LLM 101

一起入门大语言模型

https://llm101.top

哔哩哔哩@一万篇论文笔记

一万篇论文笔记

微信扫描二维码，关注我的公众号

2024.12.04

Transformer模型

- 回顾机器翻译任务的发展历史
  - 统计机器翻译(SMT)、Encoder-Decoder结构、注意力(Attention)机制、BPE算法
- Transformer模型
- 编程实践
  - RNN Encoder-Decoder with Attention、The Annotated Transformer、基于OpenNMT和Transformer训练翻译模型
  - 非代码：斯坦福CS224N 作业4 Attention和Position Embeddings分析

翻译

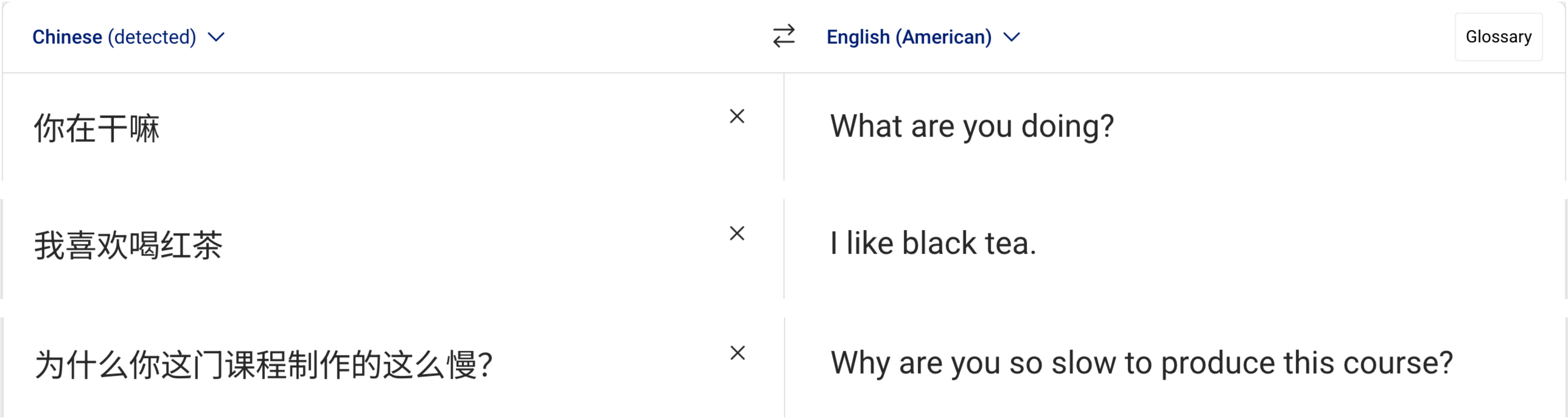
ENGLISH
RUSSIAN
CHINESE
PUNJABI
JAPANESE
ARABIC
HINDI
BENGALI
SPANISH
PORTUGUESE
THE
MOST
SPOKEN
LANGUAGES
IN THE WORLD

图片链接

玄奘

机器翻译

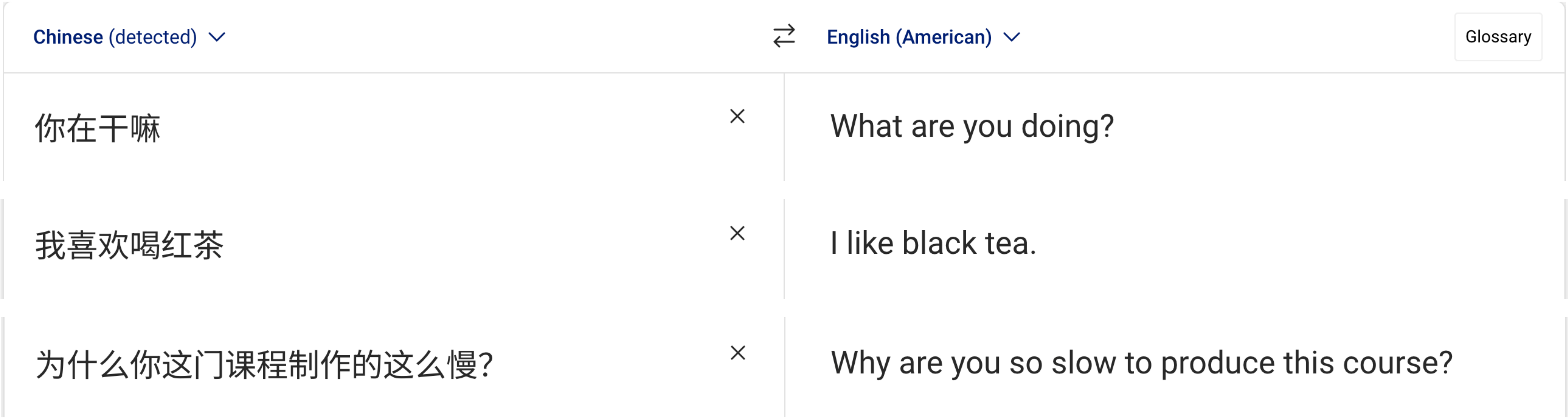
- 机器翻译(Machine Translation, MT): 研究如何利用计算机把**源语言**(source language) 翻译成**目标语言**(target language)



来自DeepL

机器翻译

- 机器翻译(Machine Translation, MT): 研究如何利用计算机把**源语言**(source language) 翻译成**目标语言**(target language)



来自DeepL

- 机器翻译的特点：
  - 输入序列和输出序列的长度无关， 单词之间不是按顺序一一对应 ===> **两个变长序列**
  - 包含语言模型，连接NLU和NLG

机器翻译

- 机器翻译(Machine Translation, MT): 研究如何利用计算机把**源语言**(source language) 翻译成**目标语言**(target language)

Chinese (detected)	English (American)
你在干嘛	What are you doing?
我喜欢喝红茶	I like black tea.
为什么你这门课程制作的这么慢?	Why are you so slow to produce this course?

Glossary

来自DeepL

- 机器翻译的特点：

sequence to sequence / seq2seq 区分 **sequence labeling**

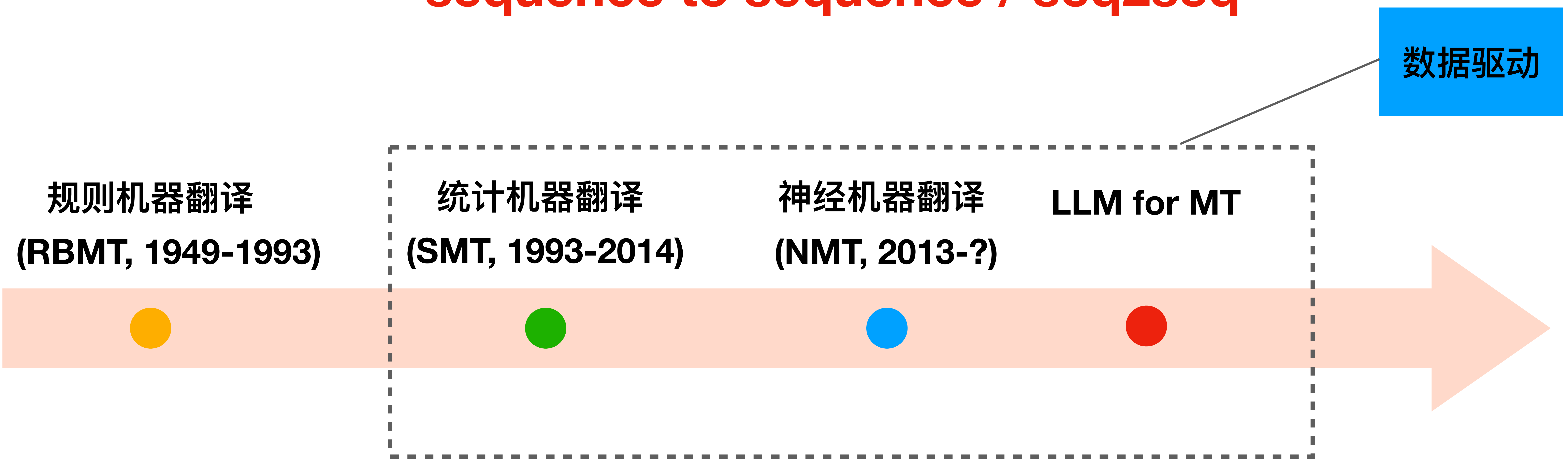
  - 输入序列和输出序列的长度无关，单词之间不是按顺序一一对应 ===> **两个变长序列**
  - 包含语言模型，连接NLU和NLG

Encoder-Decoder、Attention、Transformer、subword

机器翻译

- 机器翻译(Machine Translation, MT): 研究如何利用计算机把一种语言(**源语言**, source language) 翻译成另一种语言(**目标语言**，target language)

sequence to sequence / seq2seq

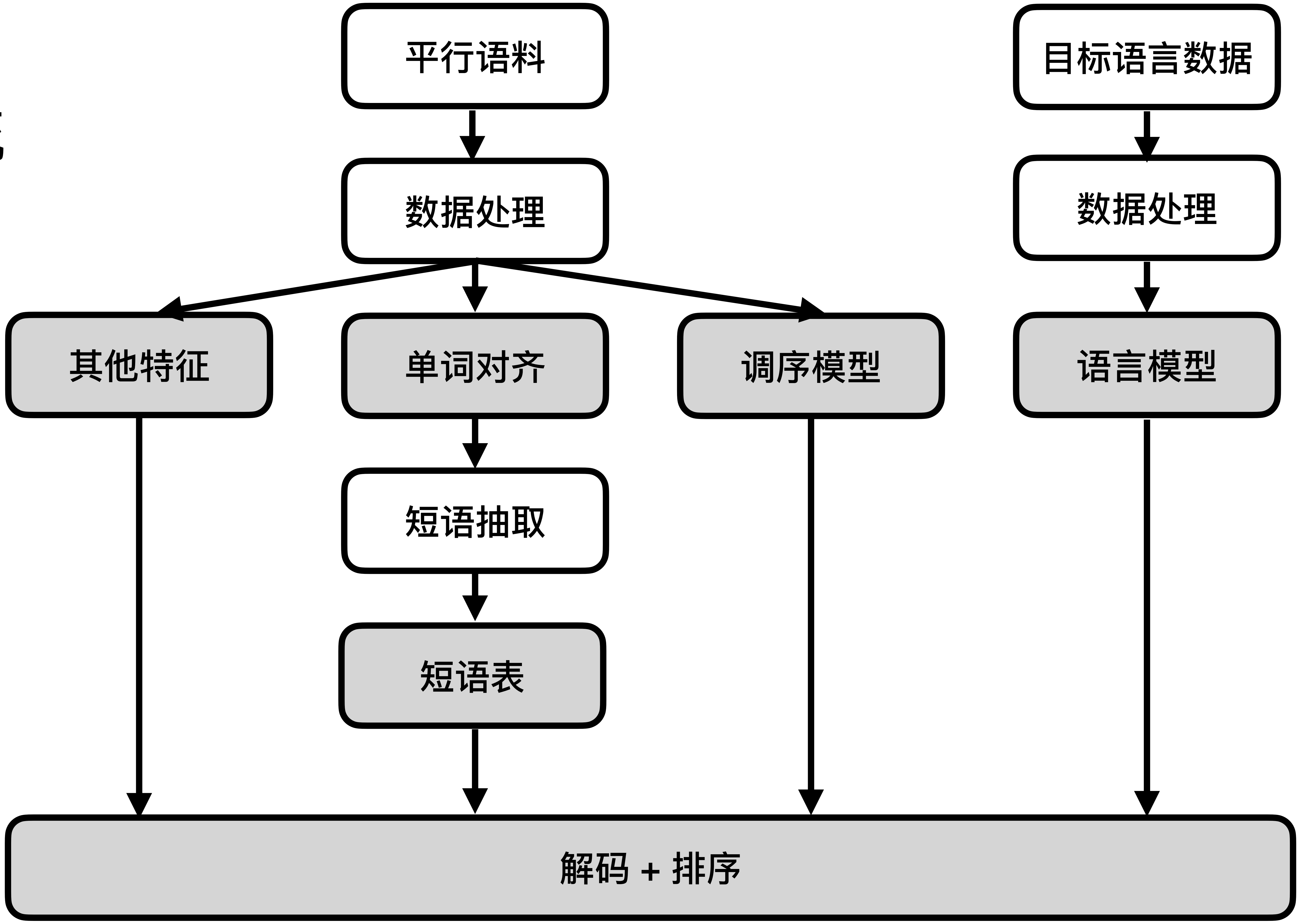


机器翻译进展综述

统计机器翻译(SMT)

