

预训练模型

Pretraining Models

王博 天津大学智能与计算学部 2019.12



- **Word2Vector**: Distributed Representations of Words and Phrases and their Compositionality. 2013 Google
- **Elmo**: Deep contextualized word representations. 2017. UW
- **ULMFiT**: Universal Language Model Fine-tuning for Text Classification, 2018 University of San Francisco
- **Transformer**: Attention Is All You Need, 2017 Google
- **GPT**: Improving Language Understanding by Generative Pre-Training, 2018 OpenAI
- **Bert**: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018.11 Google
- **ERNIE**: ERNIE: Enhanced Language Representation with Informative Entities, 2019 Baidu
- **Xlnet**: Generalized Autoregressive Pretraining for Language Understanding, 2019 CMU Google



- 模型选择
- 参数设定
- 领域知识
- ...

做任何一个任务，我们拥有的正确的、有信息量的先验知识越多，学习模型需要做的事情就越少。

对于一个领域，如果我们拥有一些适用于各种任务的、普遍的先验知识，那将是非常有价值的。



CV:

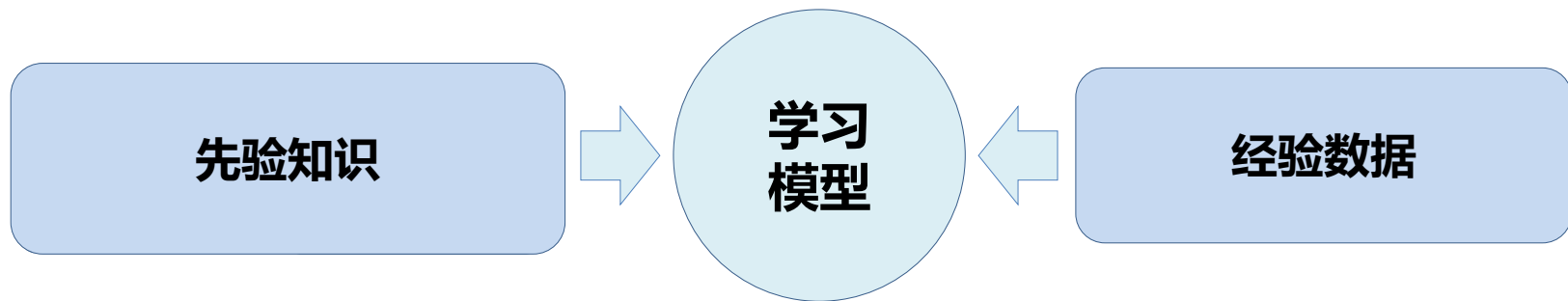
- 特殊形状
- 颜色范围
- 像素构成
- ...

NLP:

- 词义
- 语法
- 表达习惯
- ...

做任何一个任务，我们拥有的正确的、有信息量的先验知识越多，学习模型需要做的事情就越少。

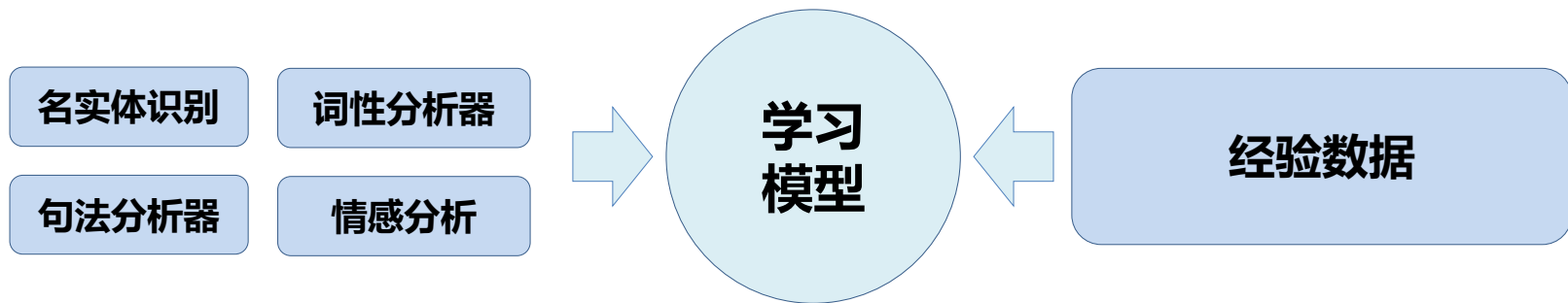
对于一个领域，如果我们拥有一些适用于各种任务的、普遍的先验知识，那将是非常有价值的。



NLP:

- 词义
- 语法
- 表达习惯
- ...

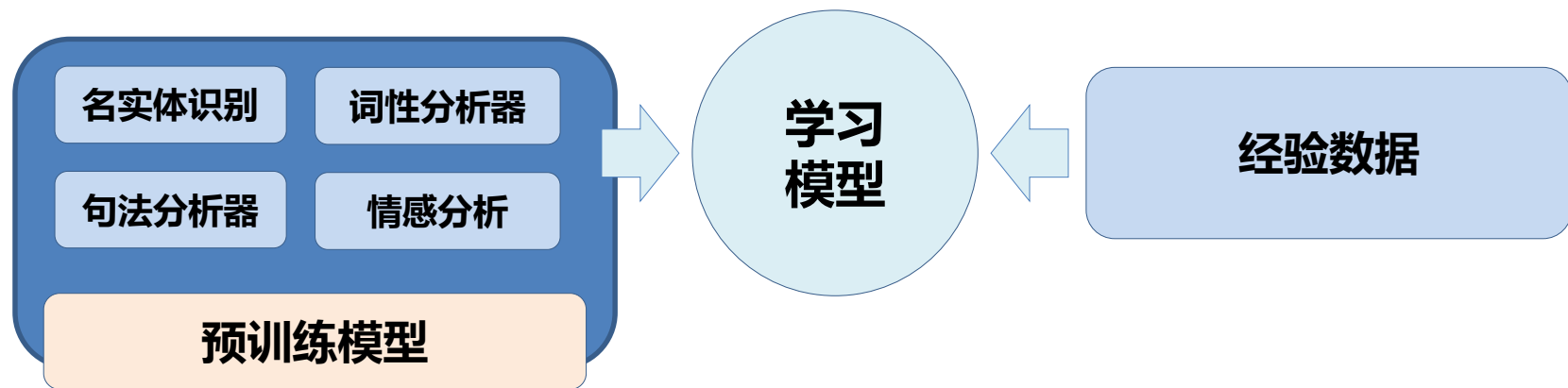
何为预训练模型?



