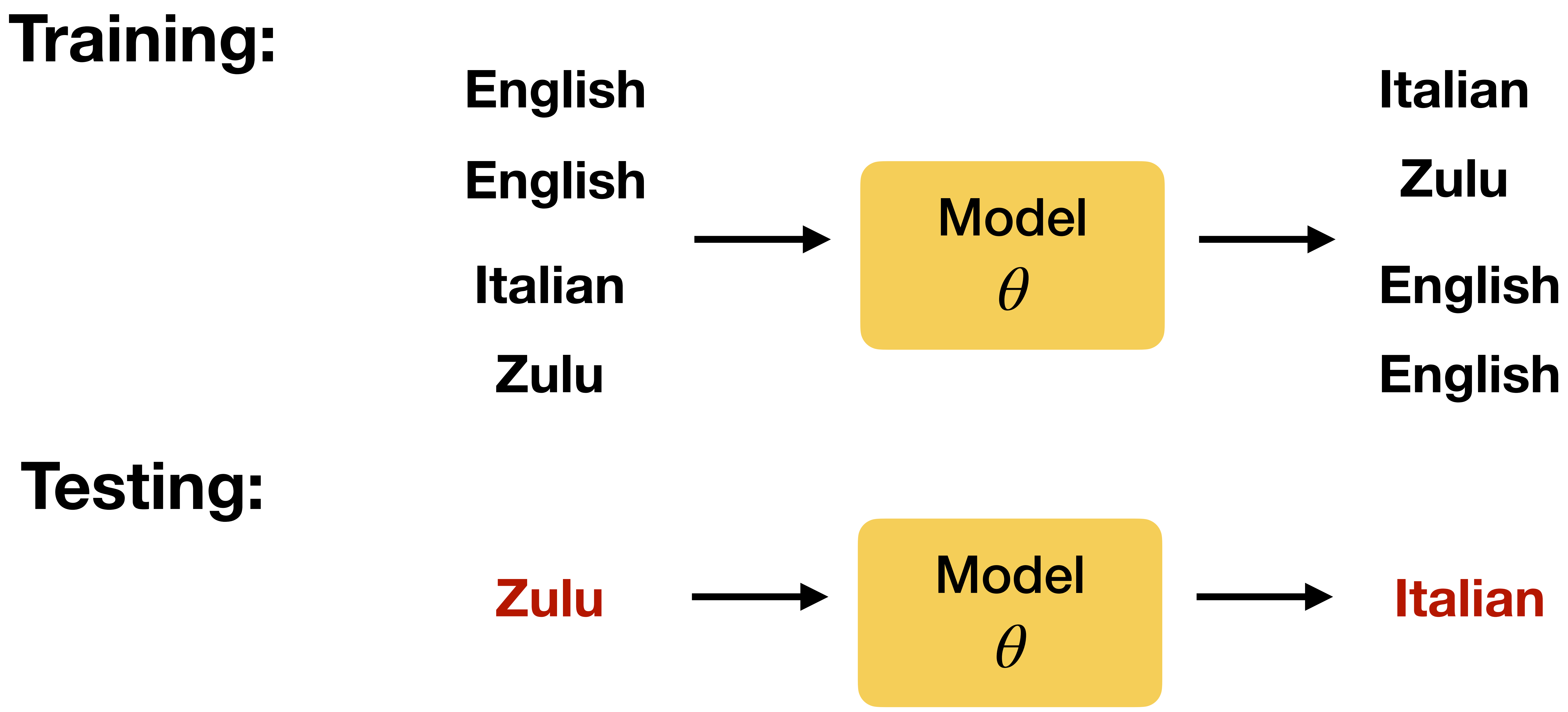
Many-to-Many NMT: Zero-shot Transfer

- Not all language pairs have parallel data



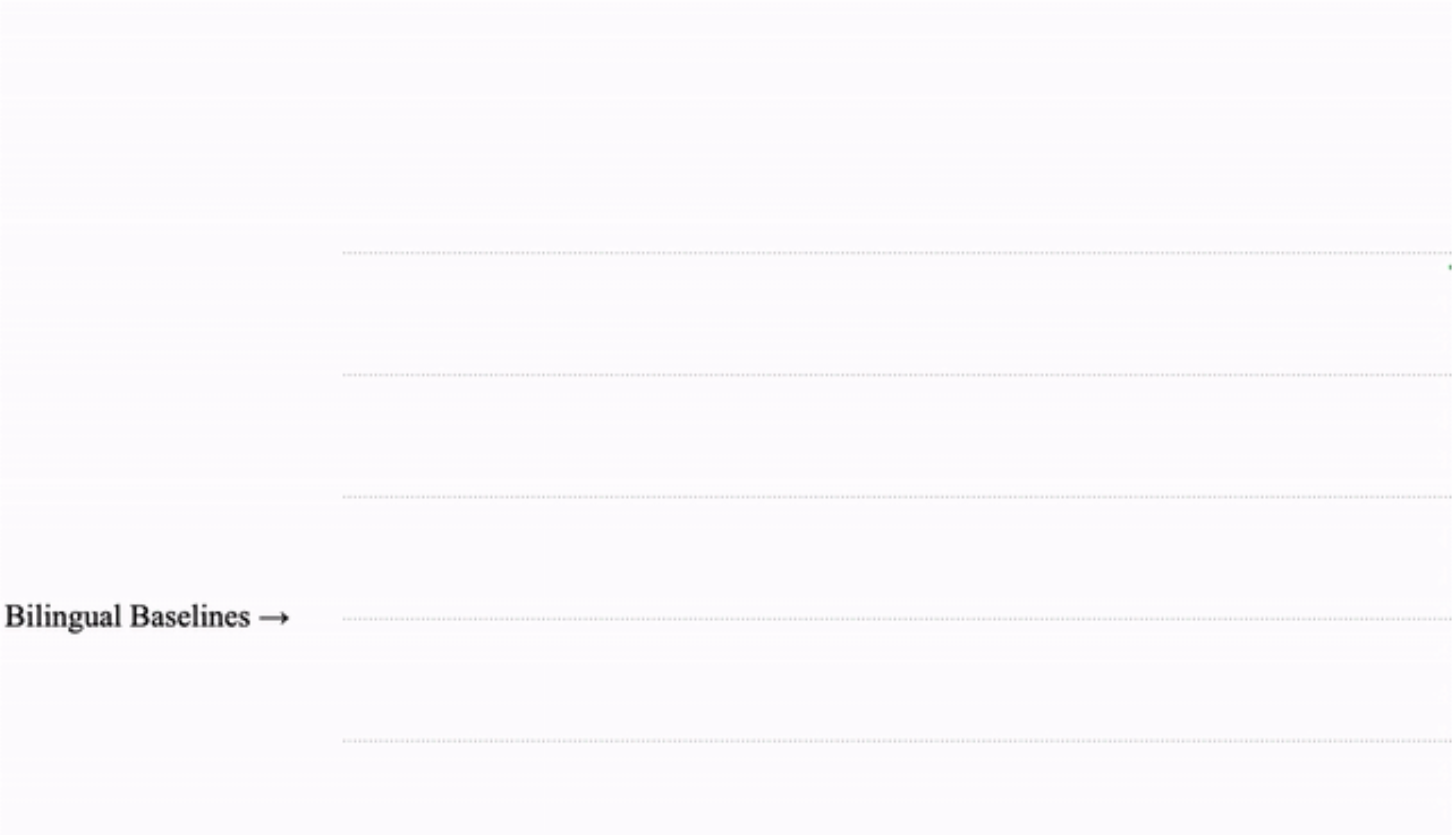
Many-to-Many NMT: Zero-shot Transfer

- Training on {English-Zulu, Zulu-English, English-Italian, Italian-English}
- Zero-shot transfer: the model can translate directly between Zulu and Italian