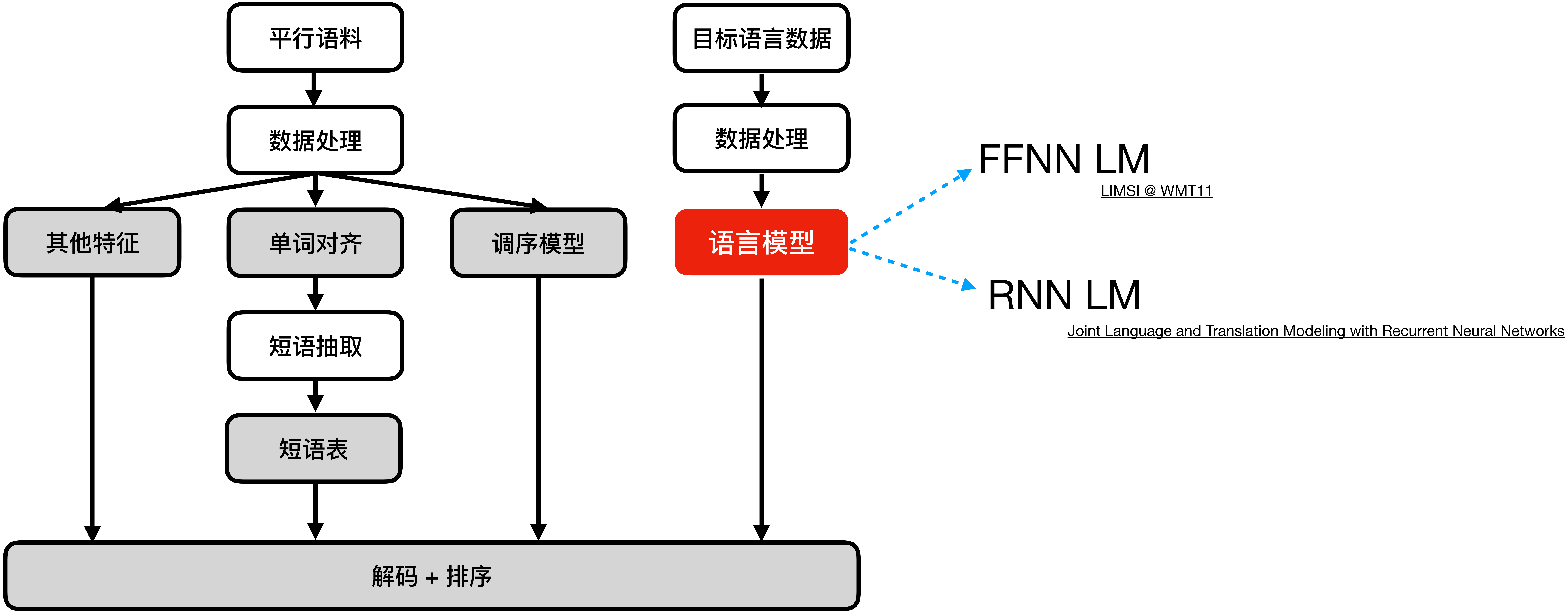
- 基于短语(phrase)的统计翻译系统
- 词对齐(word alignment)
- 短语表(phrase table)
  - p(black tea|冰红茶)=0.7
- 语言模型: N-gram LM
- Log-linear

$$log\ p(y\,|\,x) = \sum_i \lambda_i h_i(x, y)$$



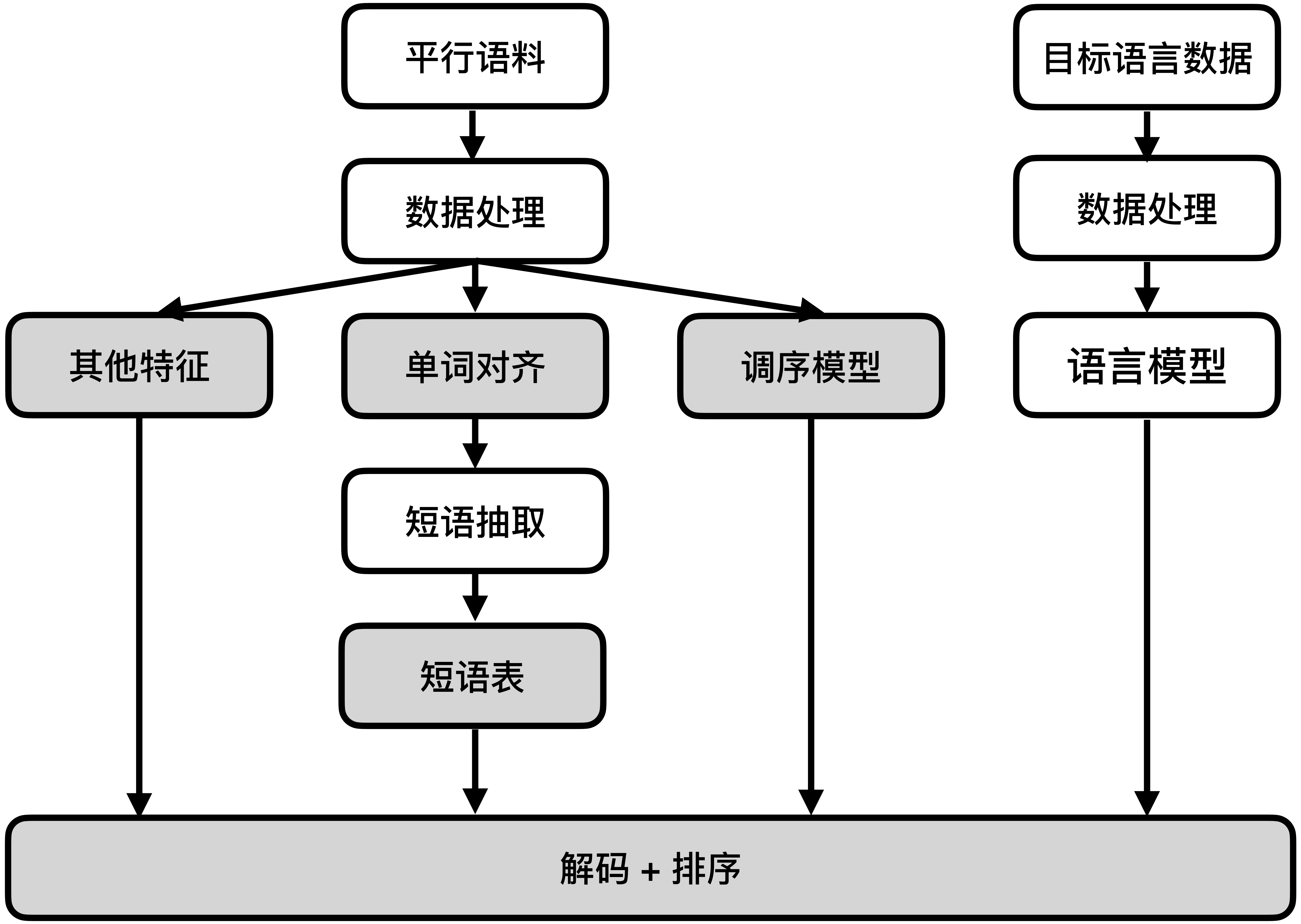
参考 Moses

SMT —> NMT



参考 Moses

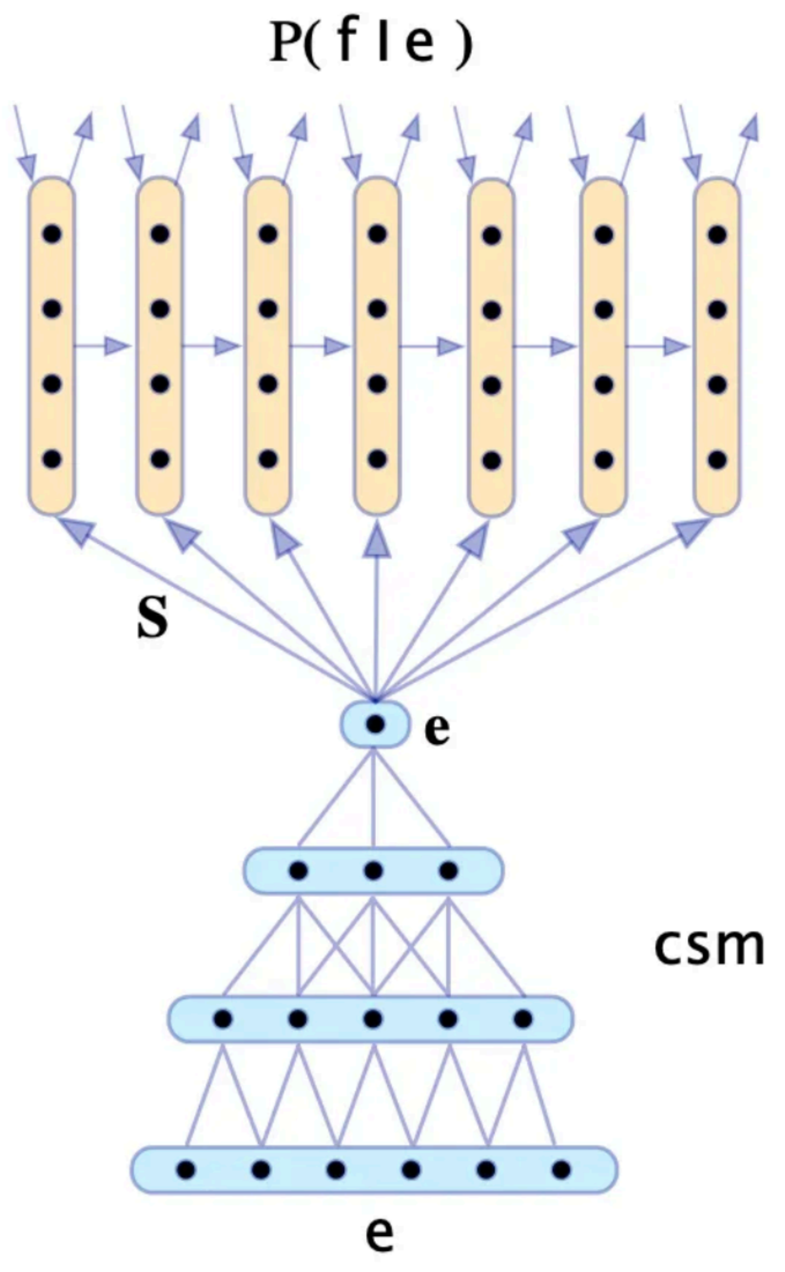
SMT —> NMT



参考 Moses

sequence to sequence/seq2seq task

Encoder-Decoder Model



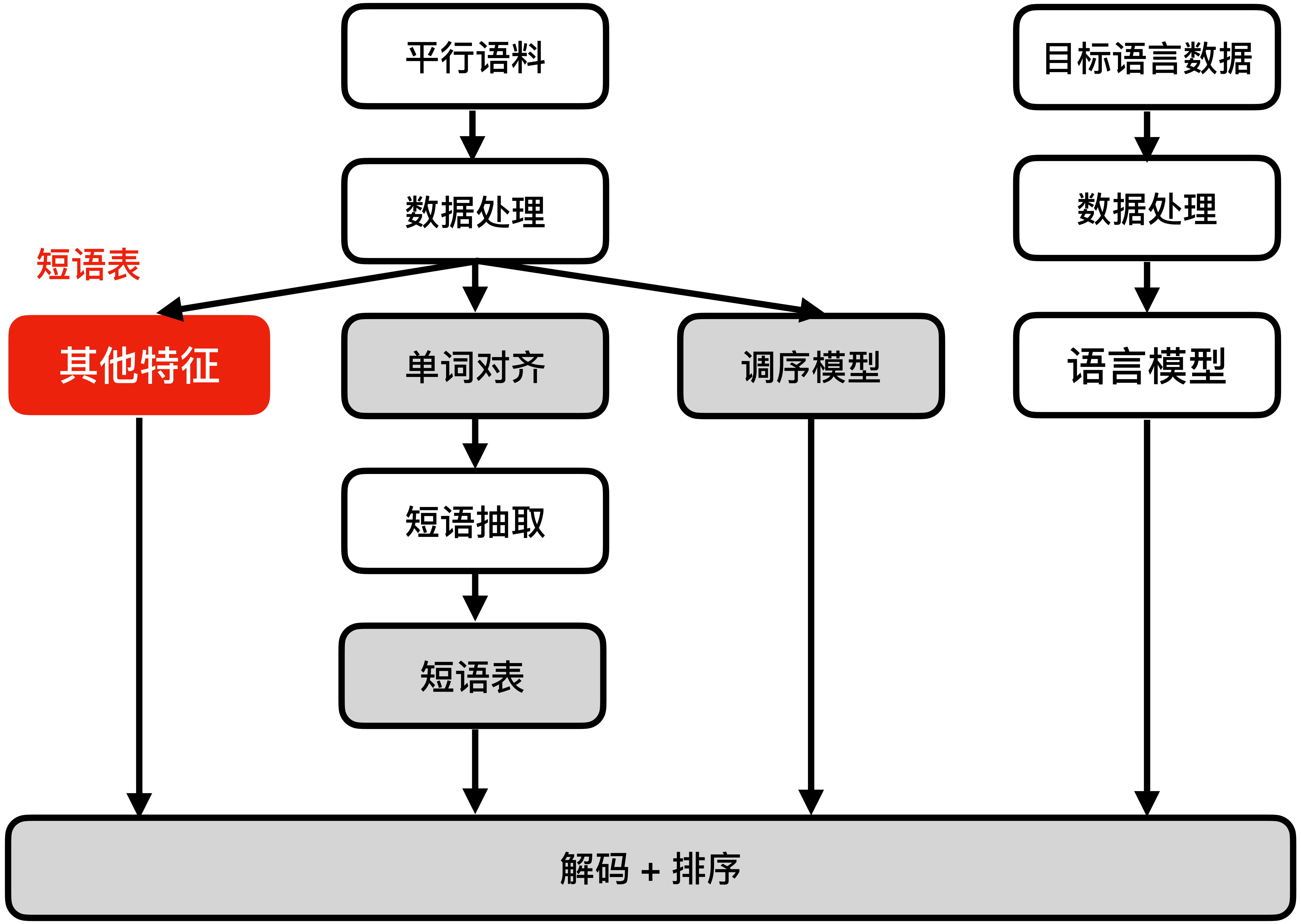
Decoder: RNN

$$\mathbf{s} = \mathbf{S} \cdot \mathsf{csm}(\mathsf{e})$$
$$h_1 = \sigma(\mathbf{I} \cdot \mathsf{v}(\mathsf{f}_1) + \mathbf{s})$$
$$h_{i+1} = \sigma(\mathbf{R} \cdot h_i + \mathbf{I} \cdot \mathsf{v}(\mathsf{f}_{i+1}) + \mathbf{s})$$
$$o_{i+1} = \mathbf{O} \cdot h_i$$