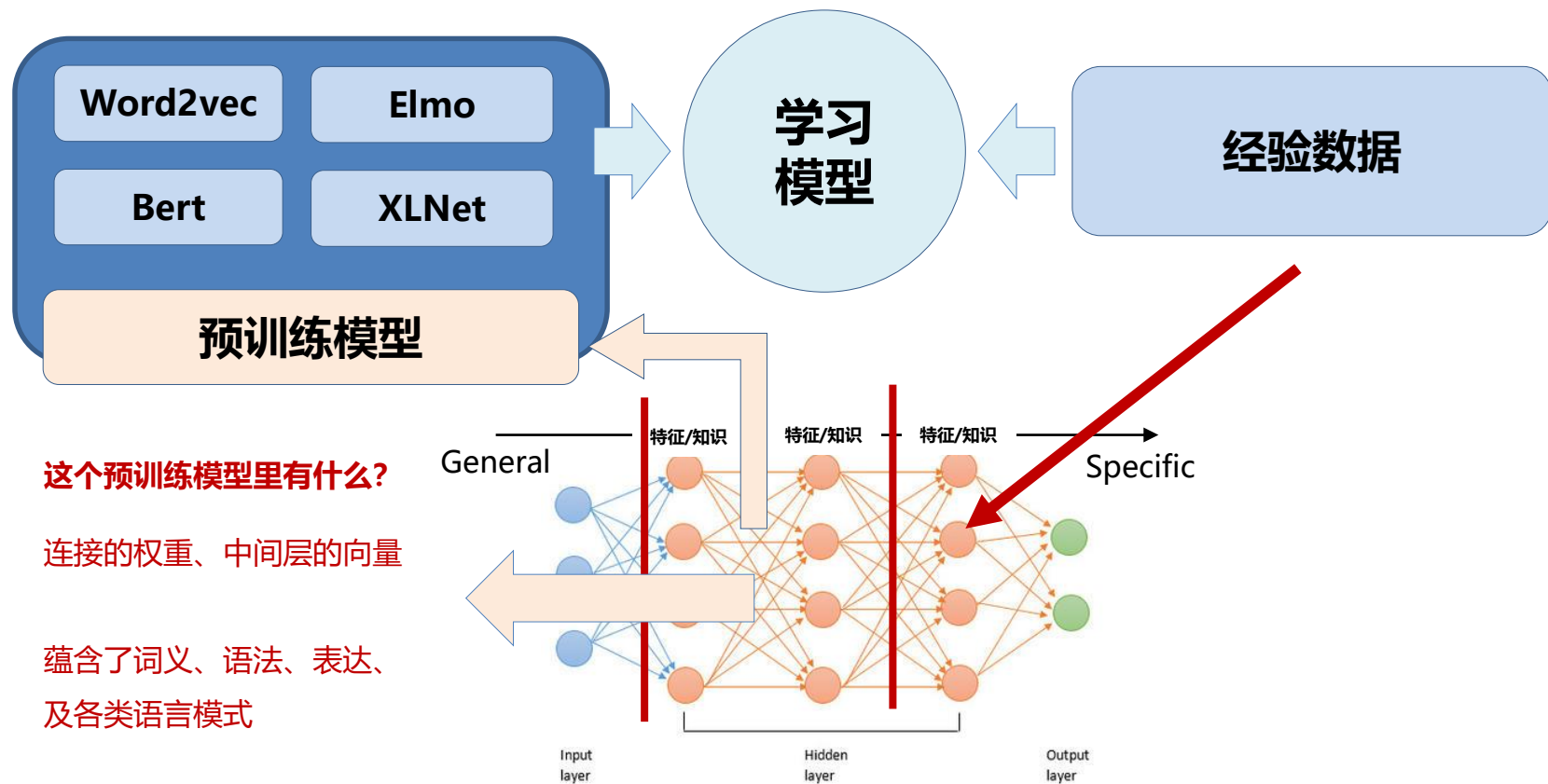
NLP:

- 词义
- 语法
- 表达习惯
- ...

何为预训练模型?



何为预训练模型?



