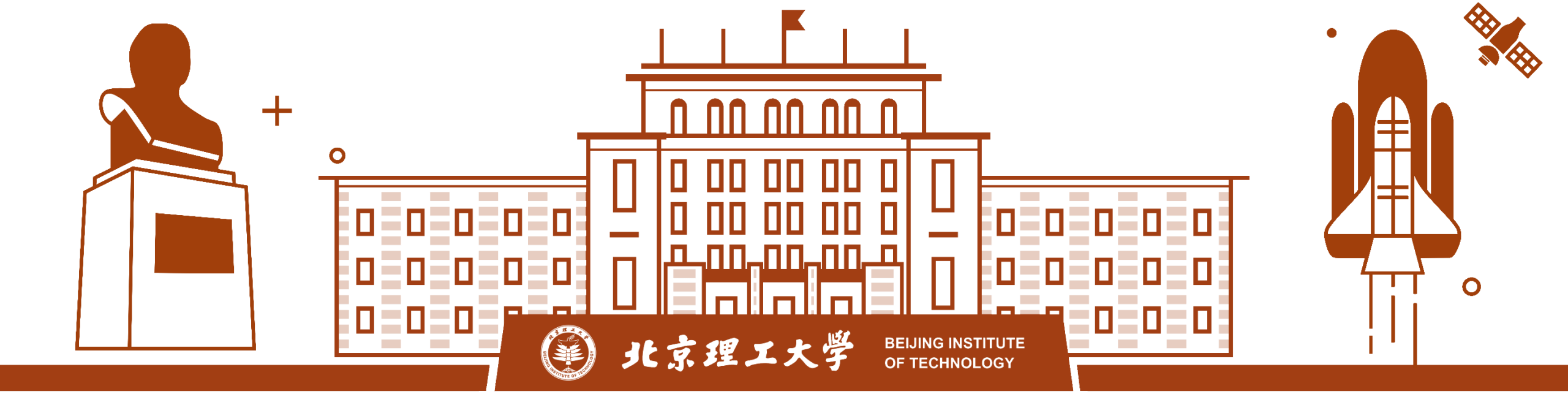


机器翻译

Machine Translation

小组成员：陈家祺，王宇杰，苏晓阔，冯雨莹，汪延诚，李劭彧

时　　间：2021.10.25

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

目录

INDEX

1 机器翻译经典综述

2 神经机器翻译

3 预训练语言模型实现机器翻译

4 机器翻译前沿进展

5 Demo展示

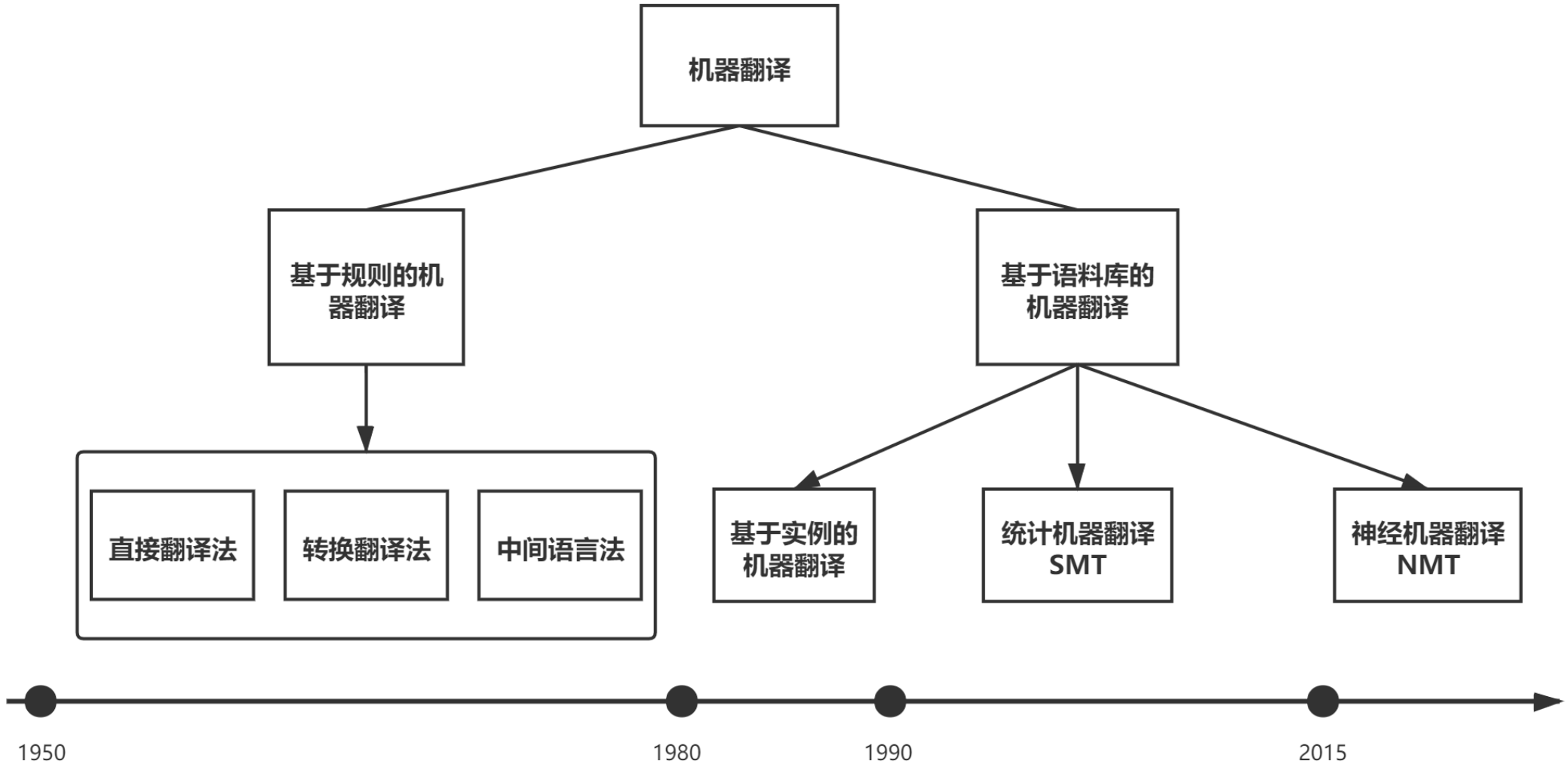
1 机器翻译经典综述

机器翻译定义

机器翻译（Machine Translation）又称自动翻译（Automated Translation），是利用计算机把一种自然源语言转变为另一种自然目标语言的过程，是自然语言处理的一个分支。

目前机器翻译发展迅速，应用场景广泛，在双语互译、语音同传、跨语言检索等方面得到了应用。

机器翻译方法及发展历史



基于规则的机器翻译

- **直接翻译法：**

将单词、短语、句子直接置换为目标语言的译文，不进行句法分析和语义分析。

How are you? → 怎么是你?
How old are you? → 怎么老是你?

- **转换翻译法：**

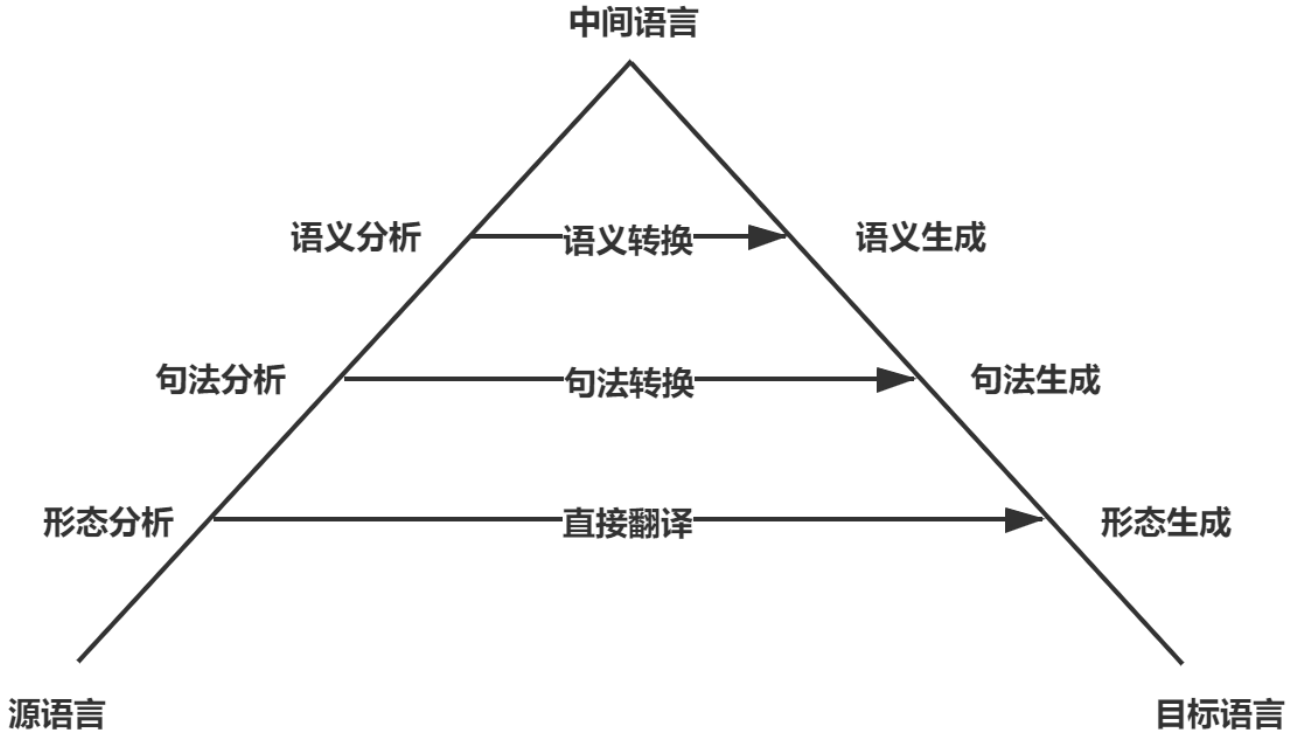
将源语言输入通过形态分析、句法分析甚至语义分析，转换生成目标语言输出。

- **中间语言法：**

将源文本转换成统一的与具体语言无关的中间语言，再由中间语言得到目标文本。

I want two apples.