Encoder: CNN

Recurrent Continuous Translation Models, 2013

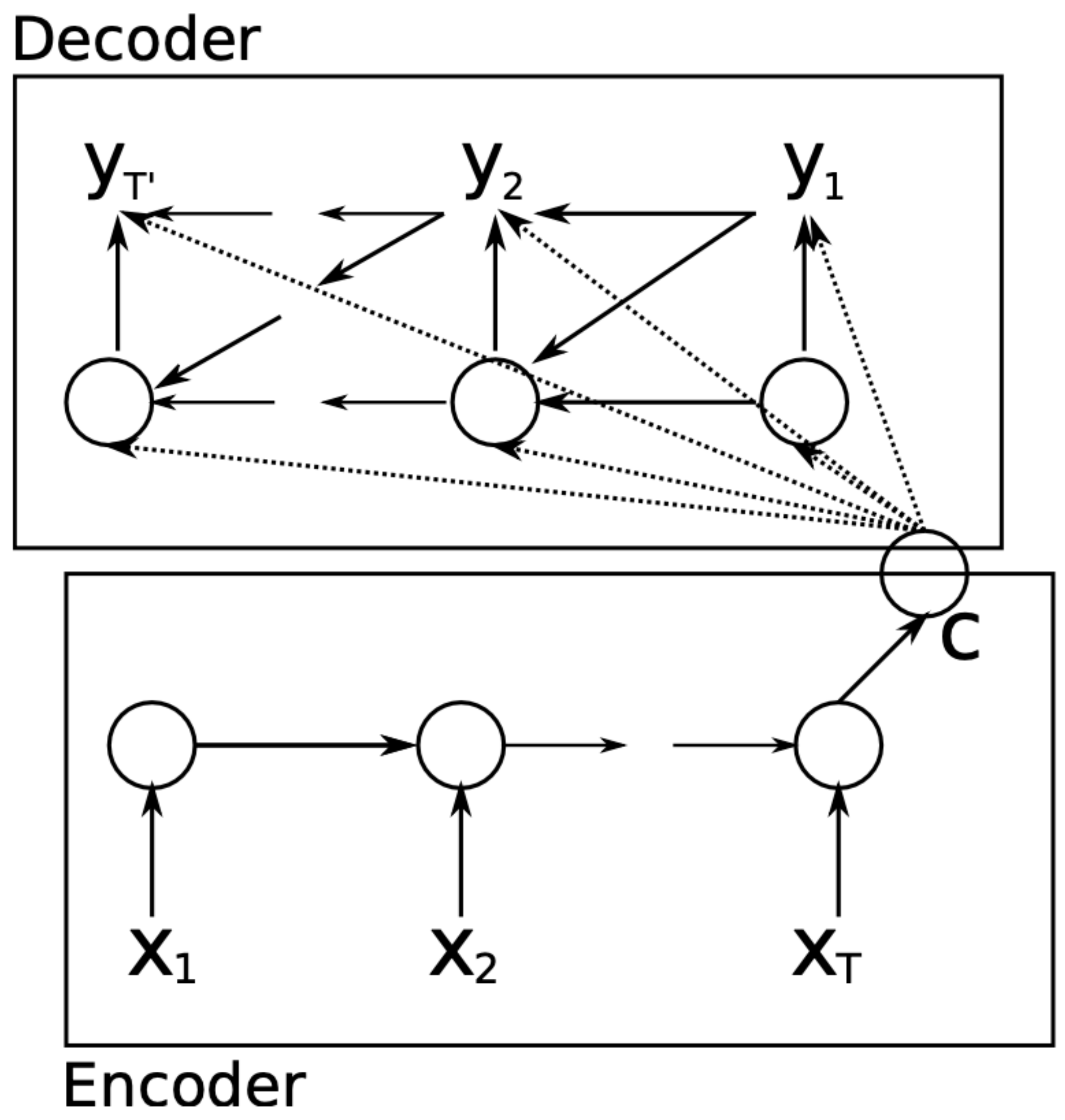
SMT —> NMT



参考 Moses

sequence to sequence/seq2seq task

Encoder-Decoder Model

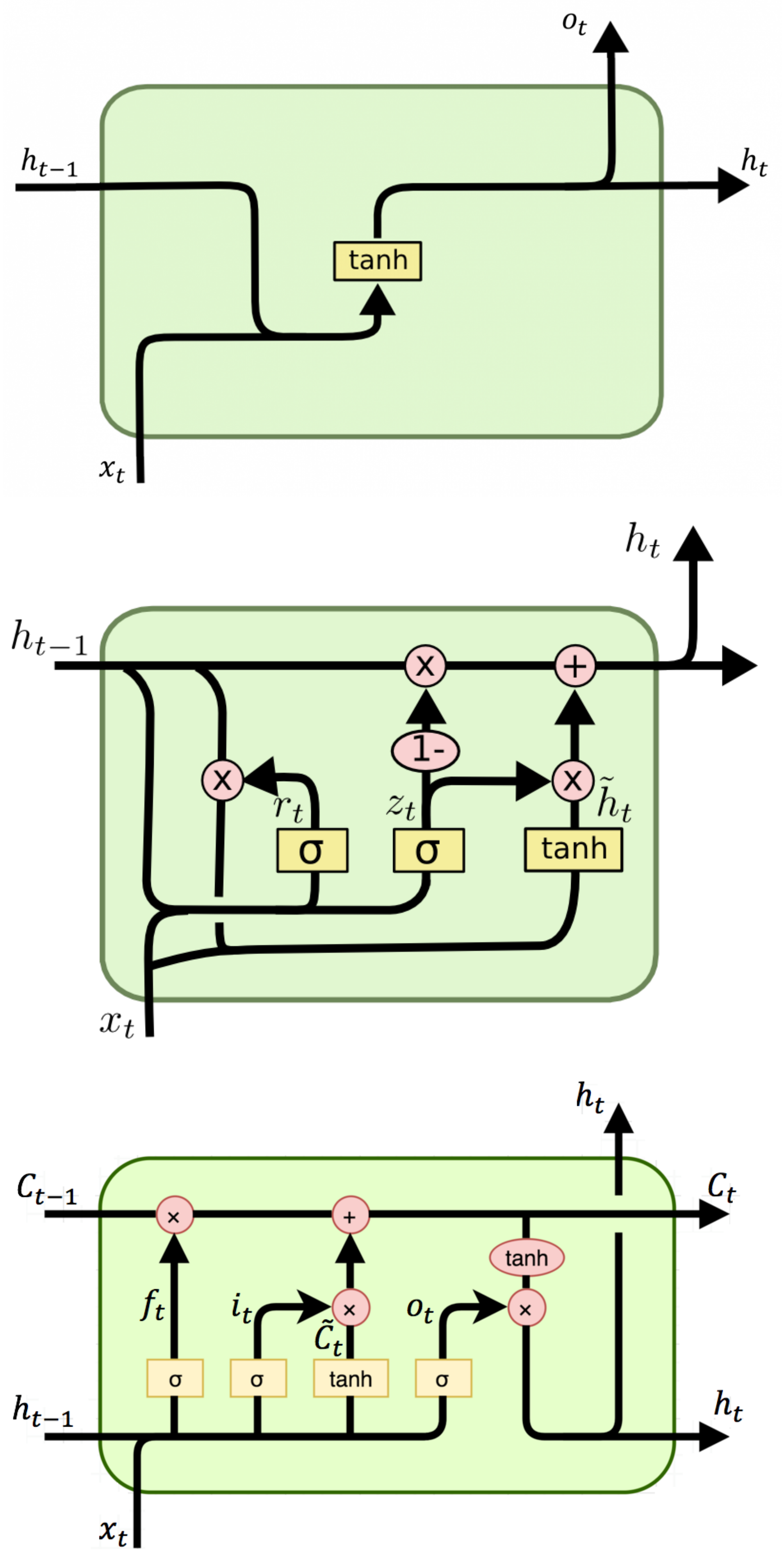


Decoder: GRU

Encoder: GRU

Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014

RNN GRU LSTM



图片链接

RNN

$$h_t = \tanh(x_t W_{ih}^T + b_{ih} + h_{t-1} W_{hh}^T + b_{hh})$$

GRU

$$\begin{aligned} r_t &= \sigma(W_{ir} x_t + b_{ir} + W_{hr} h_{(t-1)} + b_{hr}) \\ z_t &= \sigma(W_{iz} x_t + b_{iz} + W_{hz} h_{(t-1)} + b_{hz}) \\ n_t &= \tanh(W_{in} x_t + b_{in} + r_t \odot (W_{hn} h_{(t-1)} + b_{hn})) \\ h_t &= (1 - z_t) \odot n_t + z_t \odot h_{(t-1)} \end{aligned}$$

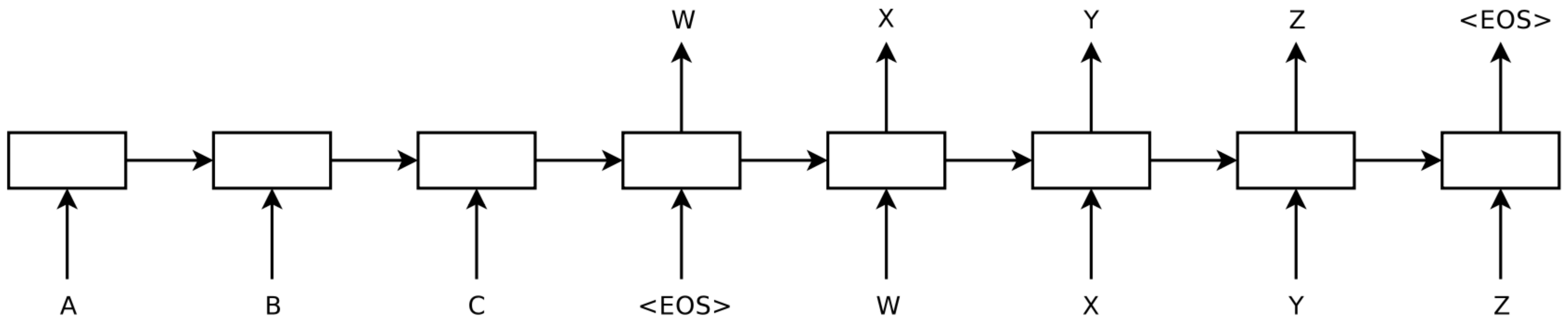
LSTM

$$\begin{aligned} i_t &= \sigma(W_{ii} x_t + b_{ii} + W_{hi} h_{t-1} + b_{hi}) \\ f_t &= \sigma(W_{if} x_t + b_{if} + W_{hf} h_{t-1} + b_{hf}) \\ g_t &= \tanh(W_{ig} x_t + b_{ig} + W_{hg} h_{t-1} + b_{hg}) \\ o_t &= \sigma(W_{io} x_t + b_{io} + W_{ho} h_{t-1} + b_{ho}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned}$$

参数量 1: 3: 4　　参数共享

神经机器翻译(NMT)

LSTM Encoder-Decoder



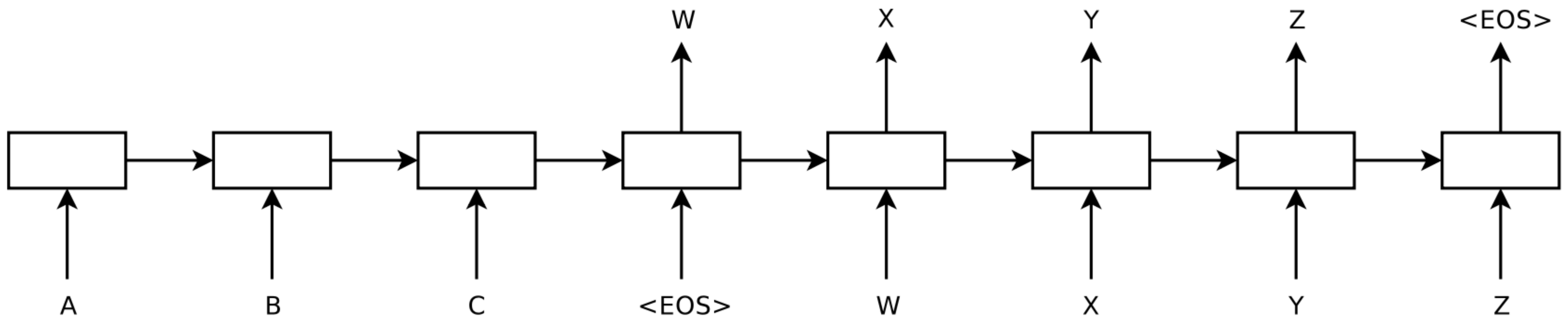
Sequence to Sequence Learning with Neural Networks, 2014

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**

< SMT SOTA

神经机器翻译(NMT)