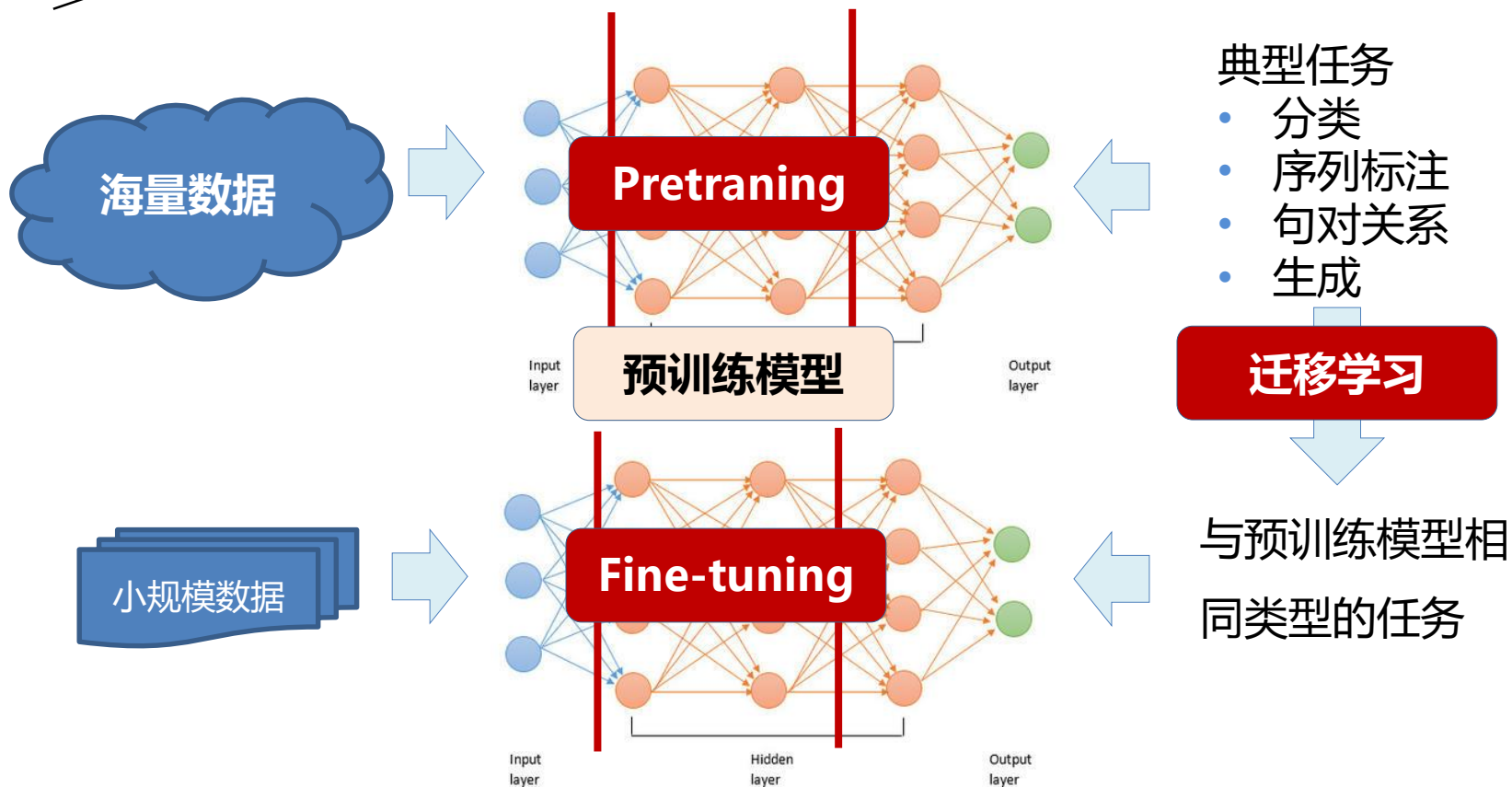
这个预训练模型里有什么?

连接的权重、中间层的向量

蕴含了词义、语法、表达、
及各类语言模式

端到端 end-end

何为预训练模型?



最简单的预训练形式：自编码器

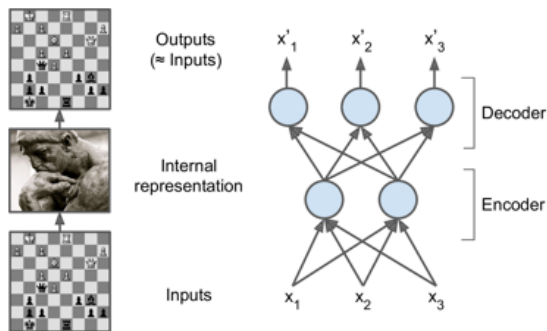


图15-1 象棋大师的记忆模式（左）和一个简单的自编码器

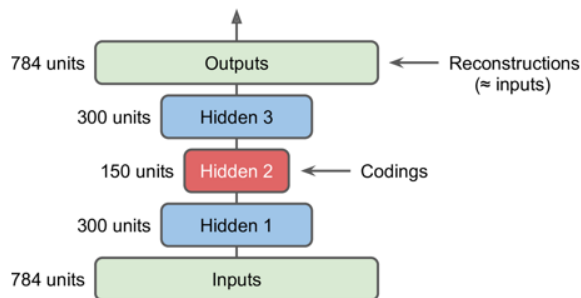
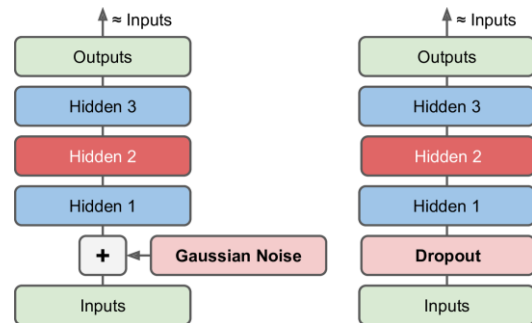


图15-3 栈式自编码器



训练目标：让输出等于输入
Encoder：编码
Decoder：解码

中间层：编码表示 Coding

增加噪音，更好的防止AD
将输入复制到输出。

(Bert的思想与此类似)