我 想要 两个 苹果。



基于语料库的机器翻译

- **基于实例的机器翻译：**

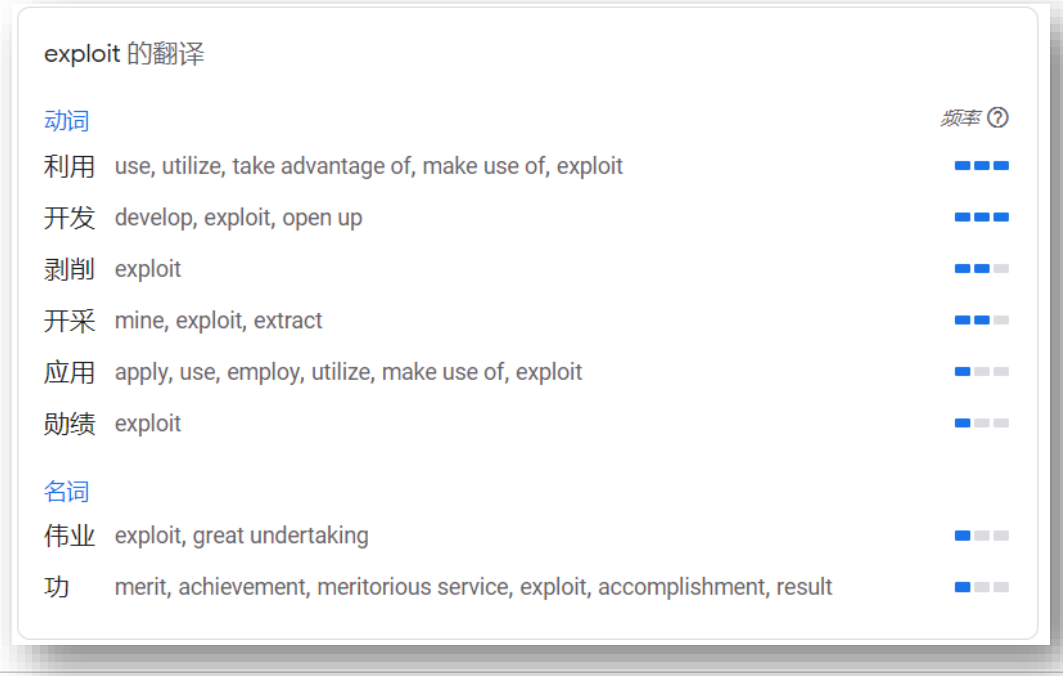
基于实例的机器翻译系统中包含一个双语对照的翻译实例库。每输入一个源语言句子时，系统把这个句子同实例库中的源语言句子进行比较，找出最为相似的句子，并模拟这个句子对应的译文，给出源文本的译文。

I’m going to the school. → 我要去学校。

I’m going to the park. → 我要去公园。

- **统计机器翻译：**

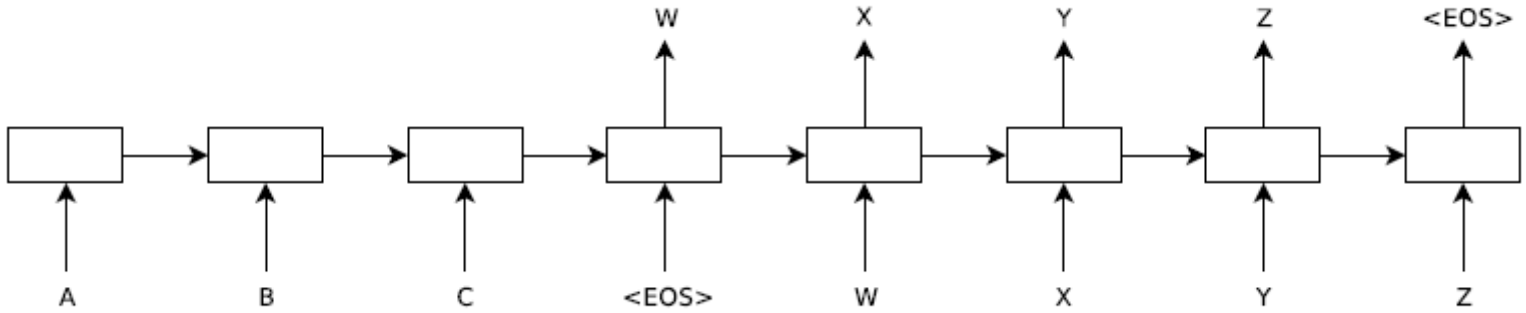
为翻译过程构建模型，从所有可能的译文中选择概率最大的译文，将翻译问题转变为搜索问题。包括基于词、短语、句法的统计机器翻译。



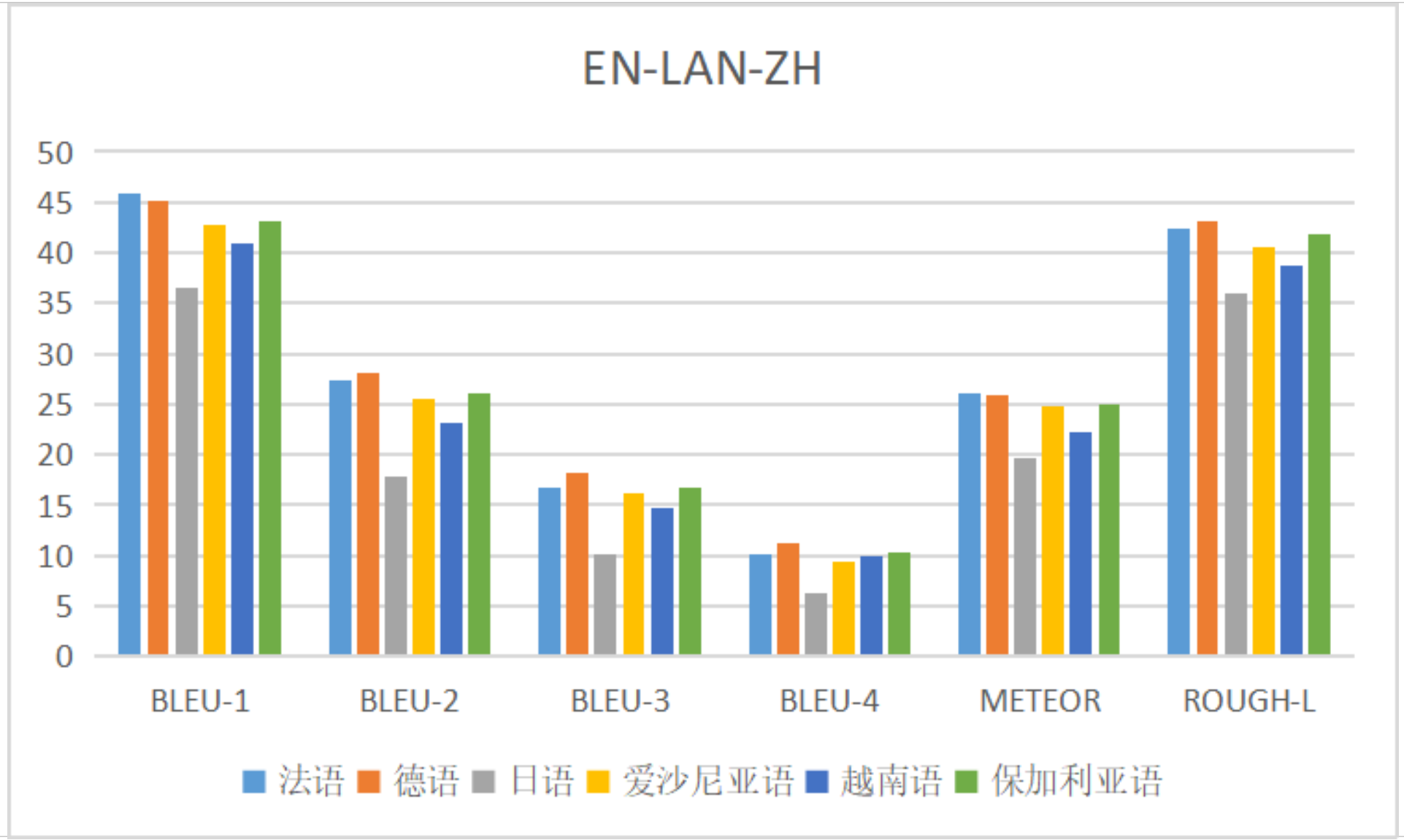
基于语料库的机器翻译

- **神经机器翻译（NMT，Neural Machine Translation）：**

神经机器翻译使用基于神经网络的技术实现源语言到目标语言的翻译过程。神经机器翻译系统是一个encoder-decoder系统，能够训练一个从一个序列映射到另一个序列的神经网络，实现精确的上下文翻译。



不同语种翻译情况


EN-LAN-ZH
50
45
40
35
30
25
20
15
10
5
0
BLEU-1
BLEU-2
BLEU-3
BLEU-4
METEOR
ROUGH-L
法语
德语
日语
爱沙尼亚语
越南语
保加利亚语

机器翻译问题

1. 翻译人名、习语等生词和低频词时效果不够理想。

e.g. No cross, no crown.

机器翻译：没有十字架，没有王冠。
人工翻译：不经历风雨，怎能见彩虹。

2. 词语存在歧义时，机器翻译难以辨别语境下词语的真实意义。

e.g. 十点了，我想起来了。

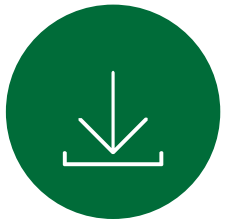
机器翻译：It's 10 o'clock, I remembered.
人工翻译：It's 10 o'clock, I want to wake up.