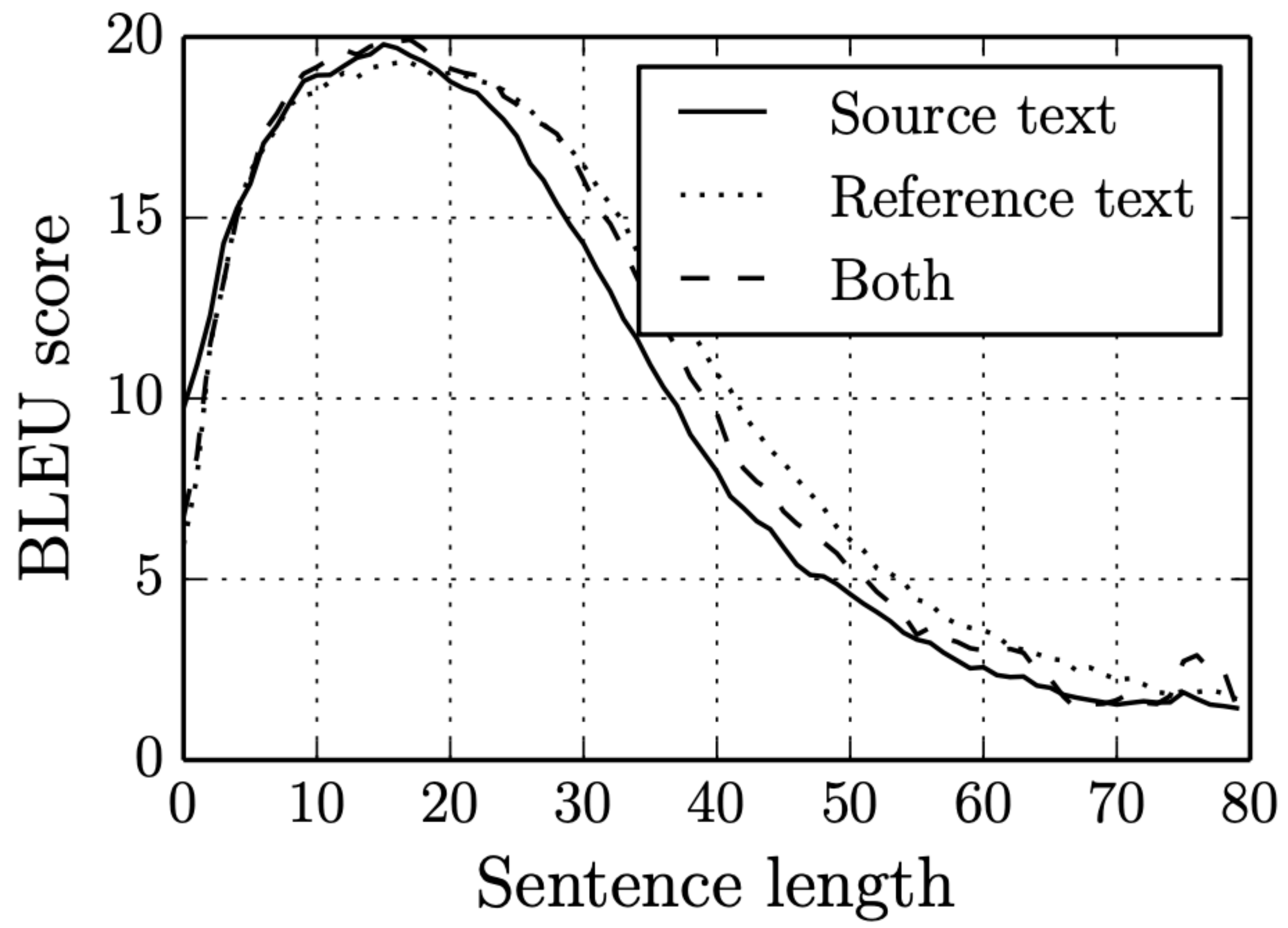
LSTM Encoder-Decoder



Sequence to Sequence Learning with Neural Networks, 2014

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**

< SMT SOTA



On the Properties of Neural Machine Translation Encoder–Decoder Approaches, 2014

RNN Encoder-Decoder with Attention

- Encoder: BiGRU, Decoder: GRU
- 输入序列 $\mathbf{x} = (x_1, \ldots, x_T)$, x_j的向量表示 $h_j = [\overrightarrow{h}_j; \overleftarrow{h}_j]$
- 如何得到第i个输出元素y_i? $p(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}) = ?$

$$e_{ij} = attention(s_{i-1}, h_j)$$
$$= v_a^T \, tanh(W_a s_{i-1} + U_a h_j)$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})}$$

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$z_i = \sigma(W_z e(y_{i-1}) + U_z s_{i-1} + C_z c_i)$$

$$r_i = \sigma(W_r e(y_{i-1}) + U_r s_{i-1} + C_r c_i)$$

$$\tilde{s}_i = tanh(We(y_{i-1}) + U[r_i \circ s_{i-1}] + Cc_i)$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i$$

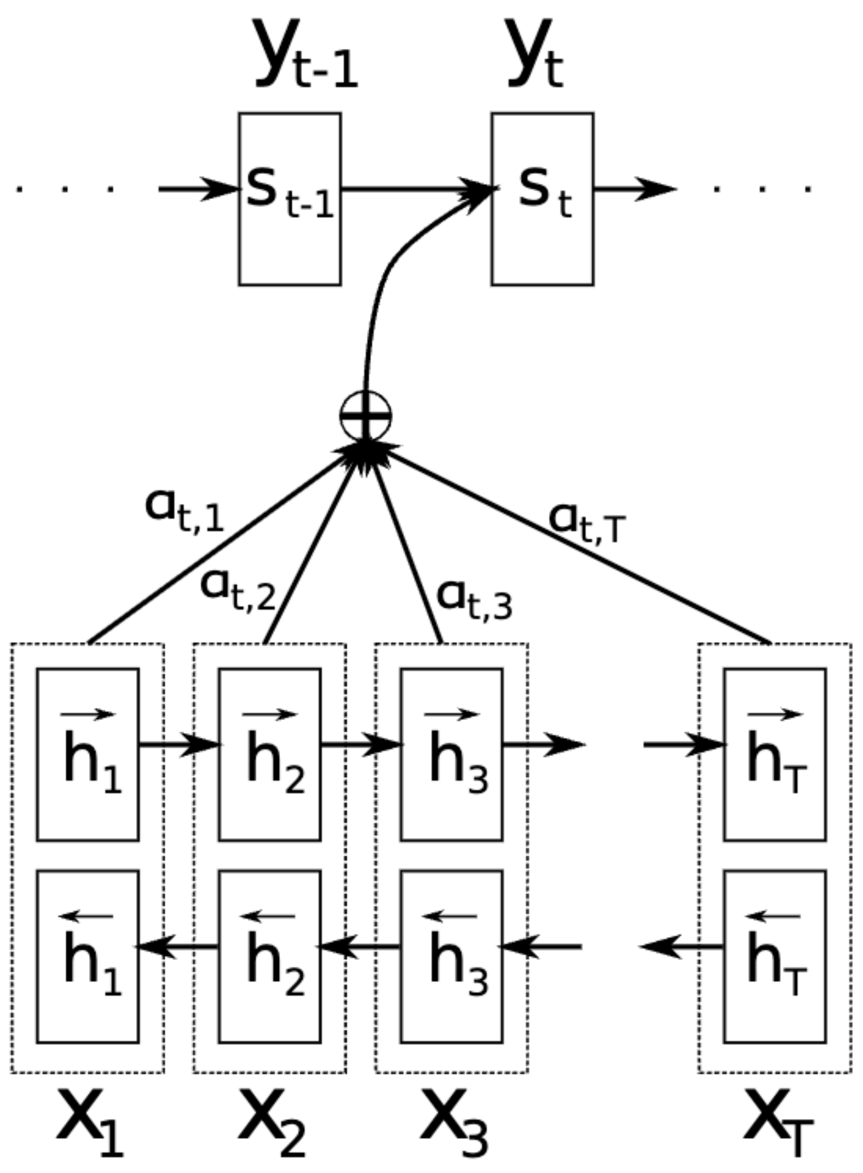


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Neural Machine Translation by Jointly Learning to Align and Translate

RNN Encoder-Decoder with Attention

- Encoder: BiGRU, Decoder: GRU
- 输入序列 $\mathbf{x} = (x_1, \ldots, x_T)$, x_j的向量表示 $h_j = [\overrightarrow{h}_j; \overleftarrow{h}_j]$
- 如何得到第i个输出元素y_i? $p(y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}) = ?$

$$e_{ij} = attention(s_{i-1}, h_j)$$
$$= v_a^T \; tanh(W_a s_{i-1} + U_a h_j)$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})}$$

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$z_i = \sigma(W_z e(y_{i-1}) + U_z s_{i-1} + C_z c_i)$$

$$r_i = \sigma(W_r e(y_{i-1}) + U_r s_{i-1} + C_r c_i)$$

$$\tilde{s}_i = tanh(We(y_{i-1}) + U[r_i \circ s_{i-1}] + Cc_i)$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i$$

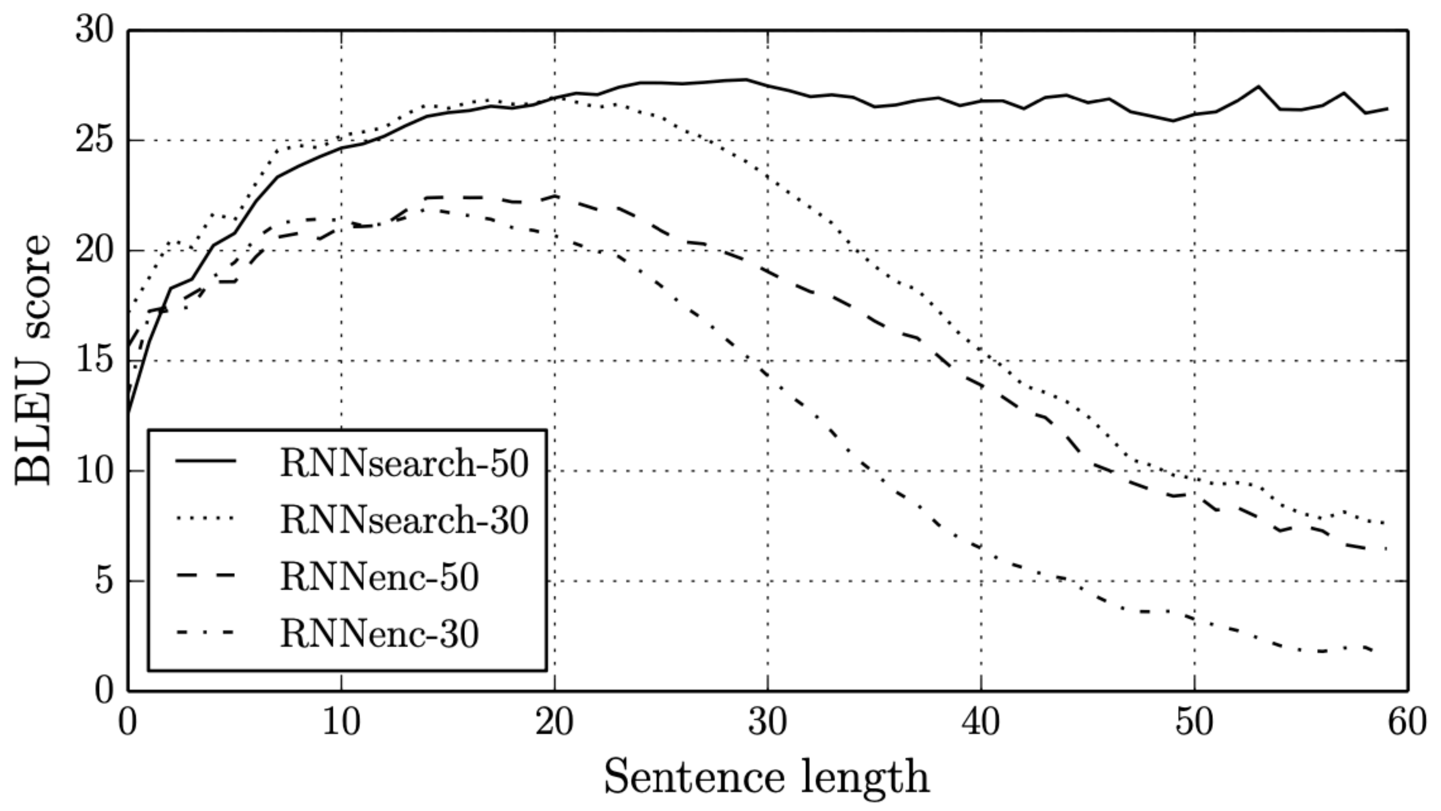
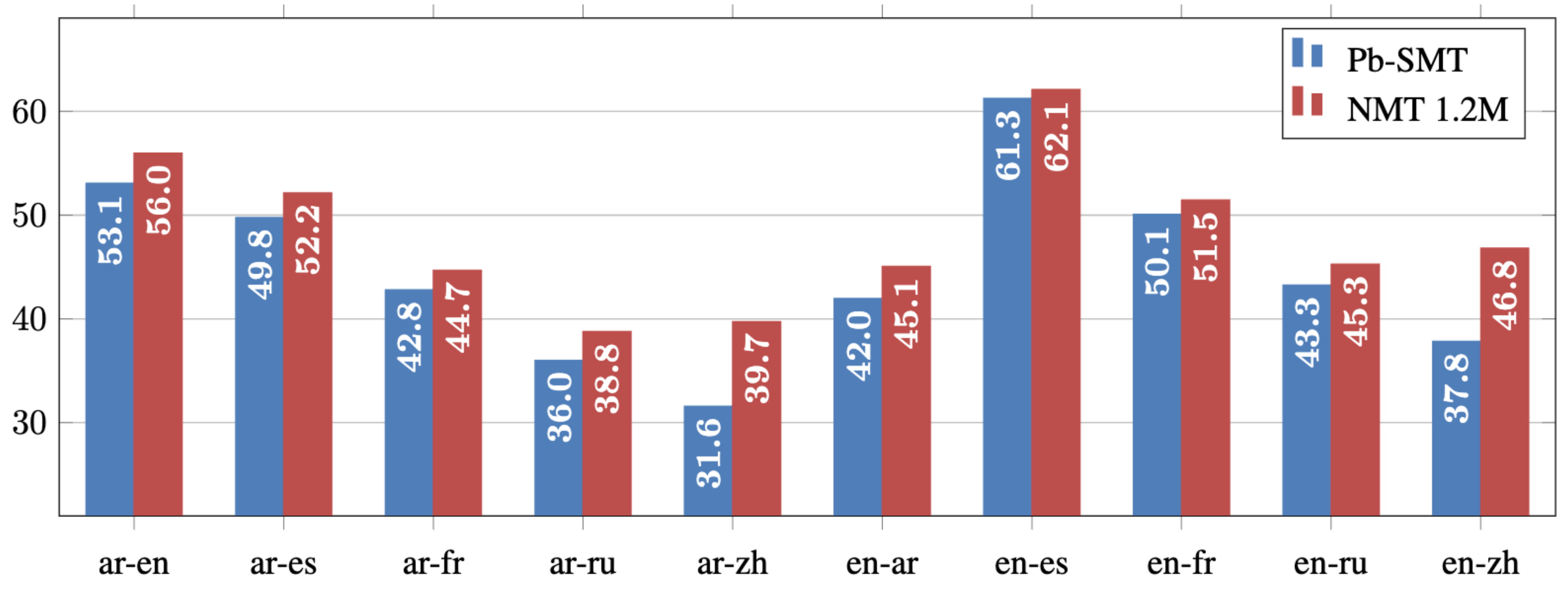