Word2vec

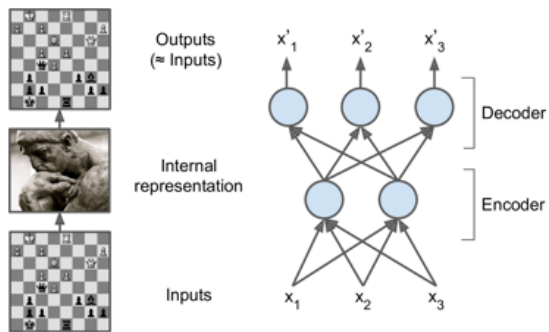
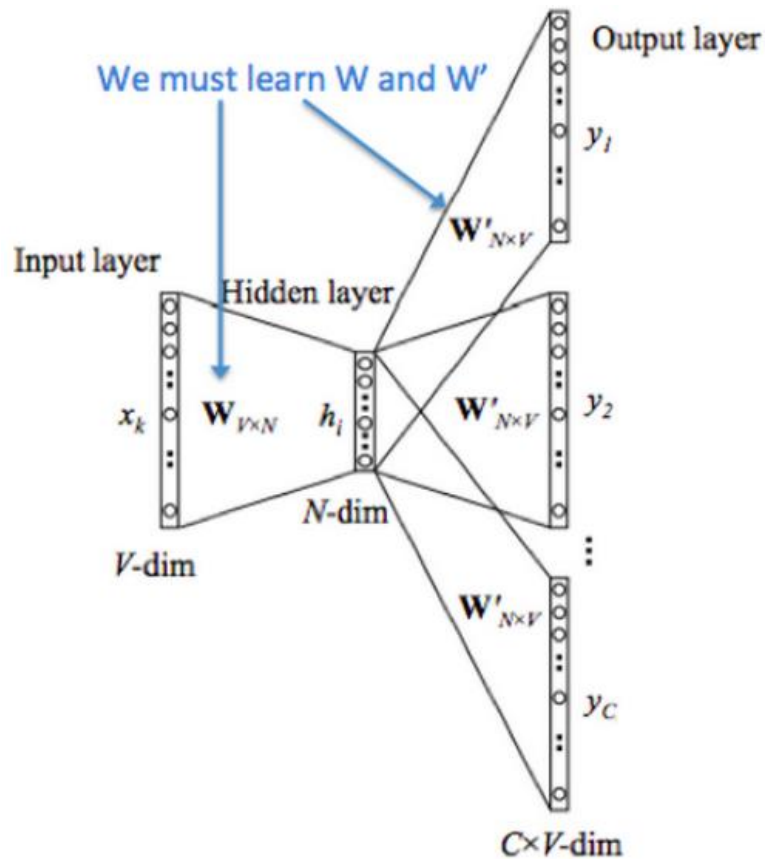
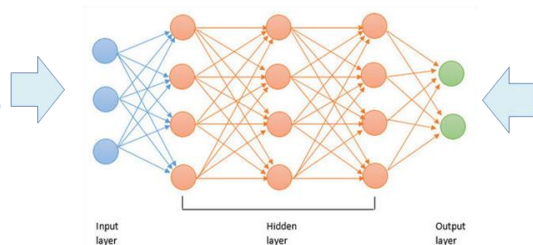
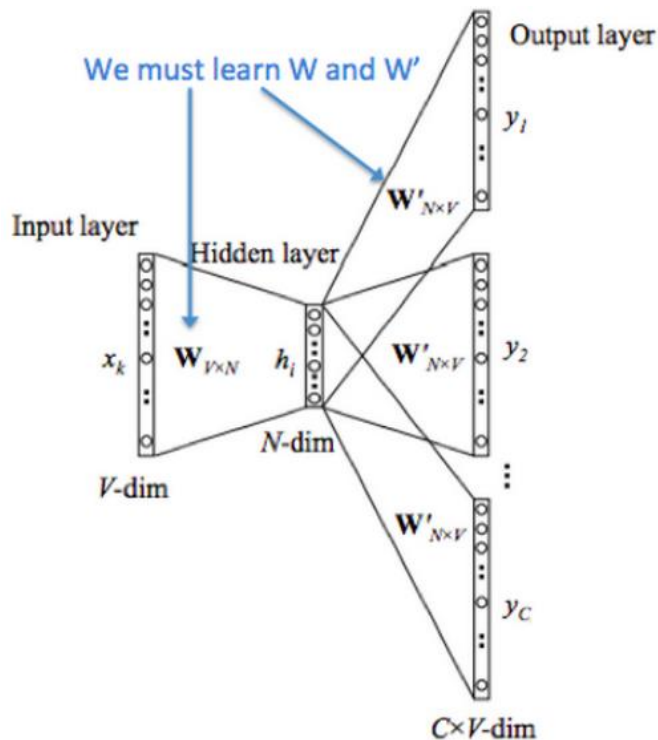