3. 在翻译长难句时容易出现错翻、漏翻等问题。

2 神经机器翻译

神经机器翻译

机器翻译的神经网络设计有很多形式，但都可被分为两大类：



递归模型

非递归模型

基于RNN的机器翻译模型

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

第一个成功的基于RNN的模型

由Sutskever等人于2014年提出，是一种纯深度RNN模型，并且性能非常接近当时统计机器翻译模型最好结果。

引入注意力机制

谷歌神经机器翻译(GNMT)是应用于谷歌翻译的一个工业级模型，并被视作是基于RNN神经机器翻译的一个里程碑。

与其他模型组合

Shazeer等人于2017年提出，将GNMT与多专家模型MoE组合，并取得了比原始GNMT模型性能更好的模型。

基于CNN的机器翻译模型

最早应用CNN的模型

由 Kalchbrenner 等 人 于 2013年提出，使用了一个CNN编码器和RNN解码器。

完全基于CNN的模型

由Kaiser等人于2016年提出， 它 应 用 了 Extended Neural GPU，并取得了与GRU+Attention 模 型 相 似 的性能。

早期最好的模型

由 Gehring 等 人 于 2016年提出，并取得了与当时基于RNN模型的相似性能。

ConvS2S & Transformer

ConvS2S

Convolutional Sequence to Sequence，由Gehring等人提出，是一种基于CNN与注意力机制相配合的机器翻译模型

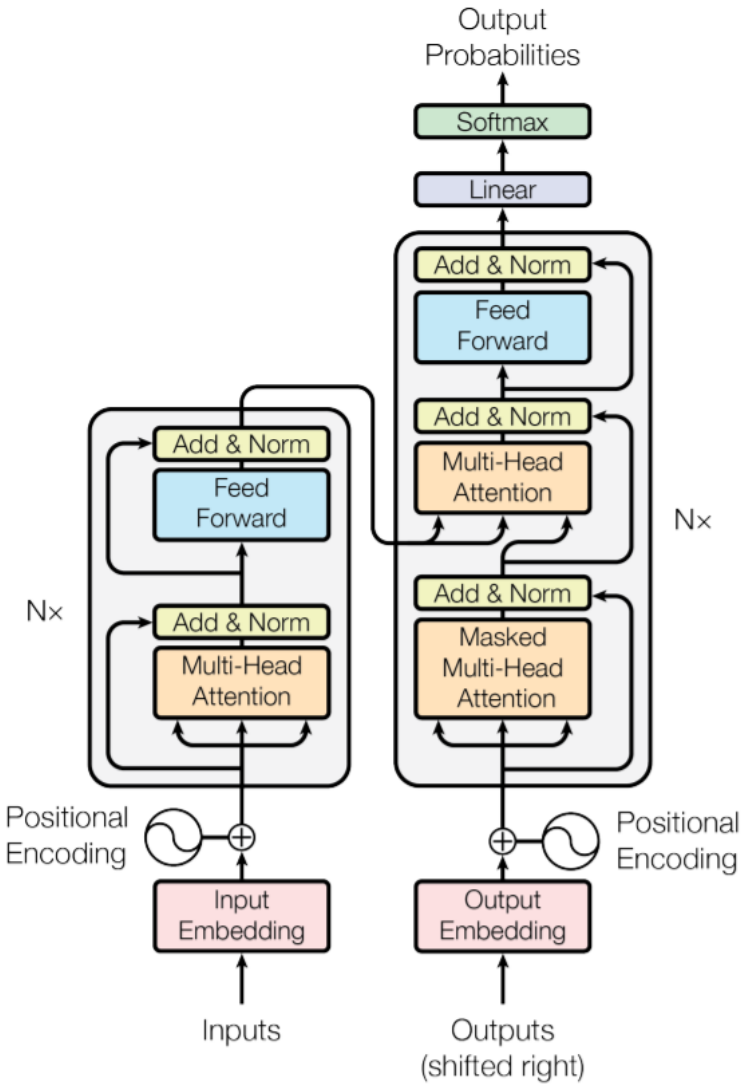
VS

Transformer

由Vaswani等人提出，它与传统的RNN/CNN模型不同，创新的使用了一种多层自注意力块与位置编码结合的方法

Transformer

Transformer



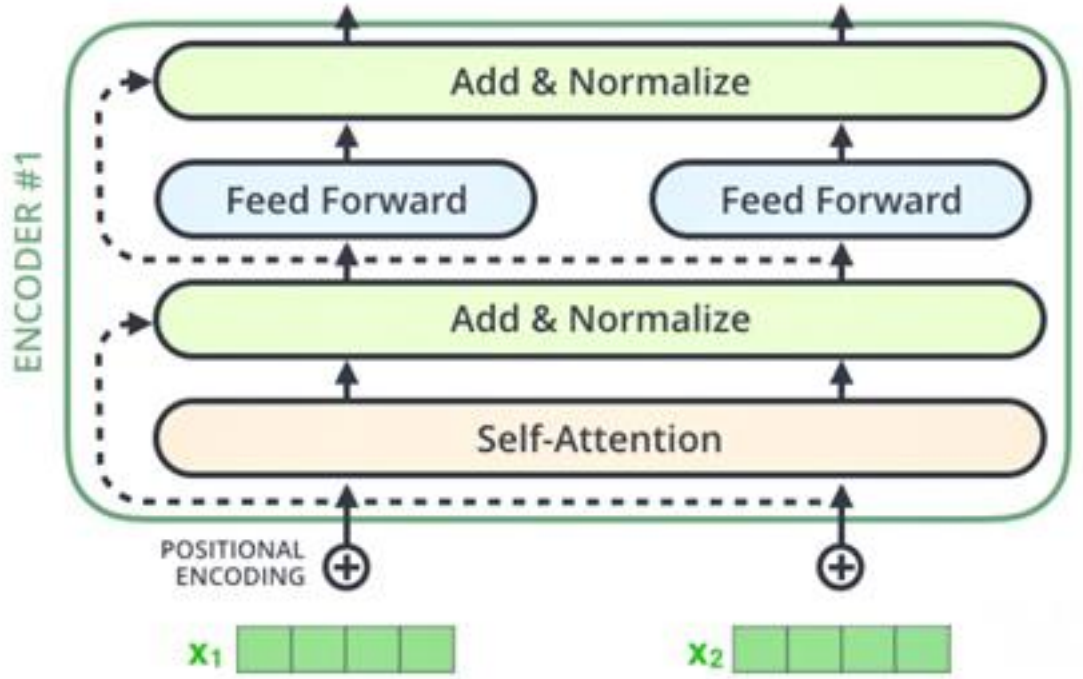
Transformer流程介绍——1.词向量化

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

首先利用词嵌入技术将句子中的每个词映射为向量，比如可以是一个长512的行向量，那么有n个单词的句子就是一个n×512的矩阵。

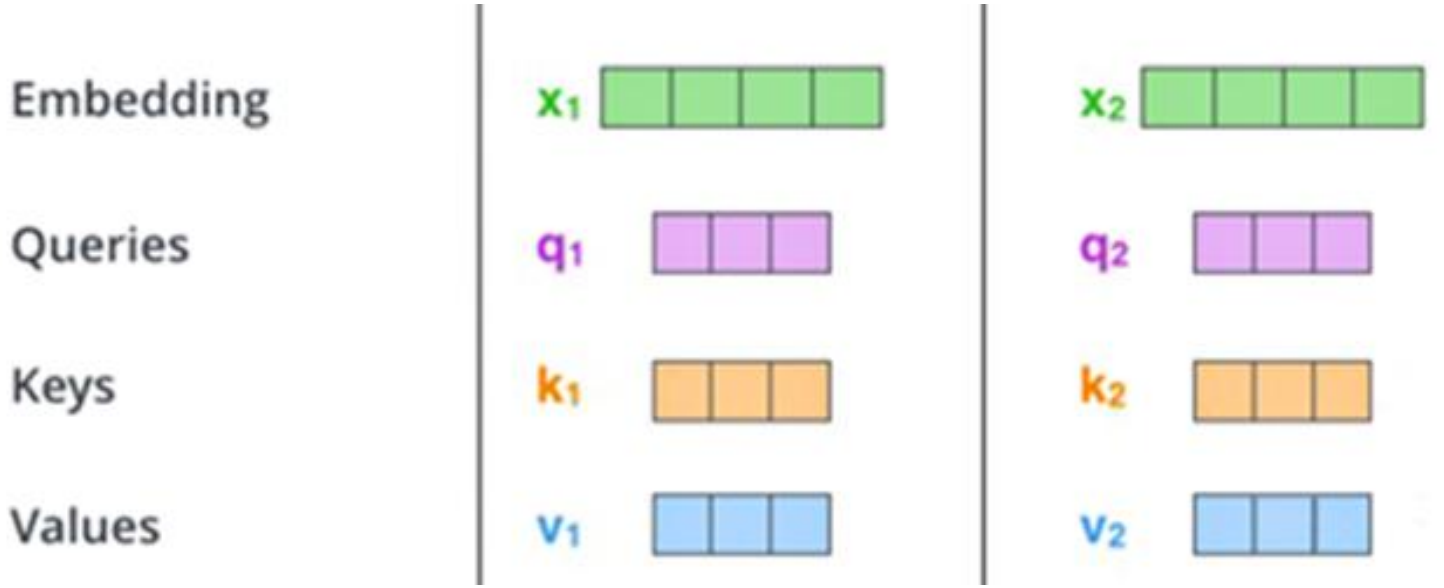
使用公式进行位置编码，其中pos代表单词的位置，i表示维度，dmodel与词向量维度相同。