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

Neural Machine Translation by Jointly Learning to Align and Translate

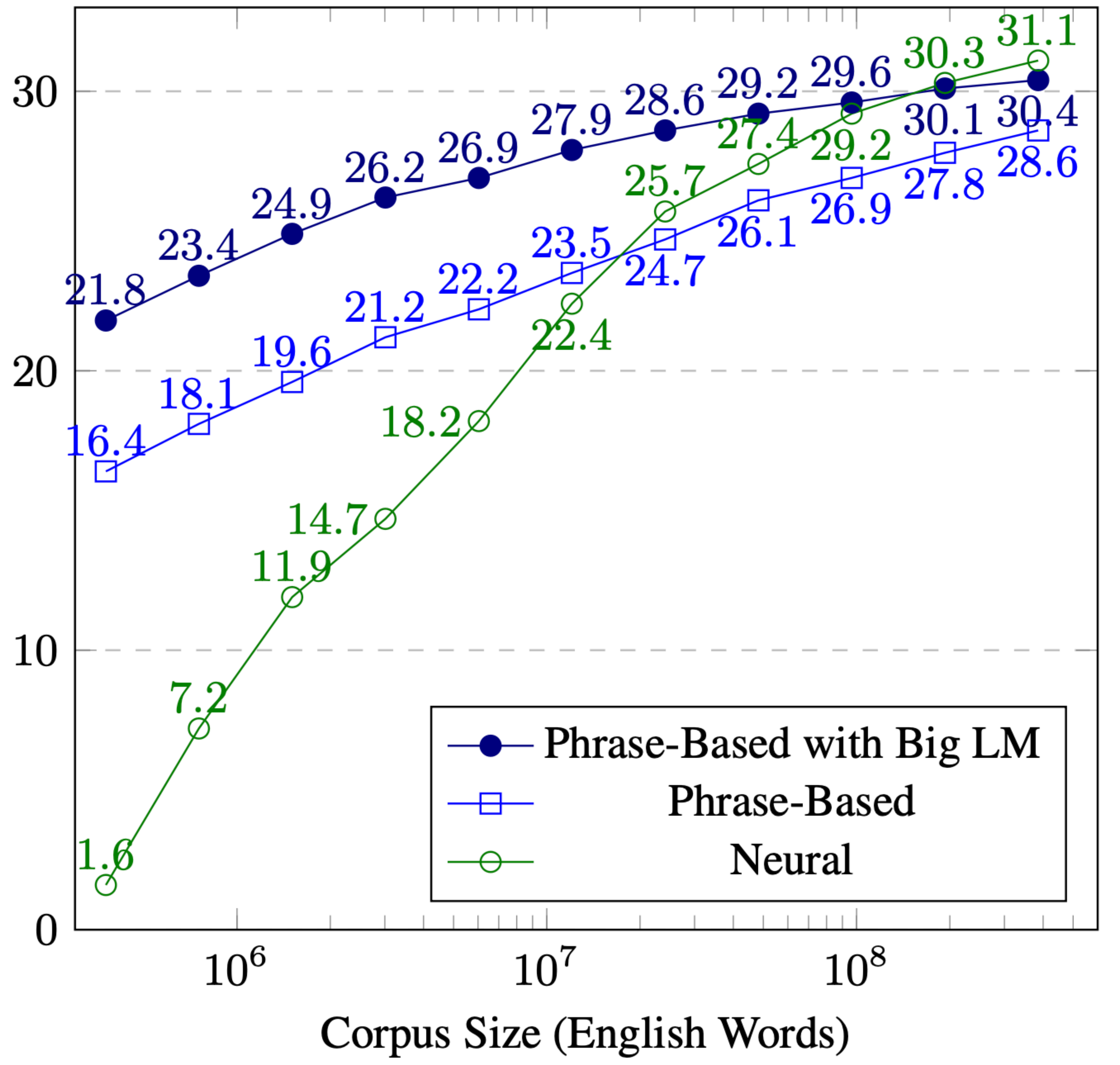
SMT vs NMT

NMT Win!



Is Neural Machine Translation Ready for Deployment?, 2016

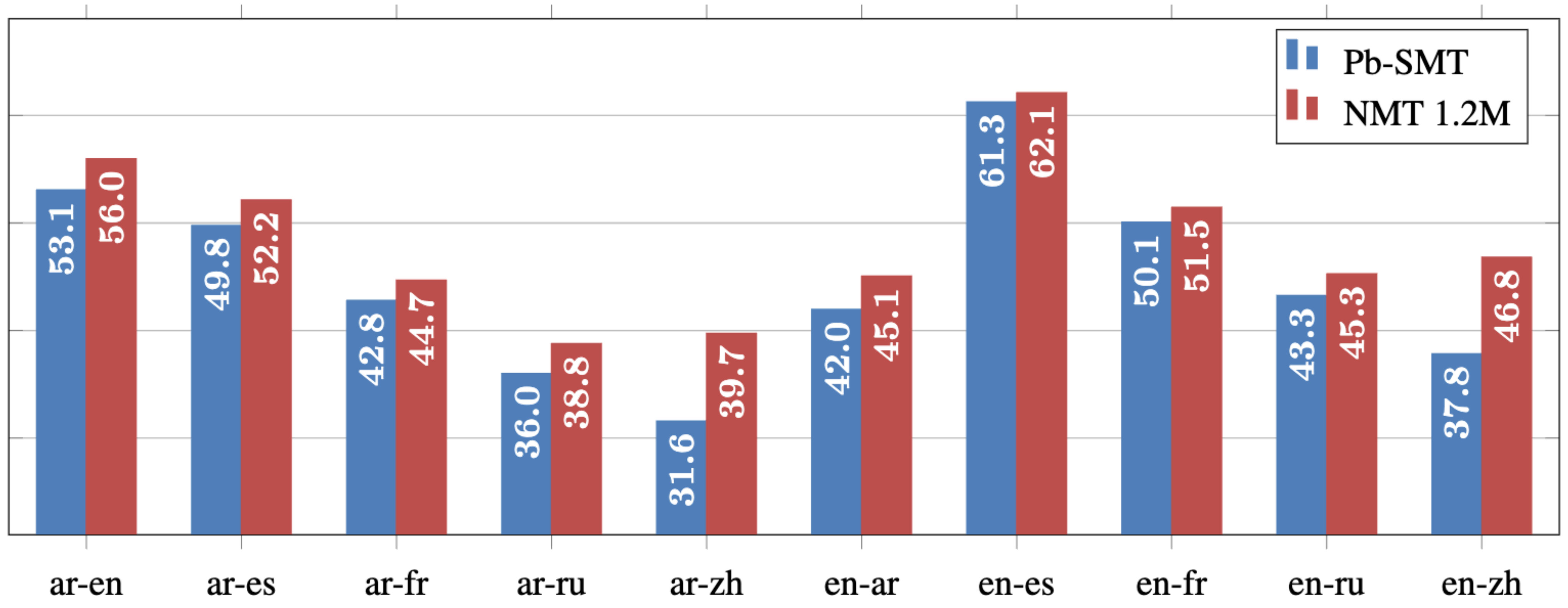
BLEU Scores with Varying Amounts of Training Data



Six Challenges for Neural Machine Translation

SMT vs NMT

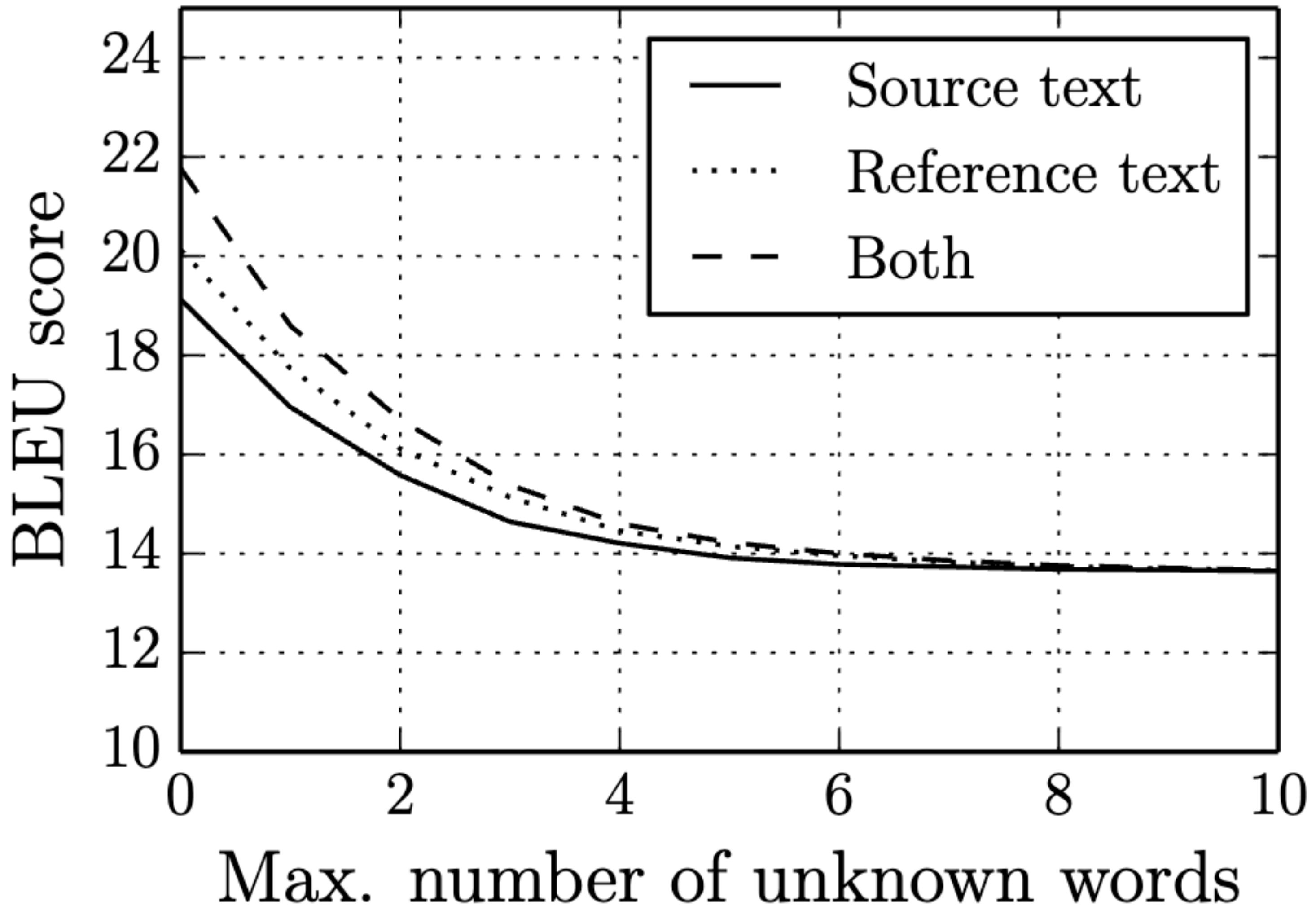
NMT Win!