图15-1 象棋大师的记忆模式（左）和一个简单的自编码器



Word2vec



典型任务

- 分类
- 序列标注
- 句对关系
- 生成

输出端任务：用中心词预测上下文 (Skip-gram) 或者反过来 (C B O W)，而不是预测自己。这是一个语言模型任务（也算是一个分类任务）。

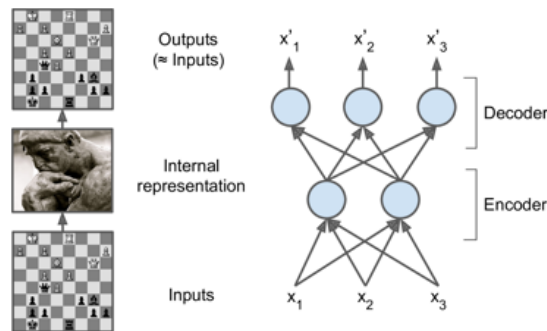
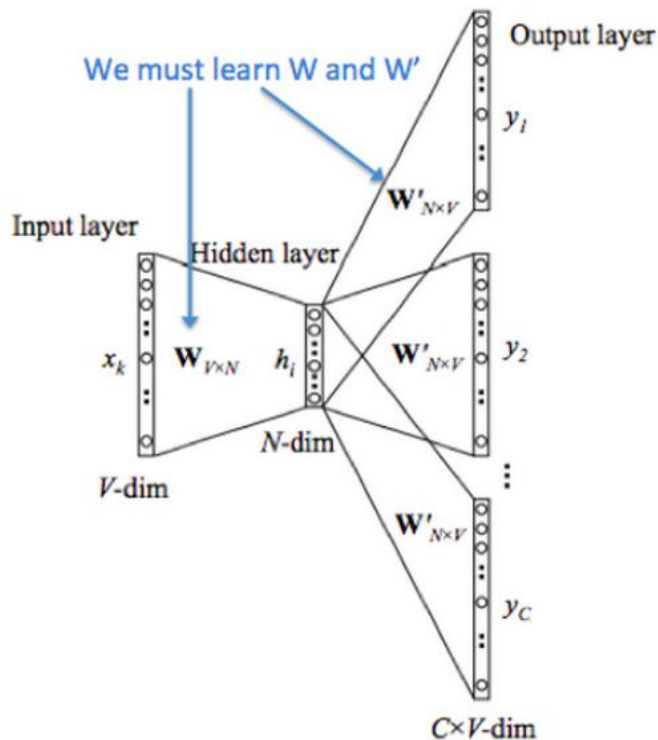


图15-1 象棋大师的记忆模式（左）和一个简单的自编码器

Word2vec



问题：如何根据上下文区分词汇的歧义？

Word2vec对每个词汇只产生一个向量

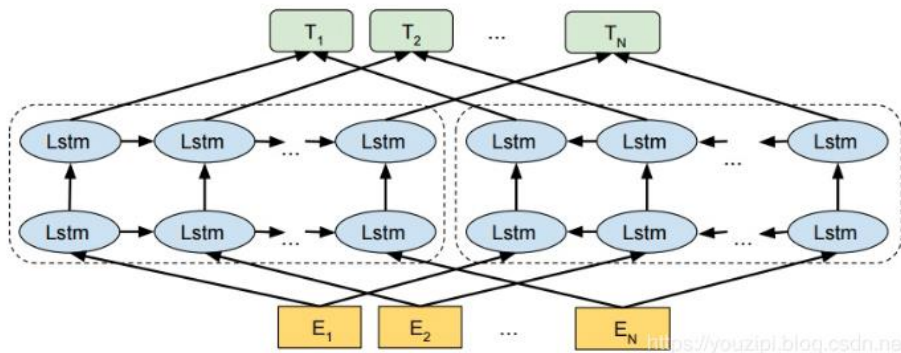
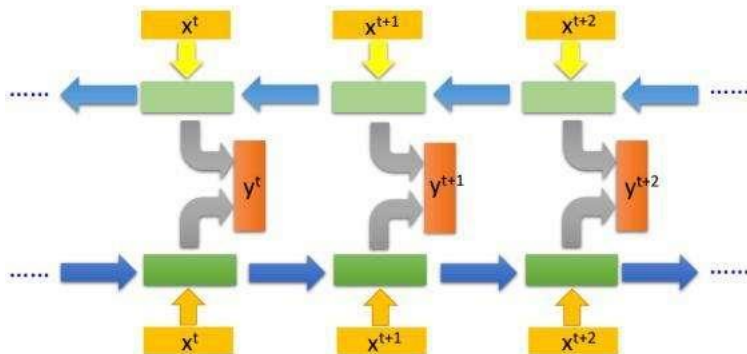
- Skip-gram: 输入是一个词，不需要上下文
- CBOW: 虽然是根据上下文预测一个词，但是将这个的所有上下文当作一个整体来看待。

Elmo

Embedding from Language Models

提出了**Bi-Lstm**，以语言模型为训练任务（根据一个词两侧的上下文预测一个词）。

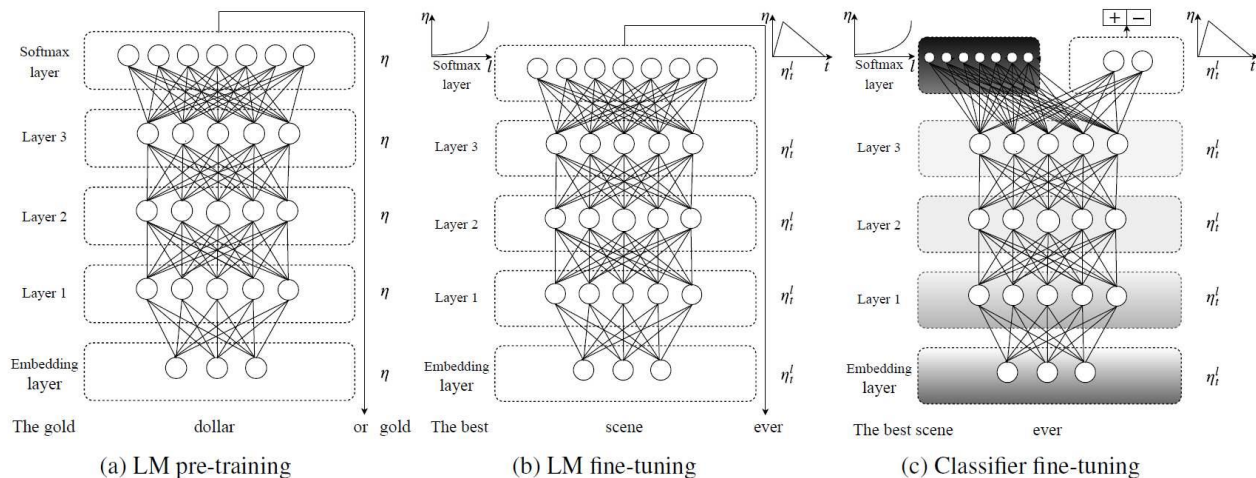
训练好的模型不会对每一个词产生一个全局的向量。
给定训练好的Elmo，要想知道一个词的词向量，必须输入一个句子，然后根据这个句子的上下文生成其中每个词的向量。