Transformer流程介绍——2.编码流程



编码器中的第一个子层就是self-attention子层

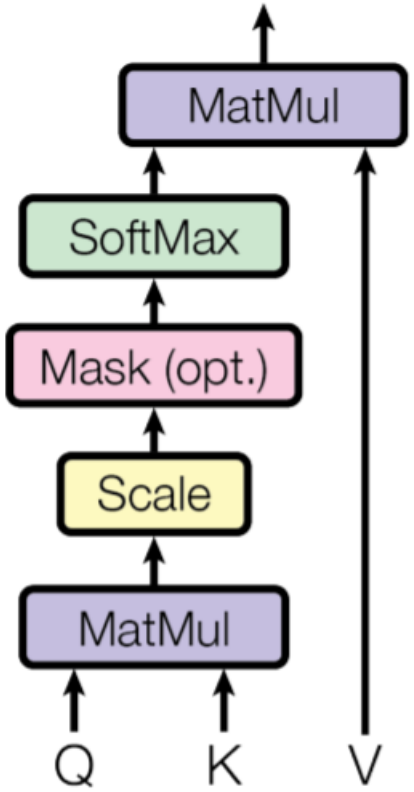
Transformer流程介绍——2.编码流程



以n=2为例，此时句子由两个单词构成，词向量分别是x1和x2。

self-attention子层先令词向量乘以三个训练得到的、维度相同的矩阵W^Q，W^K和W^V分别得到Query、Key和Value三个向量。

Transformer流程介绍——2.编码流程

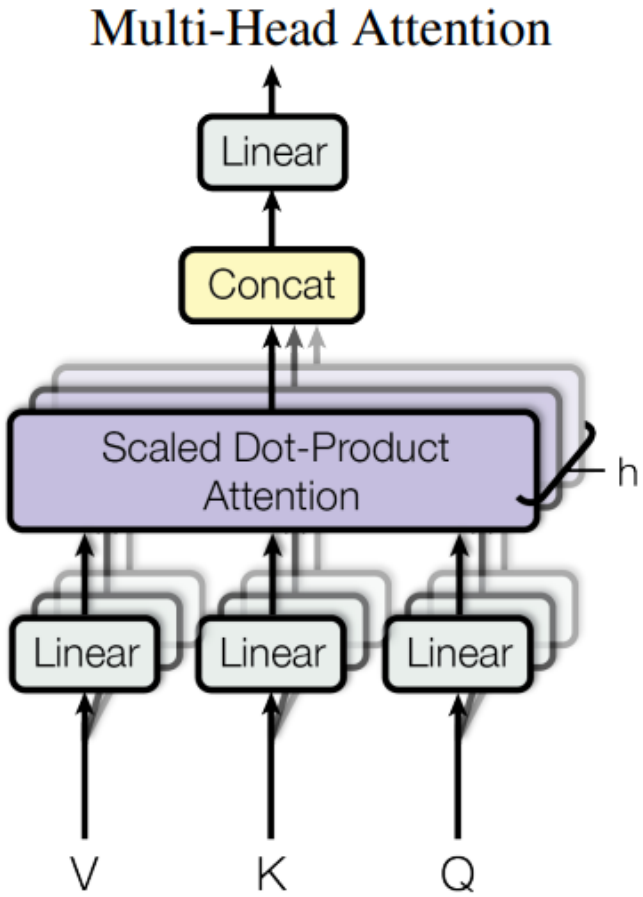


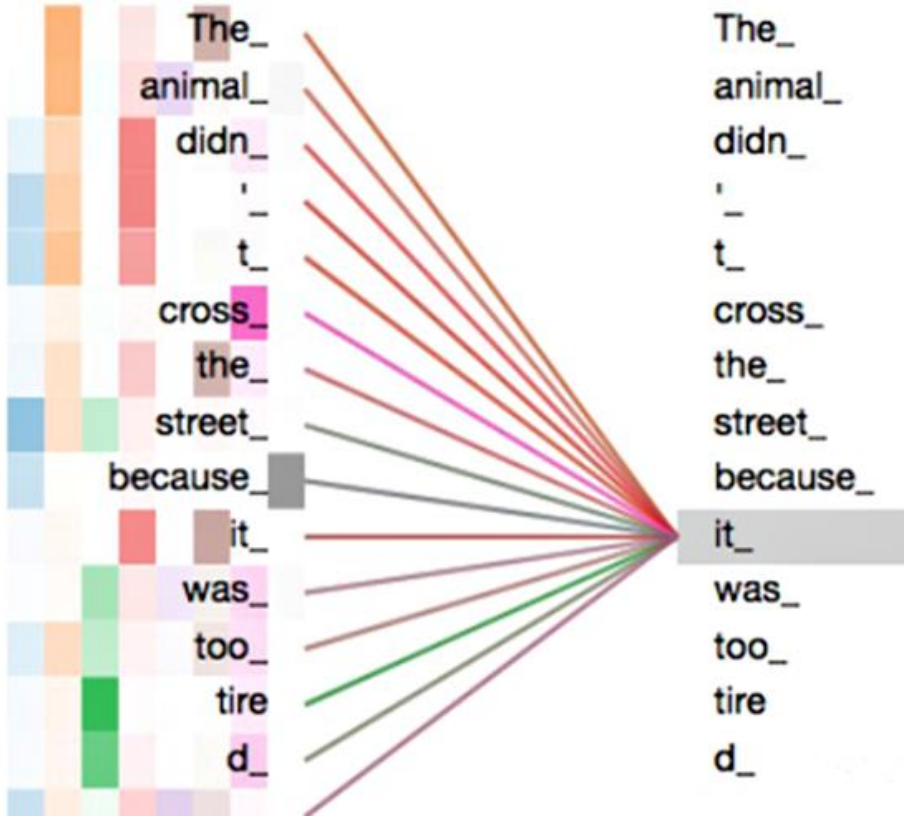
$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$MultiHead(Q,K,V) = Concat(head_1,\cdots,head_h)W^O$$
$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

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

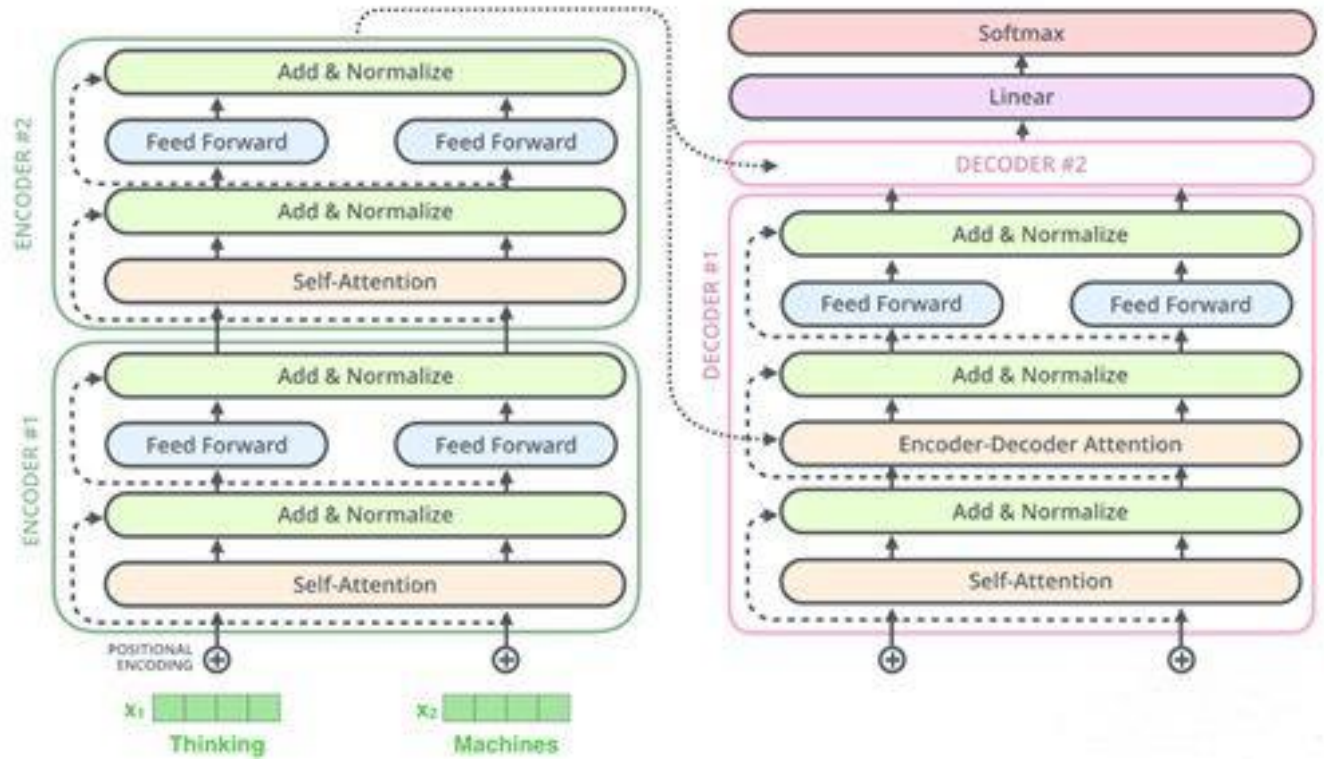
Transformer流程介绍——2.编码流程



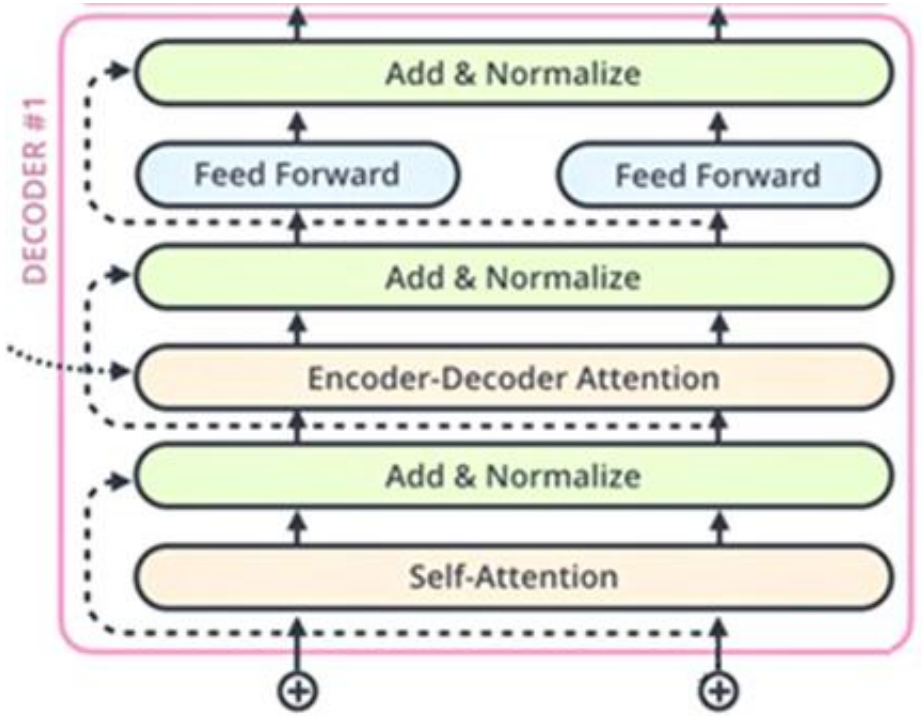


Multi-Head Attention

Transformer流程介绍——3.解码流程



Transformer流程介绍——3.解码流程



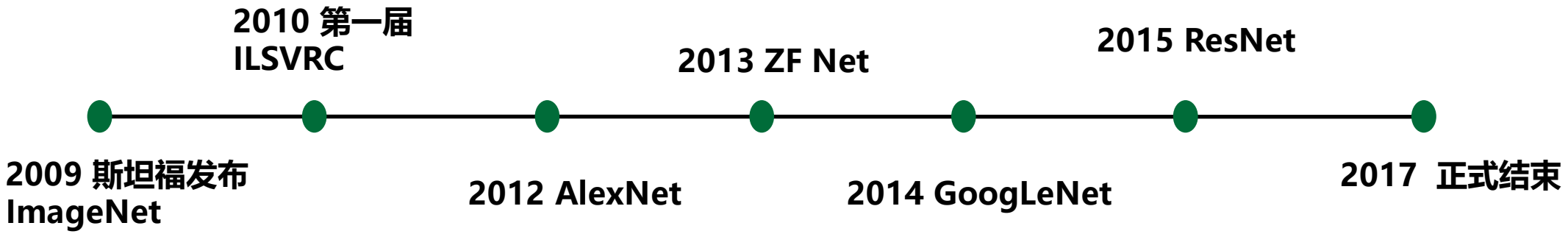
Transformer流程介绍——4.输出预测值

最后是输出预测结果，经过线性层对解码组件得到的向量进行投影成词典大小长度的向量，向量的每一维代表词典中的每个单词，对向量softmax后，向量中的值越大则代表对应单词的概率越大，最后输出概率最大的单词作为翻译结果即可。