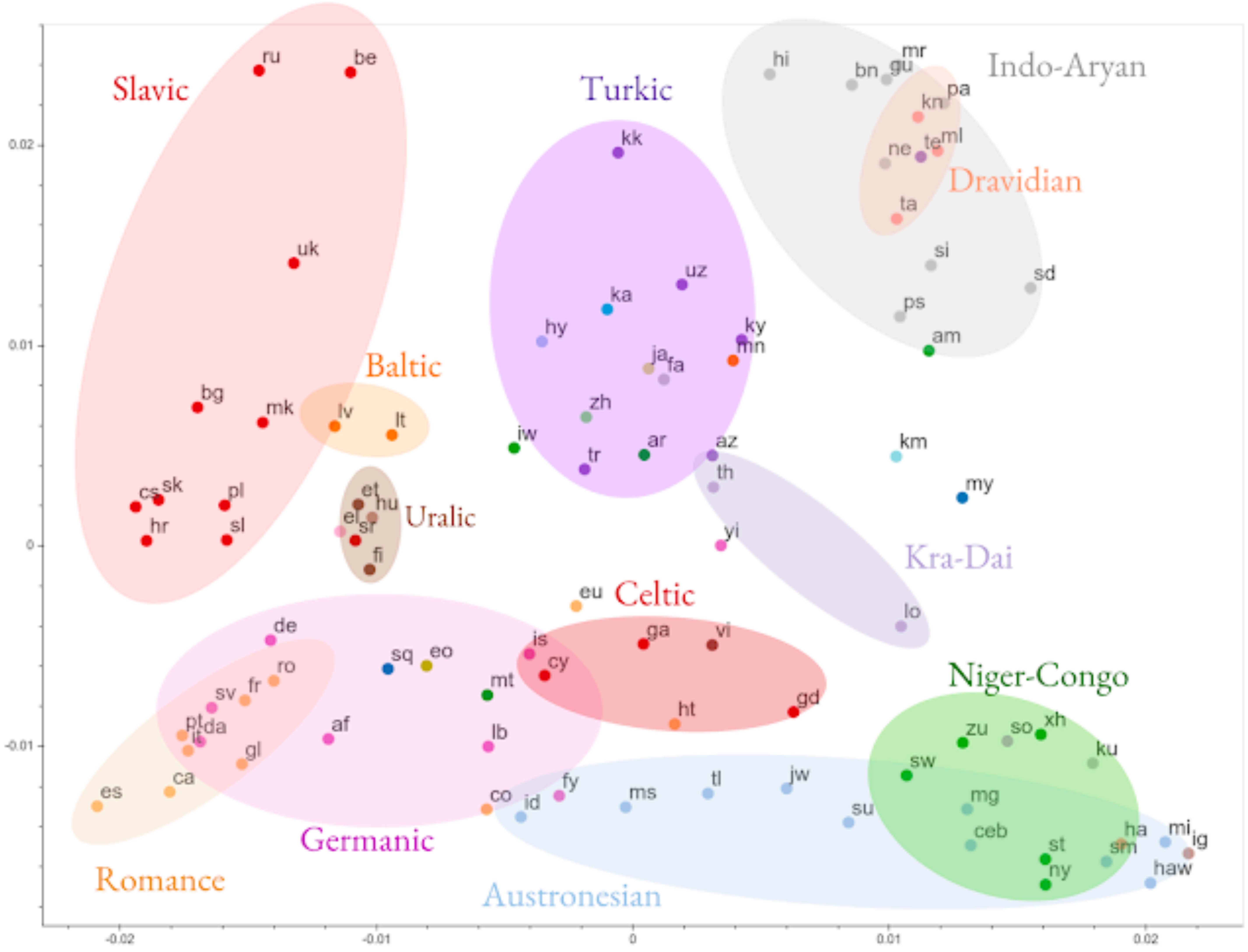
Google's multilingual neural machine translation system. (Johnson et al., 2016)

Many-to-Many NMT



Google's multilingual neural machine translation system. (Arivazhagan et al., 2019)

Many-to-Many NMT



Google’s multilingual neural machine translation system. (Arivazhagan et al., 2019)

Open Problems for Multilingual NMT

- **Imbalanced training data**
  - Important to upsample low-resource data

- **Underperforming bilingual models**
  - Degrading high-resource languages (Arivazhagan et al., 2019)
- **Vocabulary shared across many languages**
  - Upsampling low-resource languages and run joint BPE on all languages
  - Over-segment low-resource or morphologically rich languages
- **Multilingual Evaluation**
  - Average BLEU over all languages or BLUE for the worst case?
  - Are BLUE scores between two languages comparable?

Evaluation beyond BLEU

MT Evaluation

- **Criterion:**
  - Adequacy: measure of correctness
  - Fluency: measure of naturalness
  - Other aspects: hallucination? Coverage?

MT Evaluation

- **Criterion:**
  - Adequacy: measure of correctness
  - Fluency: measure of naturalness
  - Other aspects: hallucination? Coverage?
- **Automatic Evaluation**
  - BLEU score (Papineni et al., 2002): n-gram based metric
    - N-gram precision
    - Brevity penalty
  - Other metrics:
    - METEOR (Denkowski et al., 2014)
    - Translation Edit Rate (TER) (Snover et al., 2006)

MT Evaluation

- **Drawbacks of n-gram based metrics (Zhang et al., 2020):**
  - Penalize semantically-correct paraphases due to string matching
    - e.g. “No worries!” and “Don’t worry!”
  - Fail to capture distant dependencies and penalize semantically-critical ordering changes or word drops(Isozaki et al., 2010)
    - e.g. “A because B” and “B because A”
    - e.g. “A do not like B” and “A likes B”

MT Evaluation

- **Drawbacks of n-gram based metrics:**
  - Penalize semantically-correct paraphases due to string matching (Banerjee & Lavie, 2005)
  - Fail to capture distant dependencies and penalize semantically-critical ordering changes or word drops(Isozaki et al., 2010)
- **Recent Proposed Metrics: contextualized embedding based metrics**
  - BERTScores (Zhang et al., 2020)
    - Computes token similarity using contextual embedding between candidate and reference
  - BLEURT (Stellam et al., 2020)
    - A learned evaluation metric based on BERT