ULMFiT

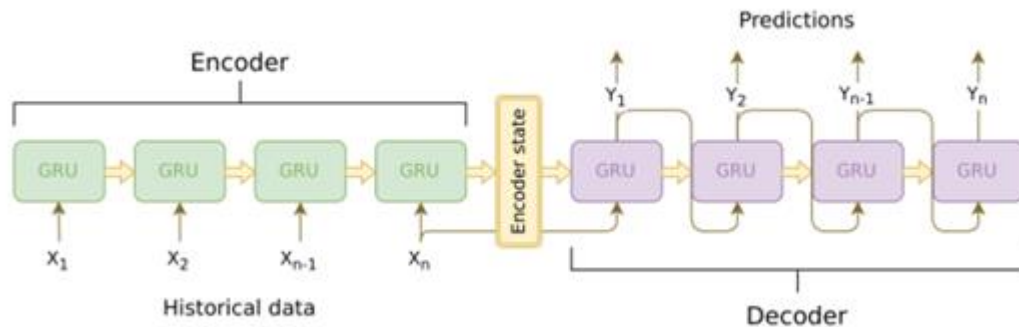
Universal Language Model Fine-tuning for Text Classification

在Elmo的基础上，接一个文本分类的下游任务进行预训练。因此得到的词向量表示理论上更适合分类任务，不过泛化能力弱于Elmo



如何用一个向量来表示一个句子?

- RNN: 不能准确捕捉远距离依赖, 不能并行
- 解决并行问题: 给每个词编码, 然后在词编码上用NN输出一个向量, NN可以并行。可以并行的网络可以做的更深
- 解决远距离依赖问题: 给每个词编码的时候用注意力机制



Transformer

Transformer是一个典型的Encoder-Decoder模型，最初用于机器翻译。其中间部分（Encoder的输出），是一个句子的向量表示。因此，Transformer的Encoder部分可以用作句子向量的预训练模型。

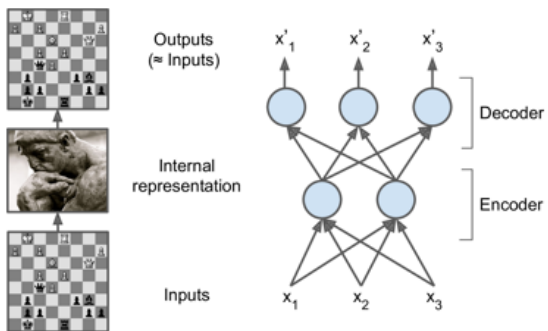
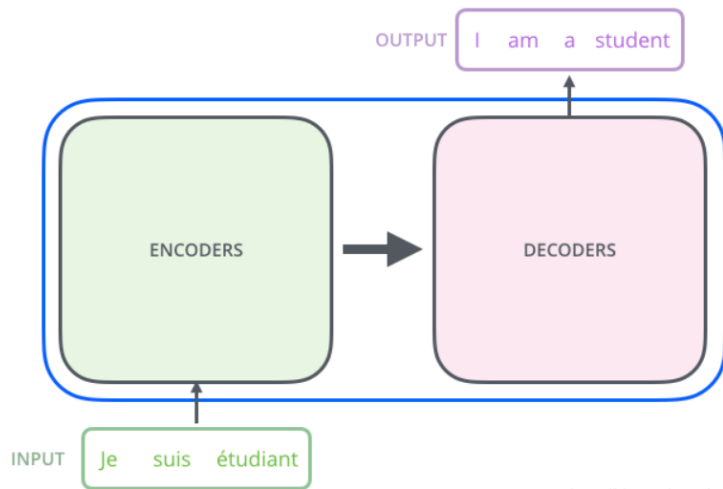


图15-1 象棋大师的记忆模式（左）和一个简单的自编码器

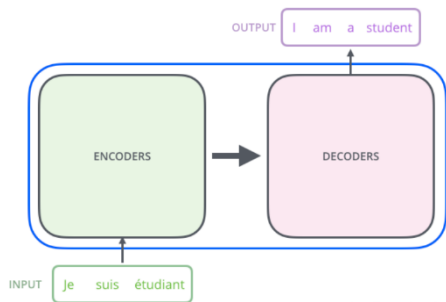


https://blog.csdn.net/qq_41664845

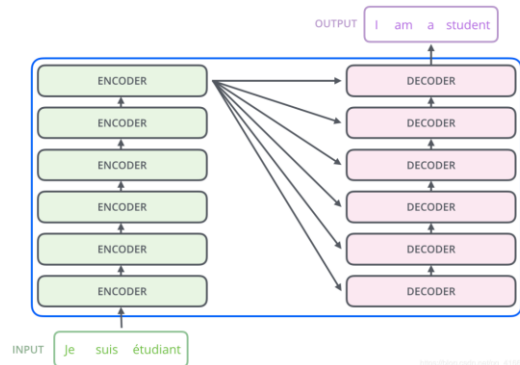
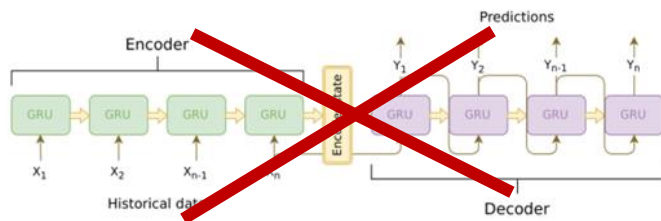
Transformer