3 预训练语言模型实现机器翻译

近年来，Pre-Training Language Models (PLMs)范式正在蓬勃发展。思想是首先在大规模语料库中预训练模型，然后在各种下游任务中对这些模型进行微调，以获得最先进的结果。

ImageNet



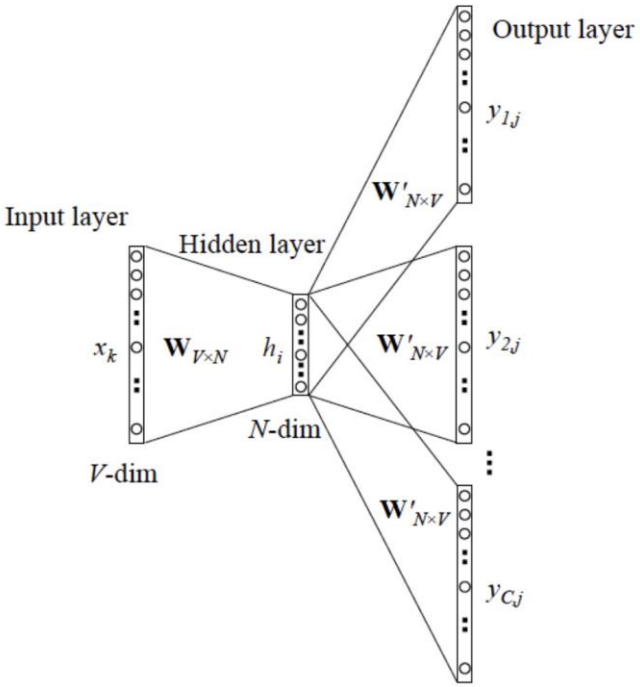
ImageNet的贡献:

（1）ImageNet是深度学习热潮的关键推动者；

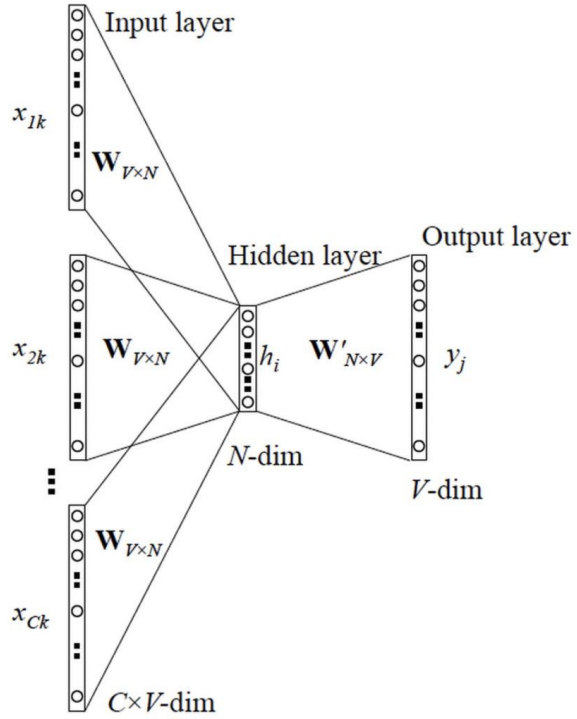
（2）ImageNet的成功证明了，在深度学习时代，数据和模型同等重要；

（3）ImageNet在迁移学习中同样取得重要性突破，开启了“预训练+微调”的范式。

Word2Vec(2013)



Skip-gram



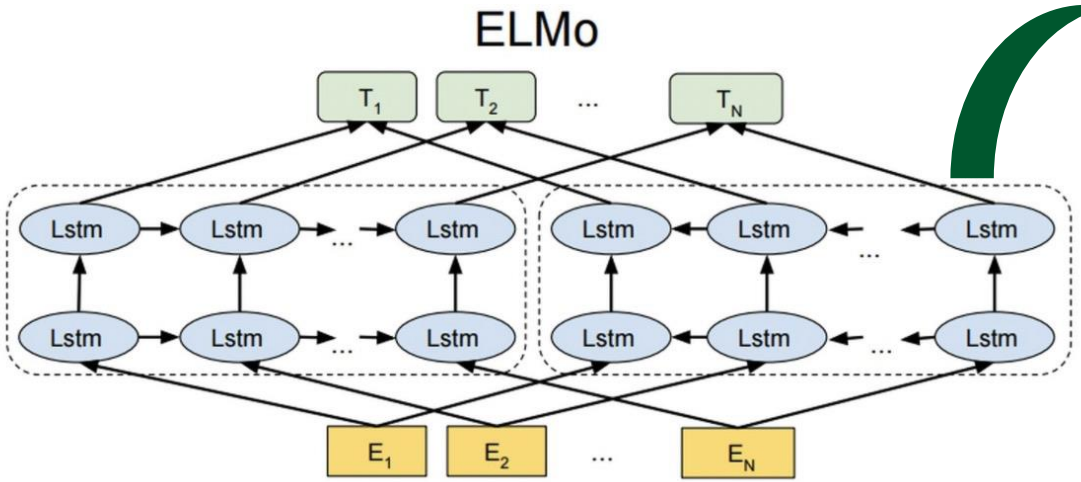
CBOW

翻译质量的评估：

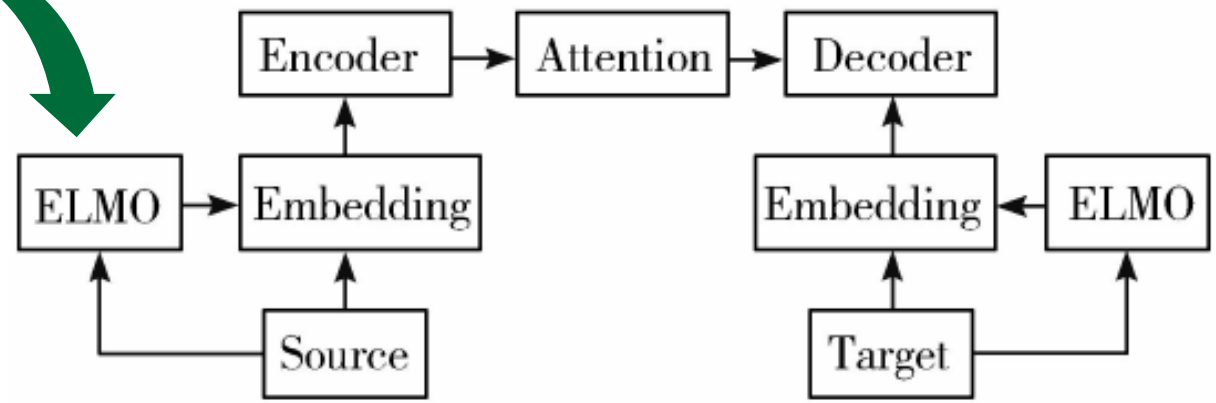
（1）BLEU（Bilingual Evaluation Understudy）：基于准确率

（2）ROUGE（Recall-Oriented Understudy for Gisting Evaluation）：基于召回率

ELMo(2018)



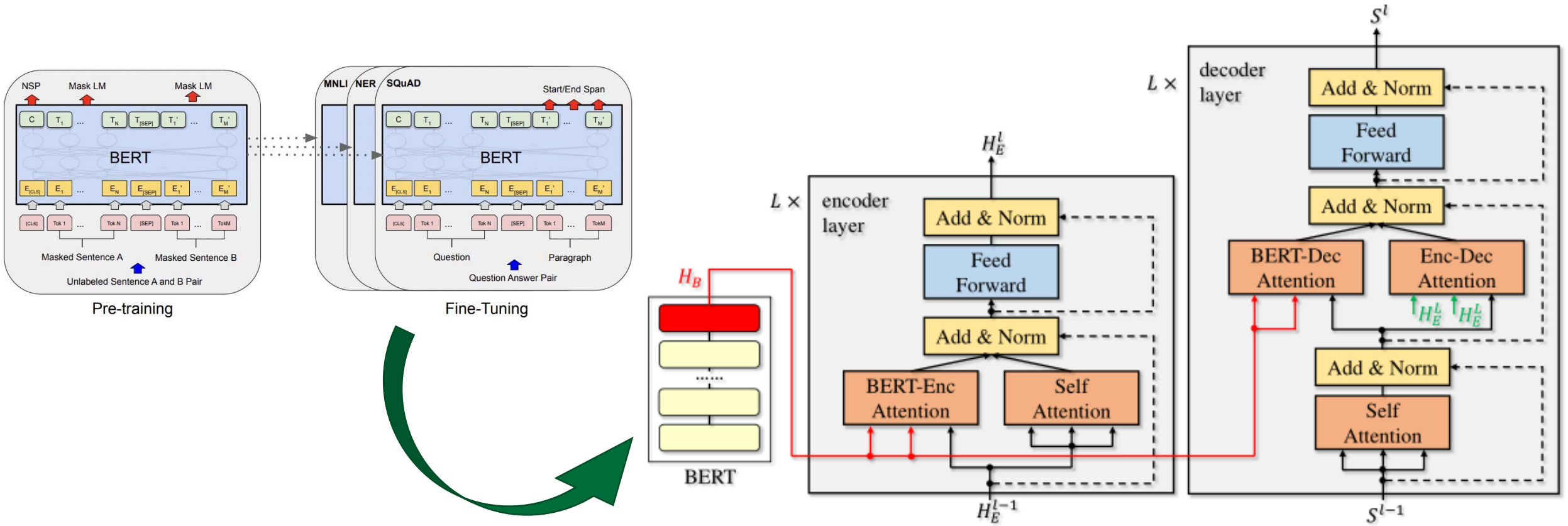
ELMo模型结构示意图



基于ELMo的seq2seq模型

Source:《 ELMO-based low-resource neural machine translation 》

BERT系列模型：BERT(2019)



BERT-FUSED MODEL

Source:《 INCORPORATING BERT INTO NEURAL MACHINE TRANSLATION 》

4 机器翻译前沿进展

大数据时代与低数据可用性的矛盾

- 翻译行业不断发展以满足客户的需求，神经机器翻译（NMT）是这一进程的最新一步。由于能够一次翻译整个句子，NMT的输出可以类似于人工翻译。该系统使用了数千万平行句子作为训练数据。如此巨量的训练数据仅仅在少数语言对可以获得，也仅限于少数特定领域，例如新闻领域或官方记录。