Is Neural Machine Translation Ready for Deployment?, 2016

OOV(Out of Vocabulary)

未登录词



[UNK]

On the Properties of Neural Machine Translation Encoder–Decoder Approaches, 2014

BPE: Byte Pair Encoding

- 分词，统计单词词频
- 将单词看作字符序列
- 单词内的最高频字符组合 进行合并(merge)操作

```
{'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')

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

Algorithm 1 Learn BPE operations

```python
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

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}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

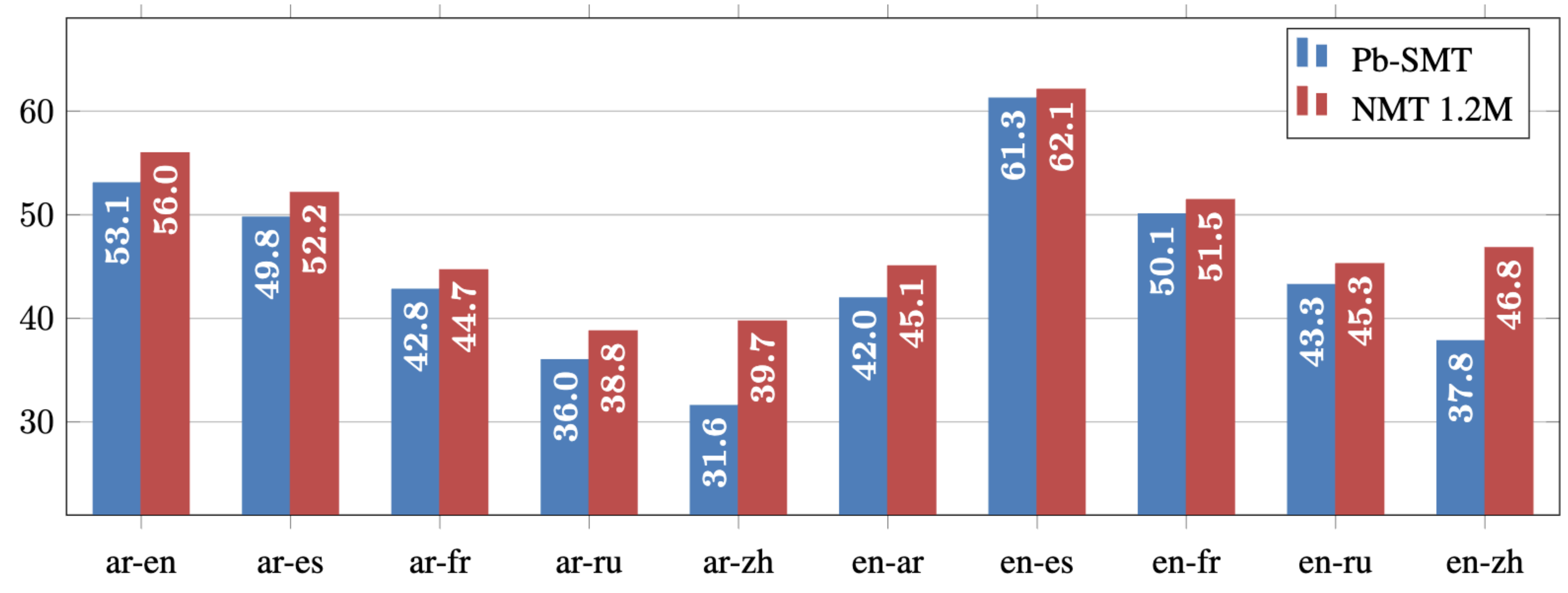
r ·	→	r·
l o	→	lo
lo w	→	low
e r·	→	er·

Neural Machine Translation of Rare Words with Subword Units， 2015

OpenAI Tokenizer

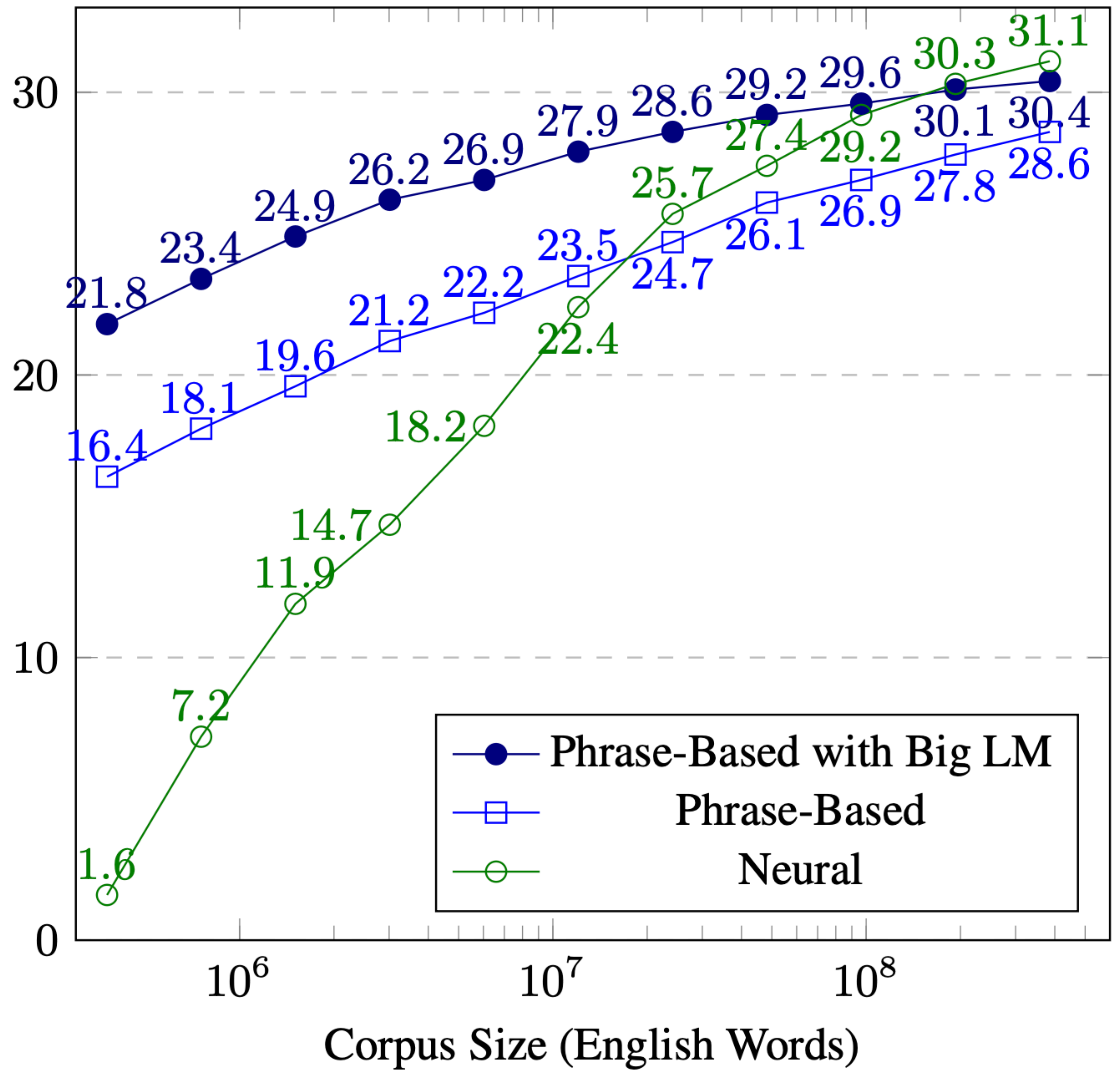
SMT vs NMT

NMT Win!



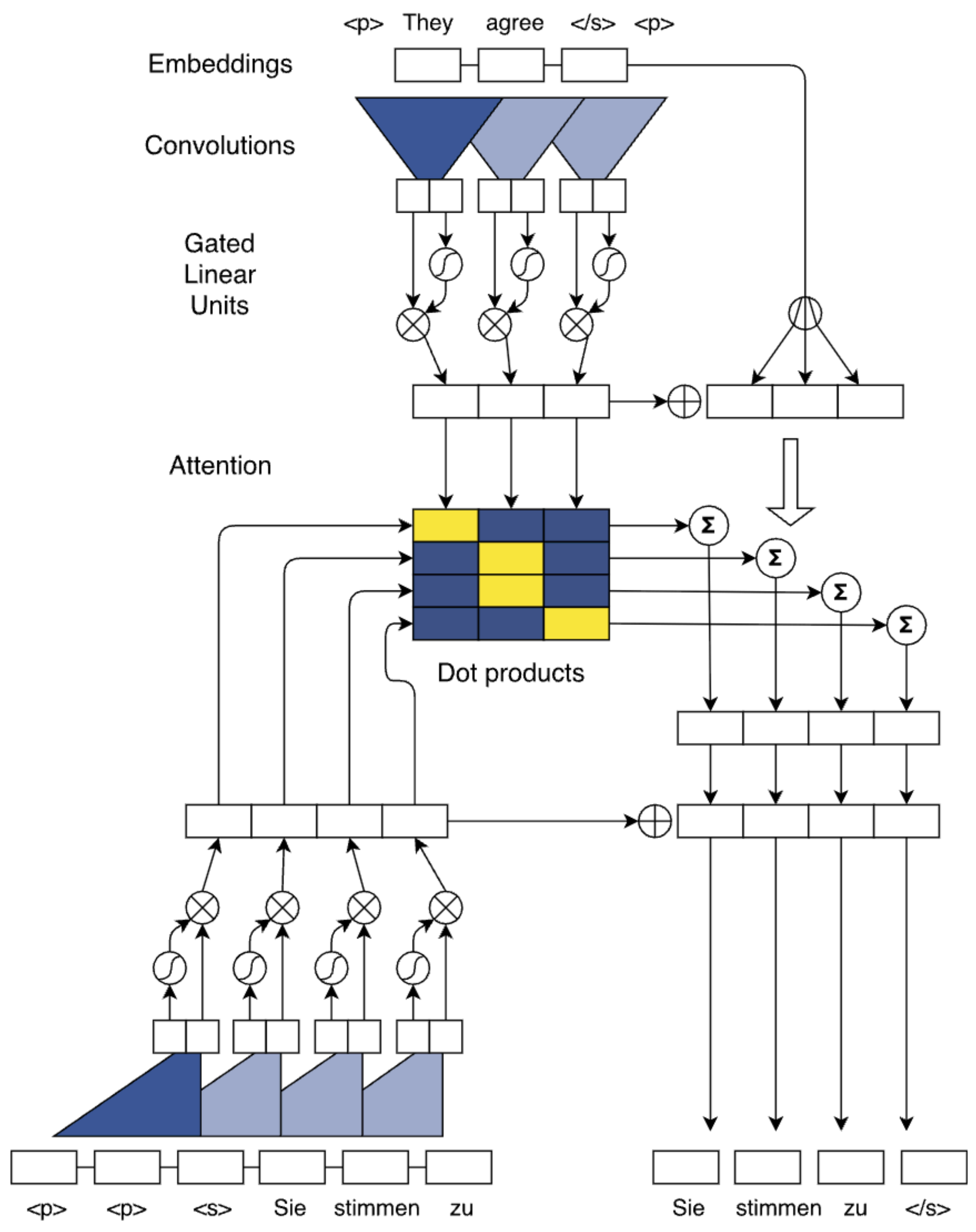
Is Neural Machine Translation Ready for Deployment?, 2016

BLEU Scores with Varying Amounts of Training Data



Six Challenges for Neural Machine Translation

ConvS2S



Convolutional Sequence to Sequence Learning, 1705

WMT'16 English-Romanian	BLEU
Sennrich et al. (2016b) GRU (BPE 90K)	28.1
ConvS2S (Word 80K)	29.45
ConvS2S (BPE 40K)	30.02

WMT'14 English-German	BLEU
Luong et al. (2015) LSTM (Word 50K)	20.9
Kalchbrenner et al. (2016) ByteNet (Char)	23.75
Wu et al. (2016) GNMT (Word 80K)	23.12
Wu et al. (2016) GNMT (Word pieces)	24.61
ConvS2S (BPE 40K)	25.16