经典Transformer的编码和解码各六层。解码部分每一层包含两个小层，分别是Self-attention层，和前向神经网络。

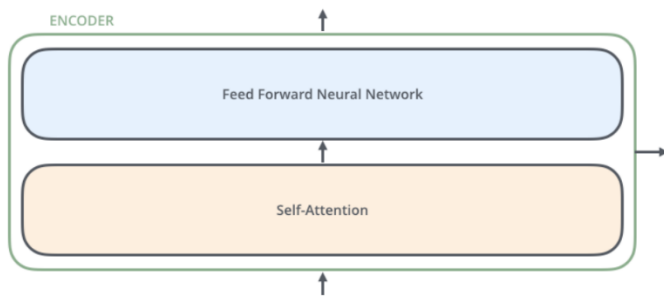
Self-attention层



https://blog.csdn.net/qz_41864845



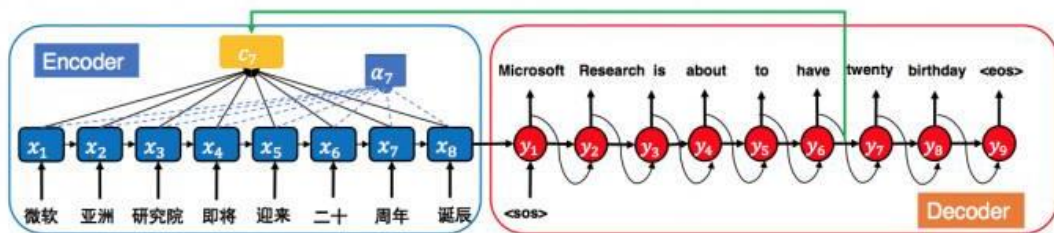
https://blog.csdn.net/qz_41864845



https://blog.csdn.net/qz_41864845

Transformer

什么是Self-attention（自注意力机制）？

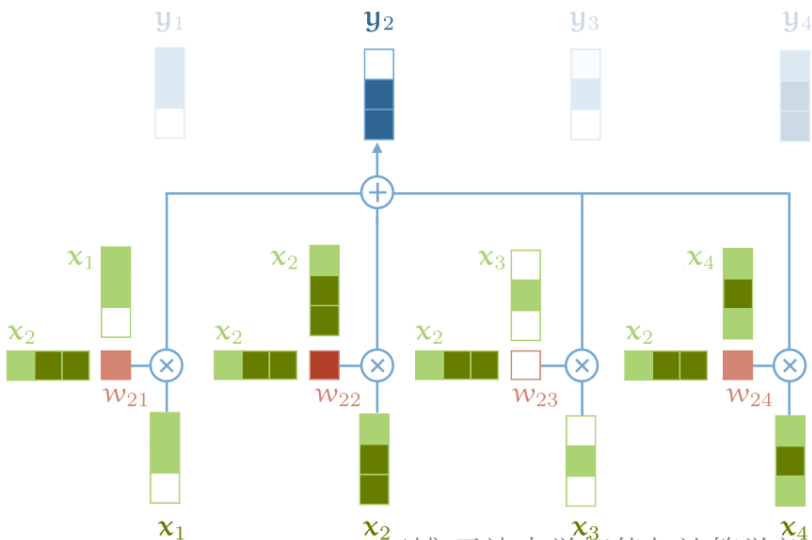
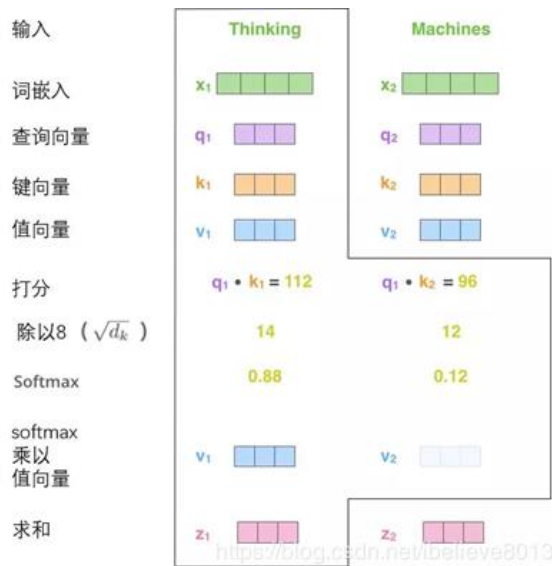


这是Sequence2Sequence方法中的注意力机制的图示。如果右边的红色输出序列不是另一个序列，而就是左边的蓝色序列的词向量，那就是自注意力机制了。

如果自注意力层输入的词向量是用全局语料训出来的（例如word2vec），每个词的向量蕴含了这个词在整个预料中的上下文信息。自注意力层输出的词向量是用这个句子的上下文获得的，每个词的向量蕴含的是这个句子中决定这个词含义的关键上下文。所以自注意力的是在词的基本含义的基础上，在特定句子上下文中进行微调。