- 事实上，尽管全球共有大约七千种口语，但是绝大多数语言都不具备训练可用机器翻译系统所需的大量资源，这就体现了“低数据可用性”。当前自然语言处理的发展为低资源语言和领域提供了挑战和机遇。

训练数据稀少时可供训练的方法

数据增强

利用已有平行句对自动生成新的平行句对，通过扩展语料库的规模和丰富训练数据的多样性。

远程监督

远程监督又称为弱监督法，通过外部信息源自动或半自动地为无标签数据打上标签。

跨语言映射

在平行语料库中，将高资源语言和低资源语言进行对齐，将高资源语言中的标签映射到低资源语言中。

噪声处理

噪声滤波：训练过滤器，通过概率阈值进行过滤。
噪声建模：构建混淆矩阵估计干净数据和噪音数据之间的关系。

小样本学习

所谓小样本学习（Few-Shot Learning），就是使用远小于深度学习所需要的数据样本，达到接近甚至超越大数据深度学习的效果，也即是小样本学习的本质就是学习的效果与数据比较的提升，或者说单位数据产生的模型收益增大了。

小样本学习

如果先验知识是充足的，那么其实可以做到“数据不足，知识来凑”。

另一种方法则是提高单位数据的使用效率，如果每个数据对模型的改进都是有效的，远离随机游走。

方法1

让给定模型具备相关任务的先验知识

方法2

让每个数据产生的学习效果进一步到位

方法1 让给定模型具备相关任务的先验知识

通过基于预训练的范式来构建先验，经过一类预训练的模型通常在后续任务里样本需求大大下降。

基于元学习的范式。指通过学习一系列任务集来掌握一个基本模型，在新的任务来到时候，用最小的数据完成训练过程。

使用模块化的系统，可以通过模块的复用或组合迅速的实现小样本学习。

增加记忆系统。深度强化学习是一个需要特别大数据样本的学习范式。

方法2 让每个数据产生的学习效果进一步到位

改善优化方法

有很多学习方法可以更充分的利用每个数据，比如recursive least square或者SVM。
对于小数据学习的致命杀手是灾难性遗忘，比如利用正交权重修改算法和情境依赖处理模块，可以使灾难遗忘问题得到有效缓解，从而提高了数据的利用效率。

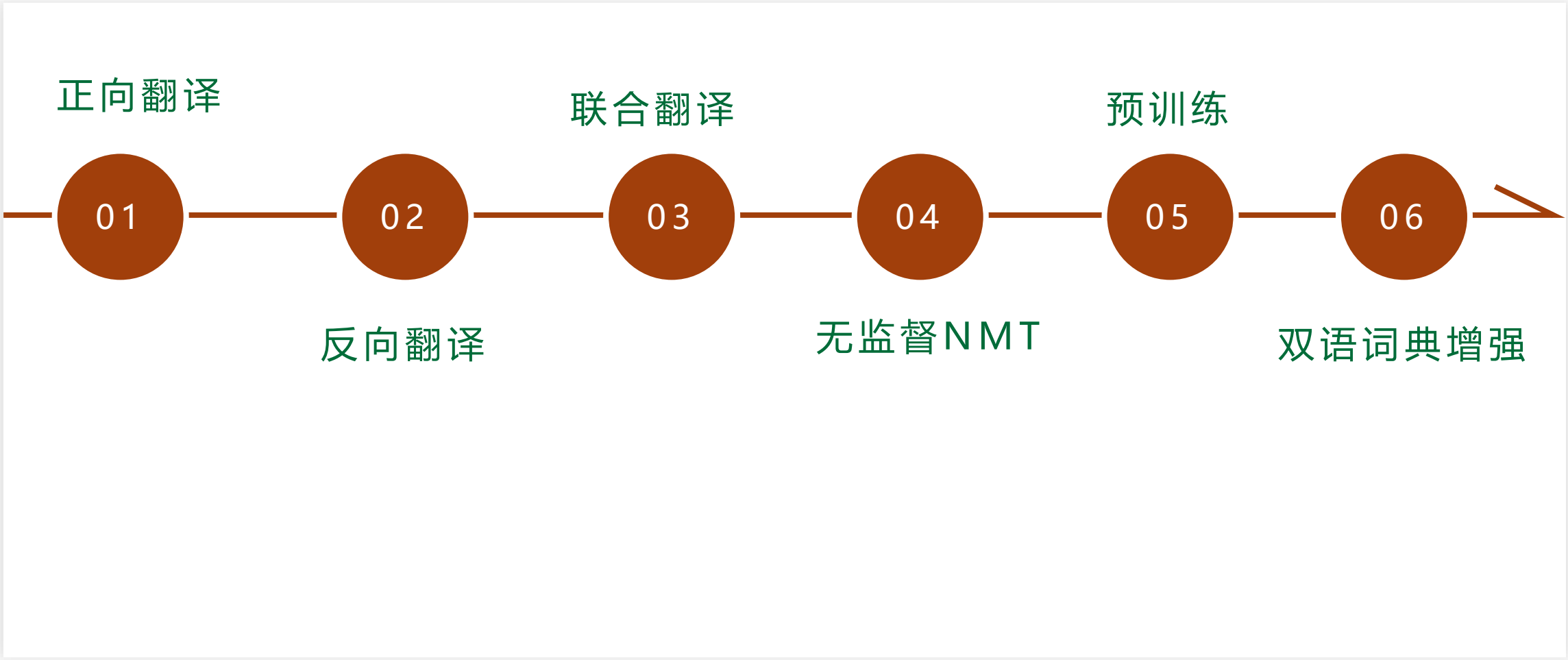
减少模型的参数量

这一类非常典型的代表就是用一个随机的网络直接提取特征， 而不去训练内在的权重， 然后只训练一个读出层得到需要的输出。由于读出层的参数数量往往比整个网络小很多， 从而减少需要的模型参数数量。

未来机器翻译的发展方向

- 利用未标记的数据来提升机器学习模型性能表现在各个领域都是一种流行且有效的方法。同样在NMT中，收集并利用未标记的单语数据比并行数据更容易，成本更低。
- 利用辅助语言（auxiliary languages）的数据。语言具有类似语法或语义是互相帮助的时候训练NMT模型。利用相关语言或富资源(rich resource)语言数据已经显示出巨大的成功。
- 利用多模态（语音，图像，视频）的数据帮助低资源（low resource）NMT模型的训练。

未来机器翻译的发展方向-单语数据



未来机器翻译的发展方向-单语数据

正反向翻译

在反向翻译中，伪平行句对是通过反向翻译系统将目的语言单语句翻译到源语言而产生的。而在正向翻译中，将源语言端单语句子通过翻译系统向目标语言进行相同方向的翻译，生成伪平行句对。然后，将伪并行数据与原始并行数据混合训练NMT模型。已有研究表明，在NMT系统上，前后平移提供了很有前景的性能增益。

未来机器翻译的发展方向-单语数据

无监督NMT

为了处理没有任何平行句子的零资源场景，采用无监督的NMT学习方法。无监督的NMT学习一般很难保证学习的质量和效率。所以一般在无监督学习中依赖两个组件:

1）双语对齐，使得模型语句之间有良好的对齐;

2）翻译质量改进，通过多次迭代学习来提高翻译的质量。

未来机器翻译的发展方向-辅助语言

基于枢纽语言

01

多语言模型

02

迁移学习

03

多语言模型

多语言训练的主要思路是让低资源语言和其他的语言进行联合训练。

1.选取辅助语言时,一般使用富资源语言作为辅助语言帮助低资源语言数据的生成。

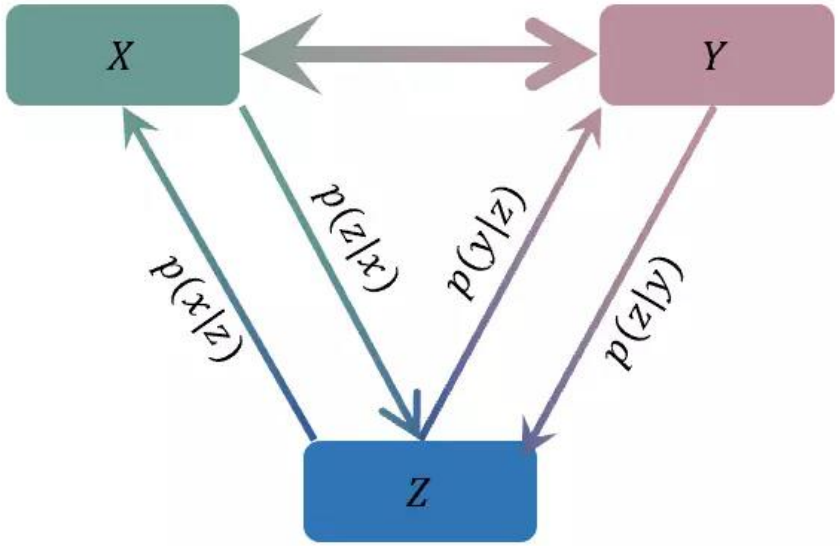
2.考虑有限模型由于不同语言的训练容量和训练数据大小不同，该模型可能偏向于资源丰富的语言。因此，平衡数据在多语言NMT中，数据量大小对于低资源语言很重要。目前提出的是一种自动加权训练数据大小的方法。