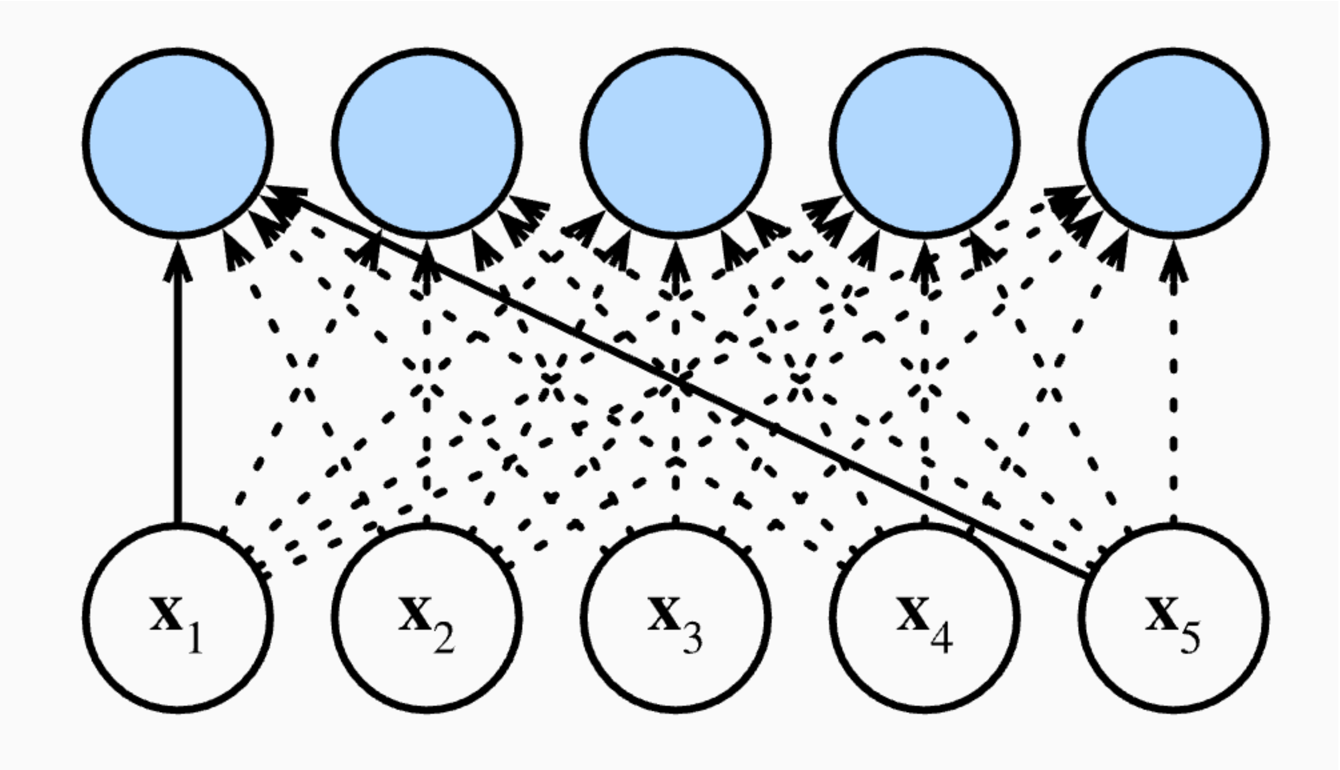
WMT'14 English-French	BLEU
Wu et al. (2016) GNMT (Word 80K)	37.90
Wu et al. (2016) GNMT (Word pieces)	38.95
Wu et al. (2016) GNMT (Word pieces) + RL	39.92
ConvS2S (BPE 40K)	40.51

Table 1. Accuracy on WMT tasks comapred to previous work. ConvS2S and GNMT results are averaged over several runs.

既生瑜何生亮

Transformer

- 自注意力(self-attention)



图片来源

- Encoder-Decoder 结构

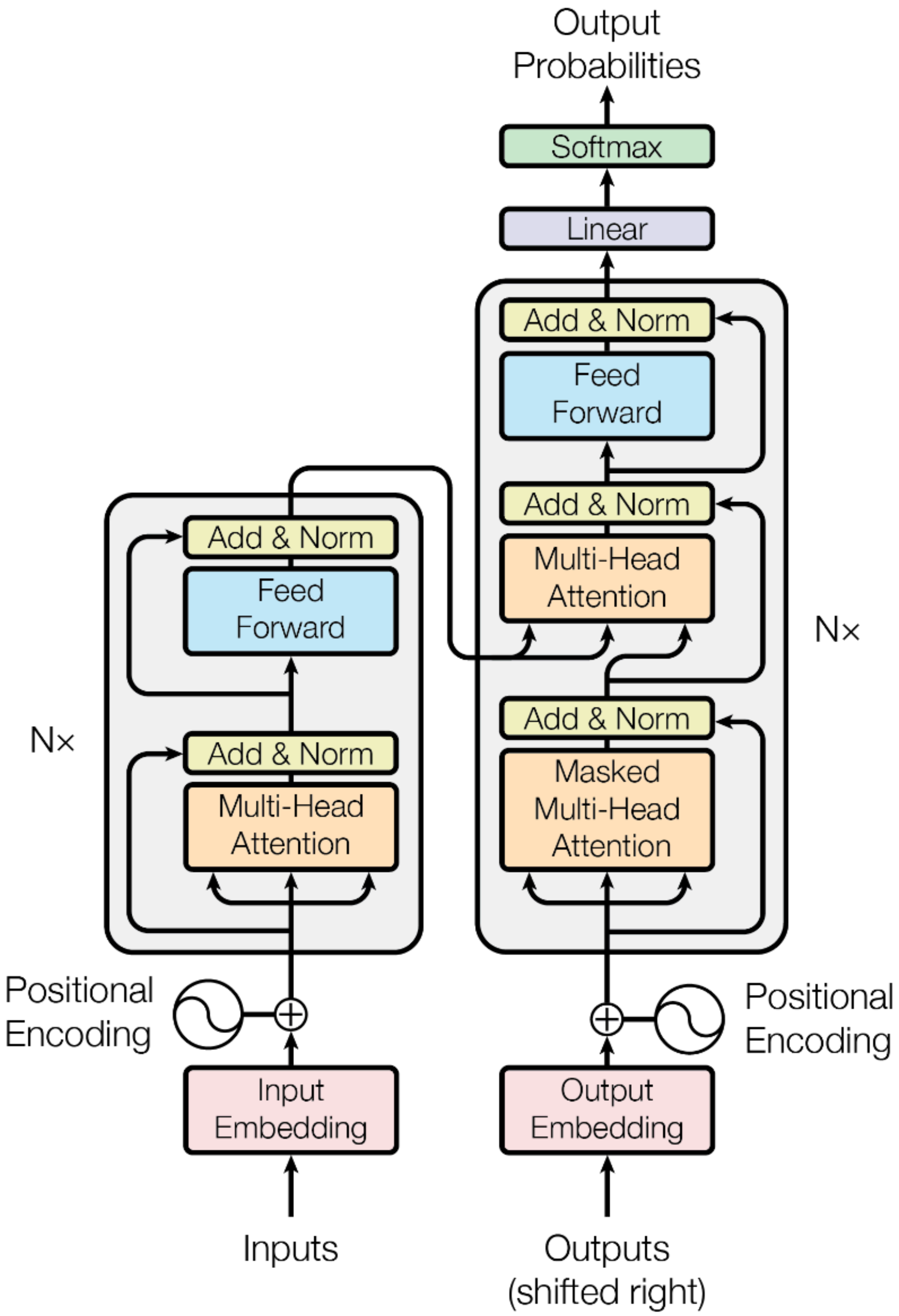


Figure 1: The Transformer - model architecture.

Attention Is All You Need

Transformer

- Encoder Layer: MHSA + FFN

$$x2 = LayerNorm(x1 + Dropout(MHSA(x1)))$$

$$x3 = LayerNorm(x2 + Dropout(FFN(x2)))$$

- Decoder Layer: MHSA + Encoder-Decoder Attention + FFN

mask

$$x2 = LayerNorm(x1 + Dropout(MHSA(x1)))$$

$$x3 = LayerNorm(x2 + Dropout(MHSA(x2, x_{encoder})))$$

$$x4 = LayerNorm(x3 + Dropout(FFN(x3)))$$

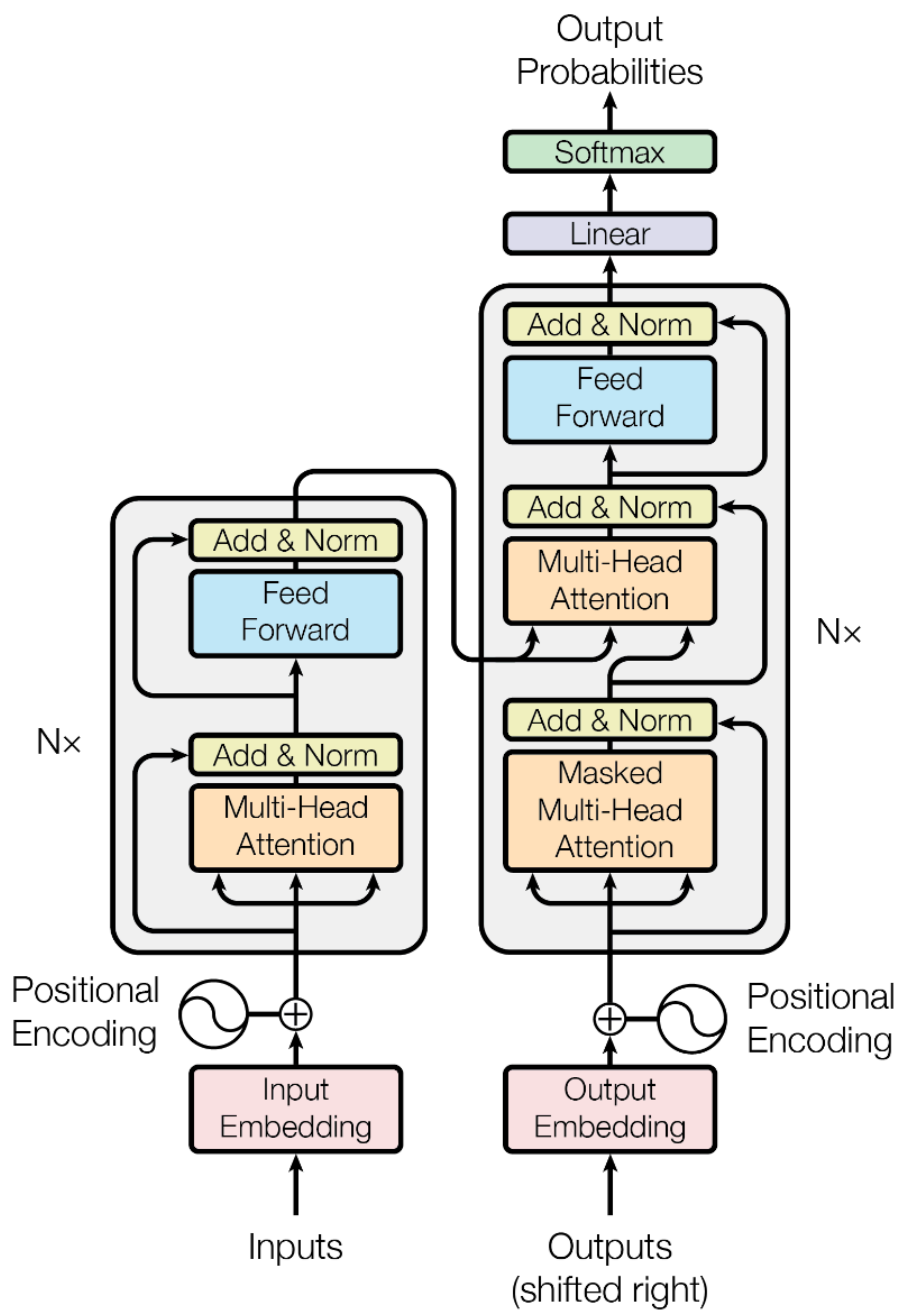


Figure 1: The Transformer - model architecture.

Attention Is All You Need

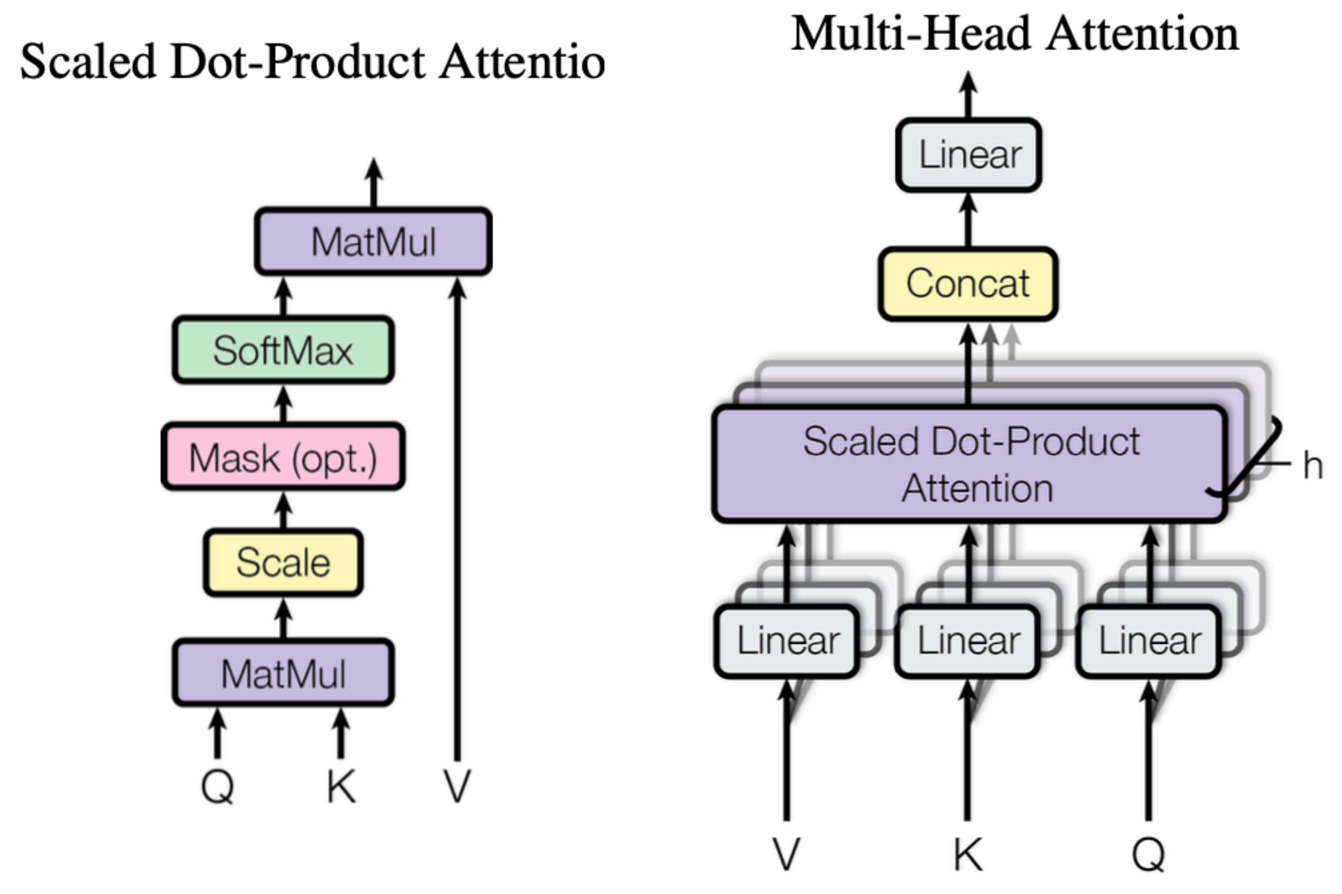
MHSA

- 【定义】注意力：输入是一个query和一堆<key, value>对，输出是**value的线性加权**，权重值由query和key计算得到。其中query、key和value都是向量
- Scaled dot-product attention   code