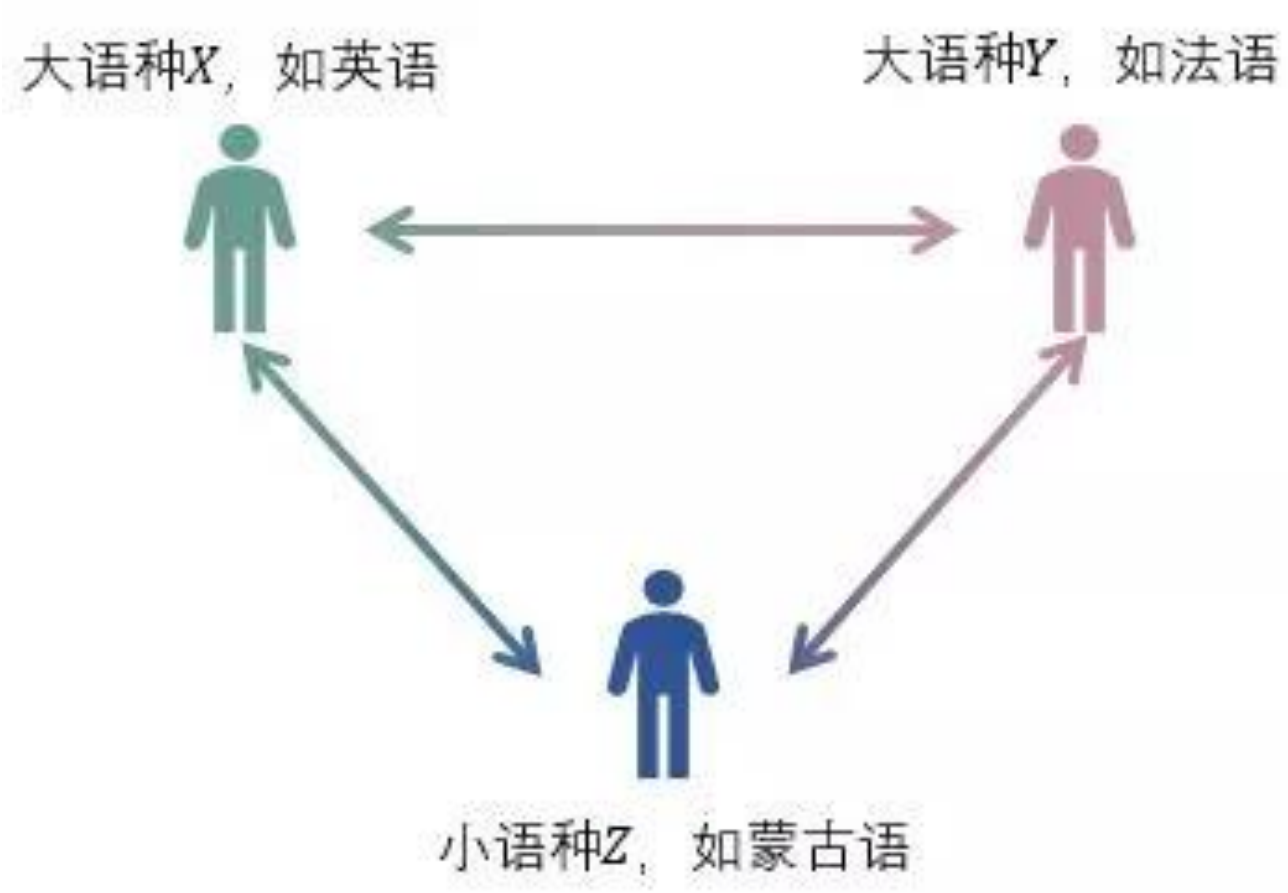
3.多语言训练的提出给zero-shot translation带来了可能，两种语言之间即使没有可供使用的平行语料库也可以完成翻译。通常的做法是：训练一个多语言模型，然后back translation产生伪平行语料库并利用其进行模型微调。

小语种翻译实例

TA-NMT

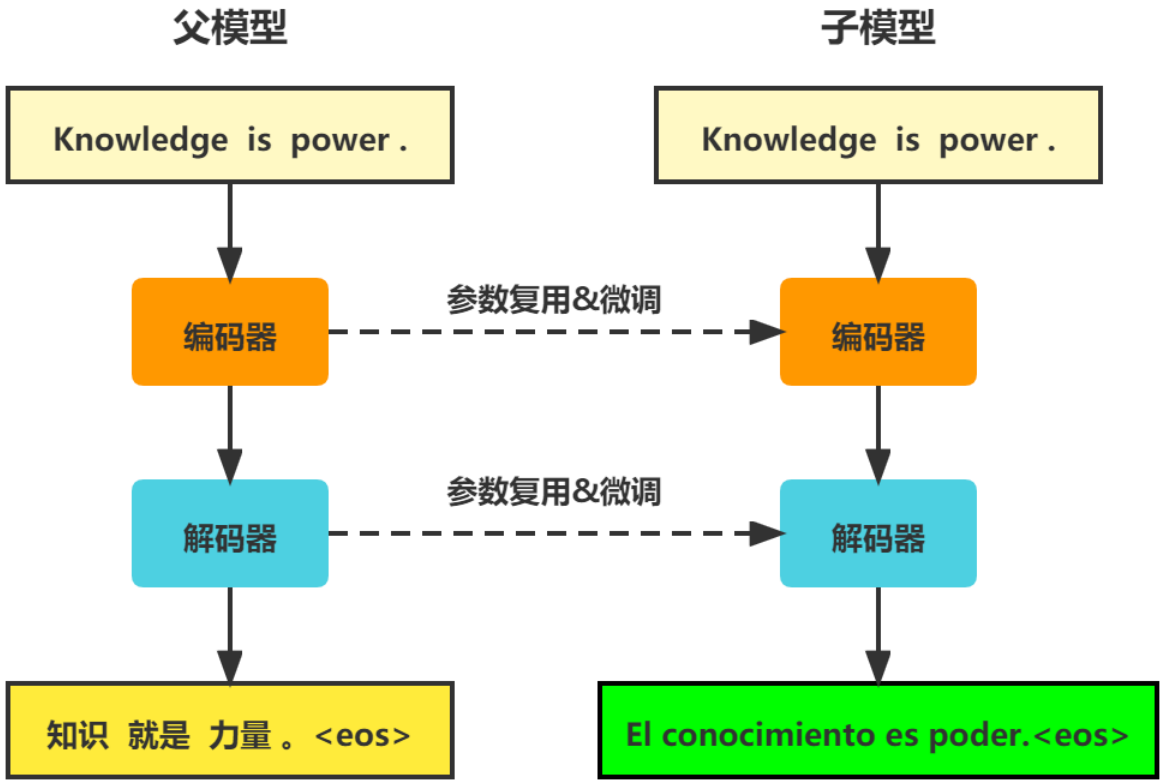
三角结构神经机器翻译模型：
充分利用大语种丰富的对齐语料
来提升小语种机器翻译的能力。





迁移学习

在机器翻译中，可以用富资源语言的知识来改进低资源语言上的机器翻译性能，也就是将富资源语言中的知识迁移到低资源语言中。迁移学习将所有任务分类为源任务和目标任务，目标是将源任务中的知识迁移到目标任务当中。

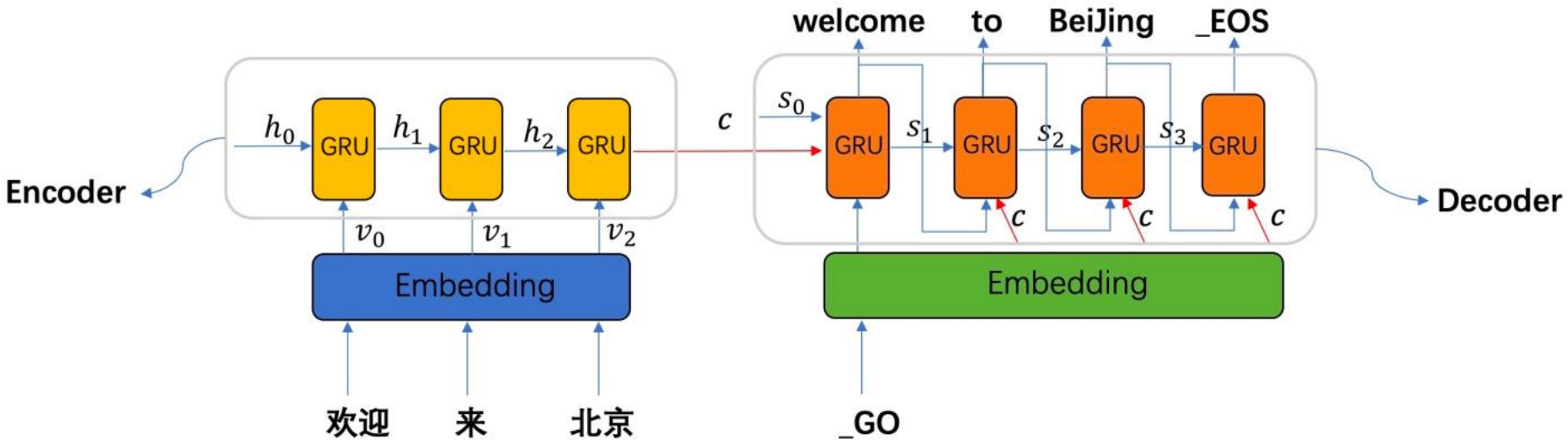


未来机器翻译发展的问题

- 在多语言和迁移学习中，使用多少辅助语言和哪些辅助语言尚不清楚。训练模型选择一种辅助语言。直观来看，使用多种辅助语言可能优于只使用一种，值得探索。
- 训练包含多种低资源语言的多语言模型成本很高。将多语言模型转换为看不见的低资源语言是一种有效的方法，挑战在于如何处理看不见的语言的新词汇。
- 如何有效地选择中枢语言很重要，但还没有得到很好的研究。
- 就多模态而言，虽然语音数据有潜力提升NMT，但这样的研究是十分有限的。例如，有些语言在语音上相近，但在文字上不同(如塔吉克语和波斯语)。
- 当前的方法已经对低资源语言做出了显著的改进，这些语言要么具有足够的单语数据，要么与一些资源丰富的语言相关。不幸的是，一些低资源语言的单语数据非常有限，并且远离资源丰富的语言。如何处理这类语言具有挑战性，值得进一步研究。

5 Demo展示

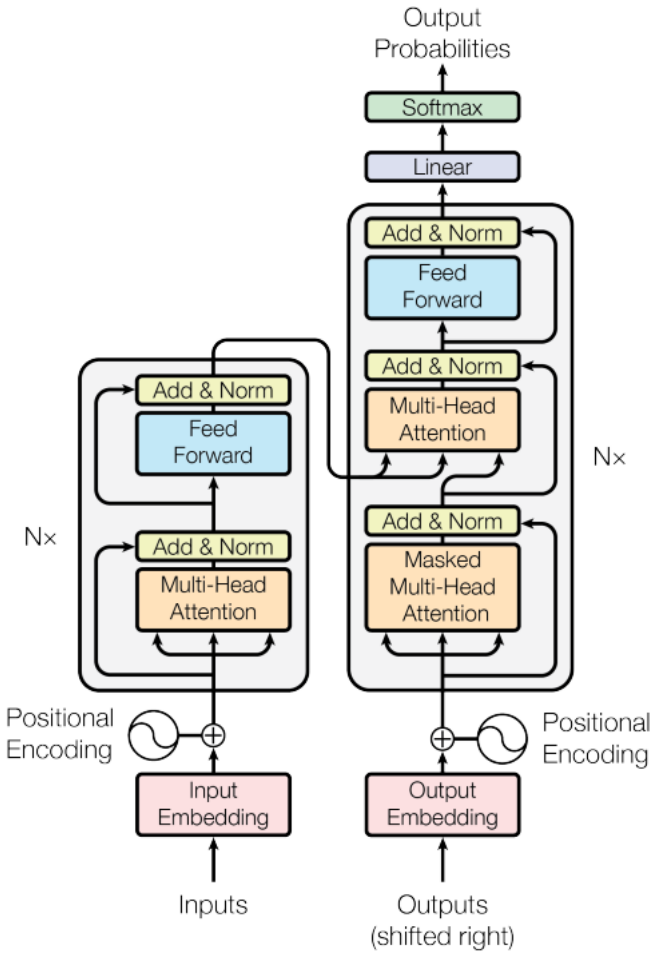
Seq2Seq翻译模型

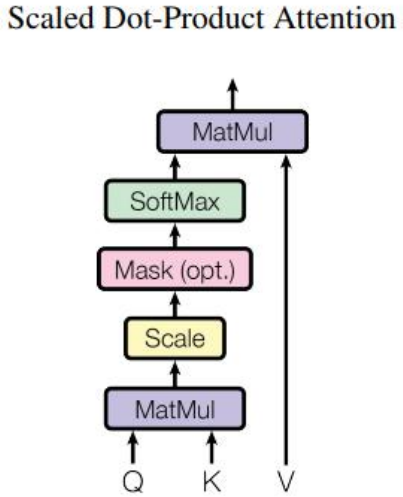


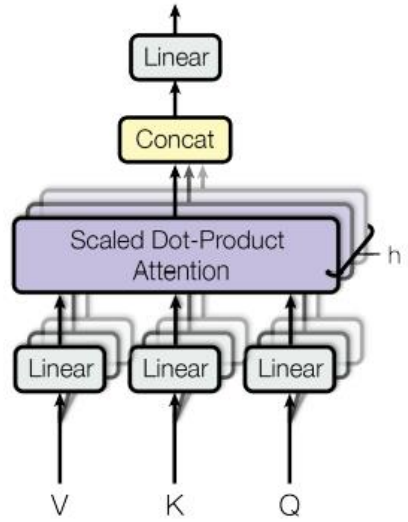
Seq2Seq翻译模型


```python
class Encoder(nn.Module):
    def __init__(self, hidden_size, embedding, n_layers=1, dropout=0):
        super(Encoder, self).__init__()
        self.hidden_size = hidden_size
        self.embedding = embedding
        self.n_layers = n_layers

        # 如果只有一层就不dropout，多层则使用初始化参数
        # input_size和hidden_size相同，这里是假设embedding输出尺寸也为hidden_size
        # Tensorflow中多层堆叠rnncell需要使用MultiRNNCell()，而PyTorch中直接指定层数，两者rnncell均可单独调用（如GRUCell()）
        self.gru = nn.GRU(hidden_size, hidden_size, int(n_layers),
                          dropout=(0 if n_layers==1 else dropout), bidirectional=True)
class Decoder(nn.Module):
    def __init__(self, embedding, hidden_size, output_size, n_layers=1, dropout=0.1):
        super(Decoder, self).__init__()

        self.hidden_size = hidden_size
        self.output_size = output_size
        self.n_layers = n_layers
        self.dropout = dropout

        self.embedding = embedding
        self.embedding_dropout = nn.Dropout(dropout)
        self.linear = nn.Linear(hidden_size+1, hidden_size)
        self.gru = nn.GRU(hidden_size, hidden_size, int(n_layers), dropout=(0 if n_layers==1 else dropout), bidirectional=False)
        self.concat = nn.Linear(hidden_size * 2, hidden_size)
        self.out = nn.Linear(hidden_size, output_size)

        self.attn = Attention(hidden_size)
```

Transformer翻译模型







Transformer翻译模型


```python
def make_model(src_vocab, tgt_vocab, N=6, d_model=512, d_ff=2048, h=8, dropout=0.1):
    c = copy.deepcopy
    # 实例化Attention对象
    attn = MultiHeadedAttention(h, d_model).to(DEVICE)
    # 实例化FeedForward对象
    ff = PositionwiseFeedForward(d_model, d_ff, dropout).to(DEVICE)
    # 实例化PositionalEncoding对象
    position = PositionalEncoding(d_model, dropout).to(DEVICE)
    # 实例化Transformer模型对象
    model = Transformer(
        Encoder(EncoderLayer(d_model, c(attn), c(ff), dropout).to(DEVICE), N).to(DEVICE),
        Decoder(DecoderLayer(d_model, c(attn), c(attn), c(ff), dropout).to(DEVICE), N).to(DEVICE),
        nn.Sequential(Embeddings(d_model, src_vocab).to(DEVICE), c(position)),
        nn.Sequential(Embeddings(d_model, tgt_vocab).to(DEVICE), c(position)),
        Generator(d_model, tgt_vocab)).to(DEVICE)
```

预训练翻译模型：Opus-MT

MARIANNMT

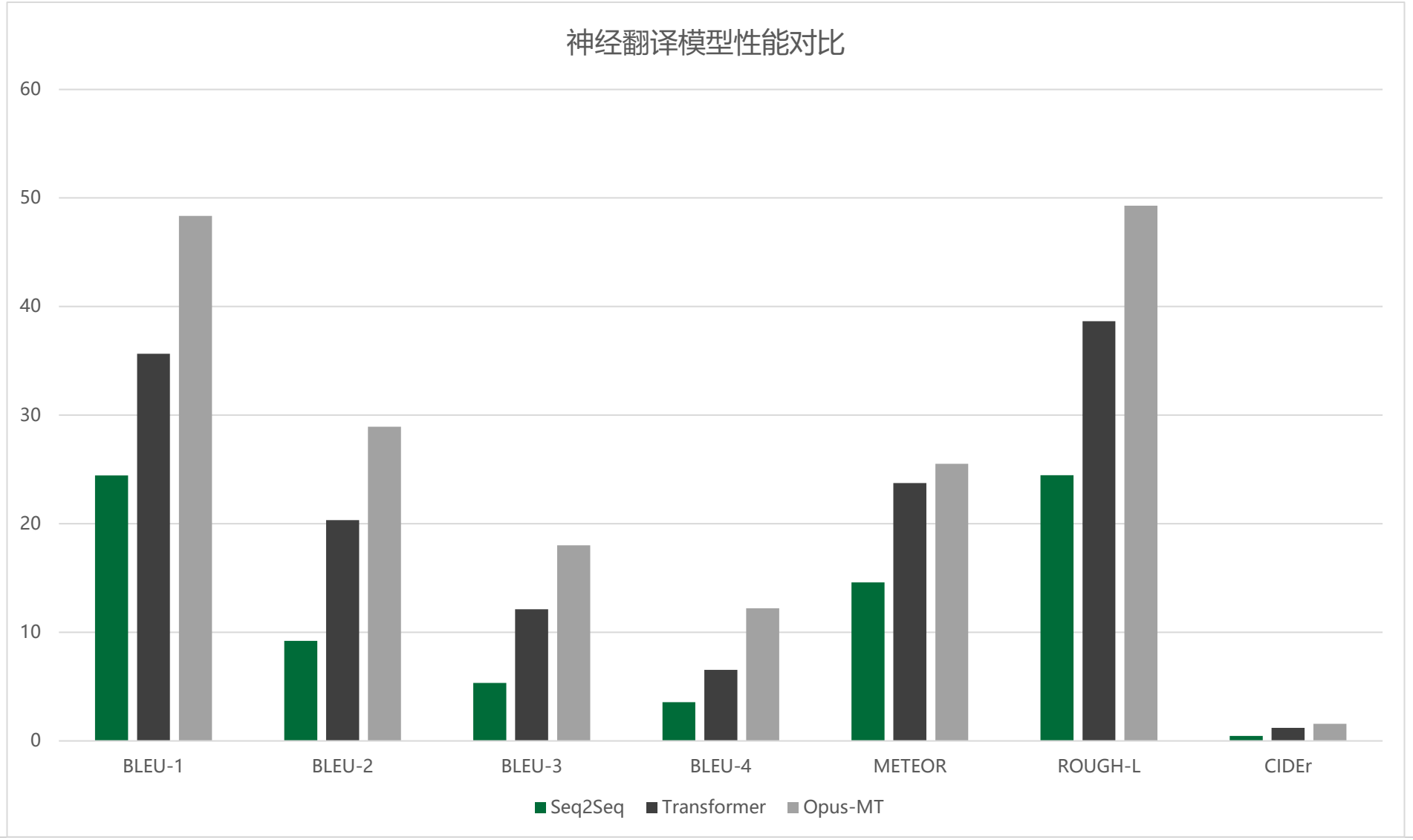
Fast Neural Machine Translation in C++

- 微软翻译引擎的核心
- C++编写，更加高效
- 支持多种模型（demo选用transformer）
- 开源

OPUS-MT

赫尔辛基大学在OPUS语料上重新训练

性能对比


神经翻译模型性能对比
60
50
40
30
20
10
0
BLEU-1
BLEU-2
BLEU-3
BLEU-4
METEOR
ROUGH-L
CIDEr
Seq2Seq
Transformer
Opus-MT

实例展示

Input	Seq2Seq	Transformer	PTM	Target
Another rule? That’ s rule number one.	另一个规则?是的规则。	还有一个规则?那是第一条规则。	另一条规则?这是第一条规则。	又是一条规则?这是规则一。
I have suspected him for ages.	我早就怀疑他了了。	我已经怀疑他已成长。	我怀疑他已经很久了。	我早就怀疑他了。
I call on you in the name of liberty.	在叫我名义名义。	我以自由的名义要求你。	我以自由的名义呼吁你。	以自由的名义我呼吁你们。

- 实时在线翻译
- API：免费使用
- AutoML：领域适应
- Media：实时音频、视频

- 实时在线翻译
- API：每月免费额度
- 离线翻译
- 开源（代码、预训练模型）

DeepL

- 实时在线翻译
- API：Free/Pro
- Pro：文档、术语表
- 本地工具集成

华为

华为云

面向商用

API：按需计费/套餐

音频/视频翻译生成

腾讯

- 实时在线翻译
- API：个人免费，商用收费
- OCR翻译
- 同传服务

百度

- 实时在线翻译
- API：个人免费，商用收费
- 垂直领域翻译
- 文档、图片、语音翻译

德以明理　　学以精工

谢谢