$$q, k \in R^{d_k}; q \sim N(0,1), k \sim N(0,1)$$

$$q \cdot k = \sum_{i=1}^{d_k} \sim N(0, d_k)$$

$$scaled: \frac{q \cdot k}{\sqrt{d_k}}$$



Attention Is All You Need

MHSA

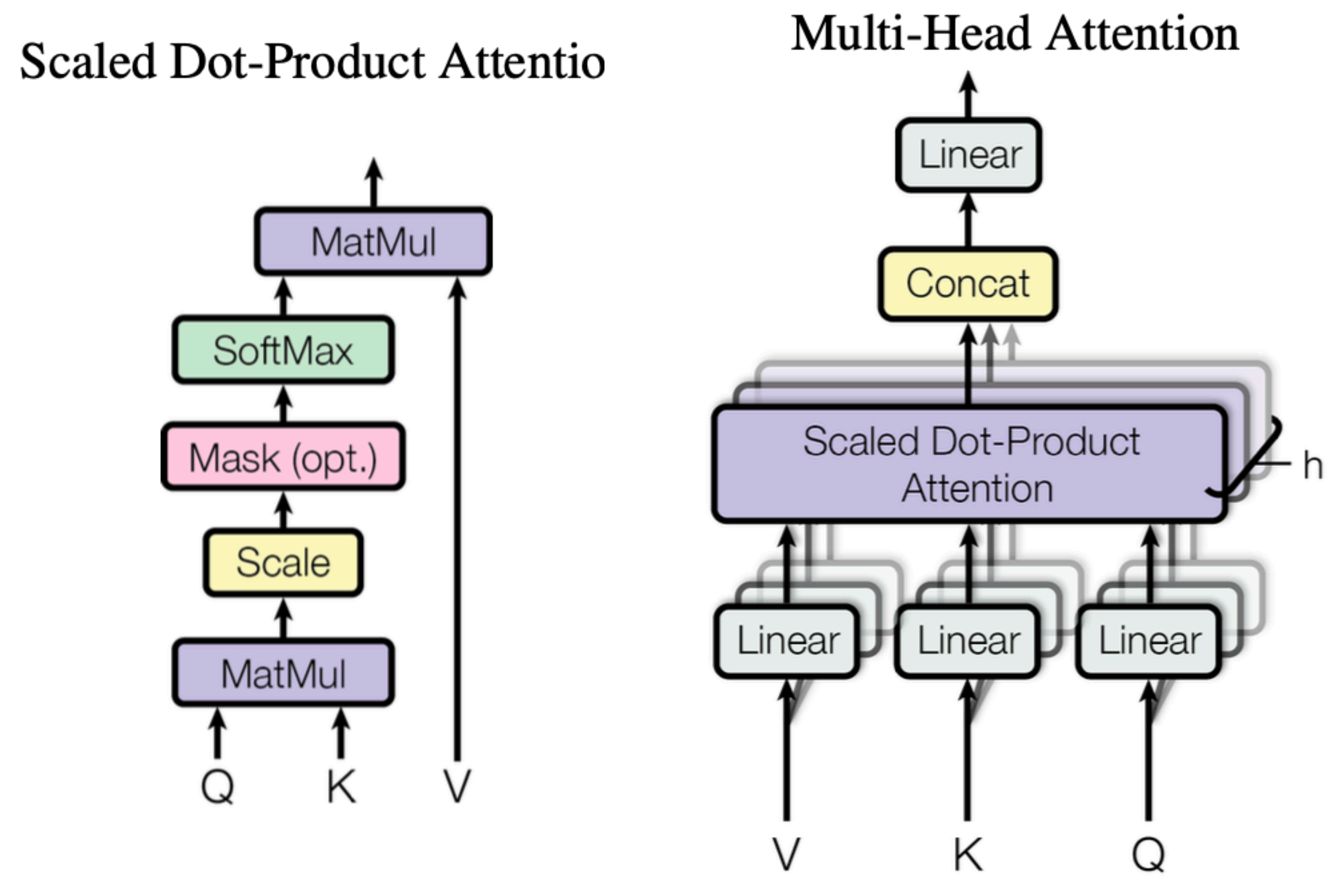
- 【定义】注意力：输入是一个query和一堆<key, value>对，输出是**value的线性加权**，权重值由query和key计算得到。其中query、key和value都是向量

- Scaled dot-product attention

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

- MHSA

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Attention Is All You Need

FFN: Position-wise Feed-Forward Networks

- FFN

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

- MHSA和FFN互补

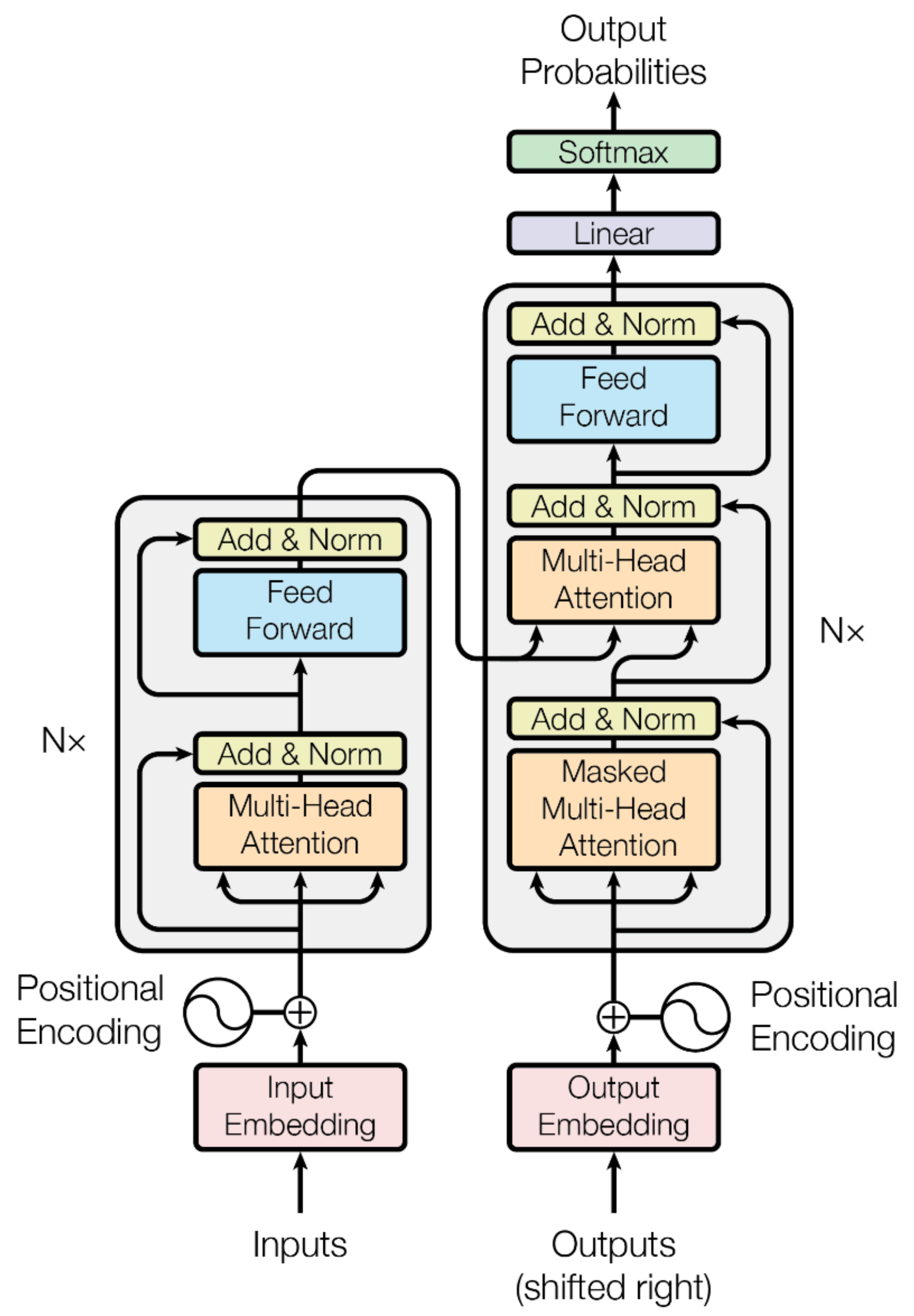


Figure 1: The Transformer - model architecture.

Attention Is All You Need

位置编码 Positional Encoding

- 绝对位置编码：token embedding + positional embedding

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

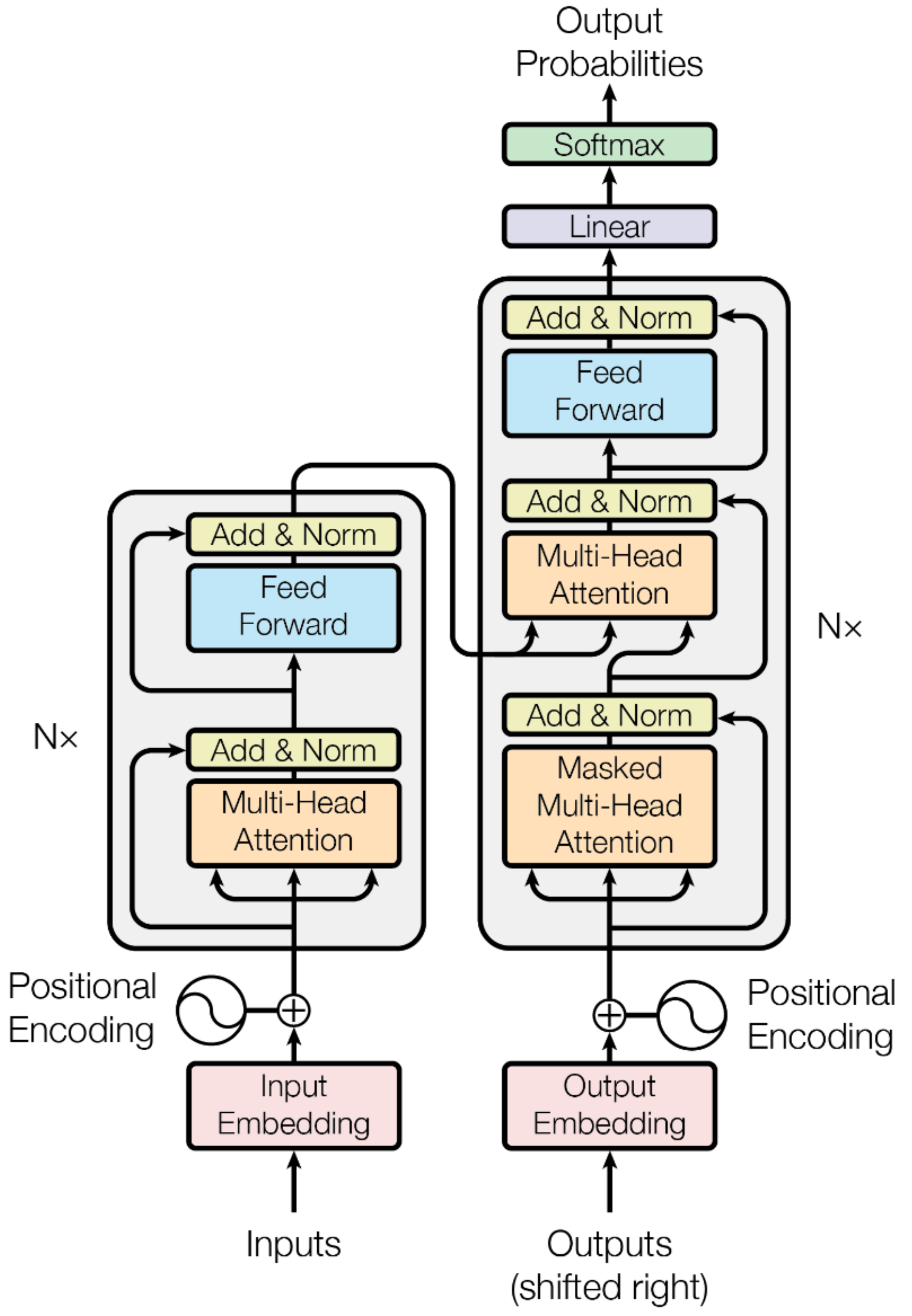


Figure 1: The Transformer - model architecture.

Attention Is All You Need