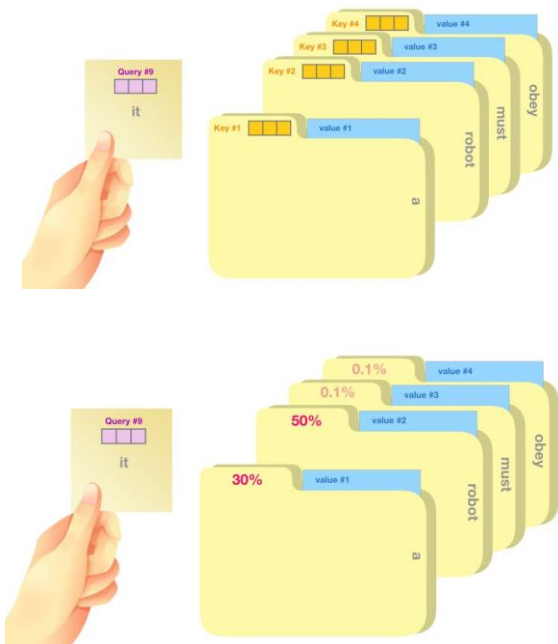
Transformer




















自注意力机制中每个词有三个输入向量：值向量、键向量和查询向量。在决定一个词A的向量时，用这个词的查询向量与句子中每个词的键向量匹配来决定每个词的attention权重，然后将每个词的值向量根据权重加权，形成A的向量。



Transformer

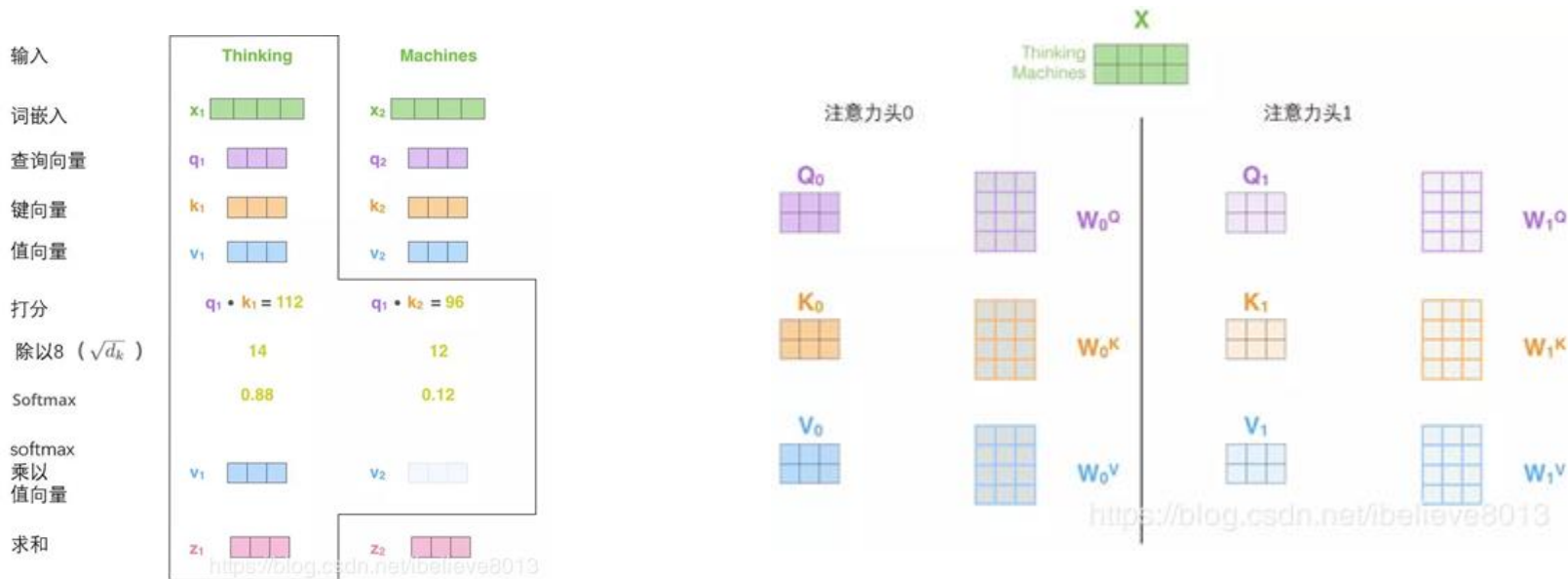
自注意力机制中每个词有三个输入向量：值向量、键向量和查询向量。在决定一个词A的向量时，用这个词的查询向量与句子中每个词的键向量匹配来决定每个词的attention权重，然后将每个词的值向量根据权重加权，形成A的向量。



Word	Value vector	Score	Value X Score
<s>		0.001	
a		0.3	
robot		0.5	
must		0.002	
obey		0.001	
the		0.0003	
orders		0.005	
given		0.002	
it		0.19	
		Sum:	

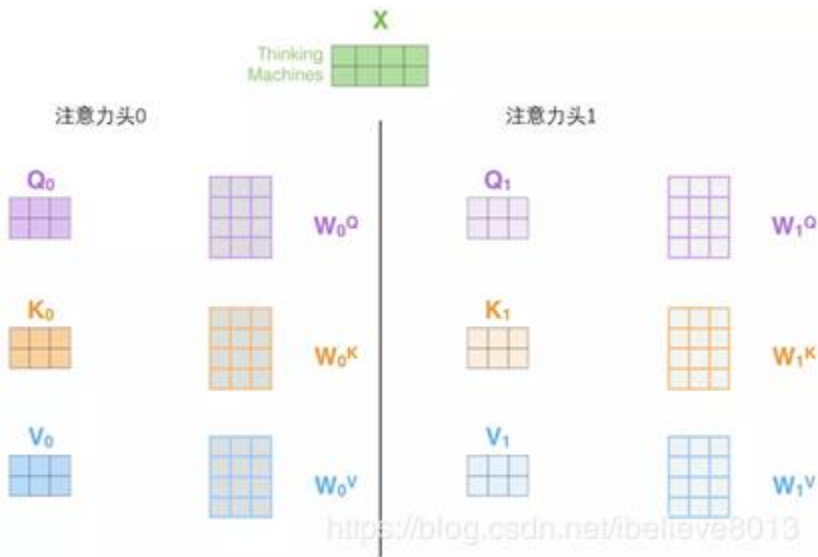
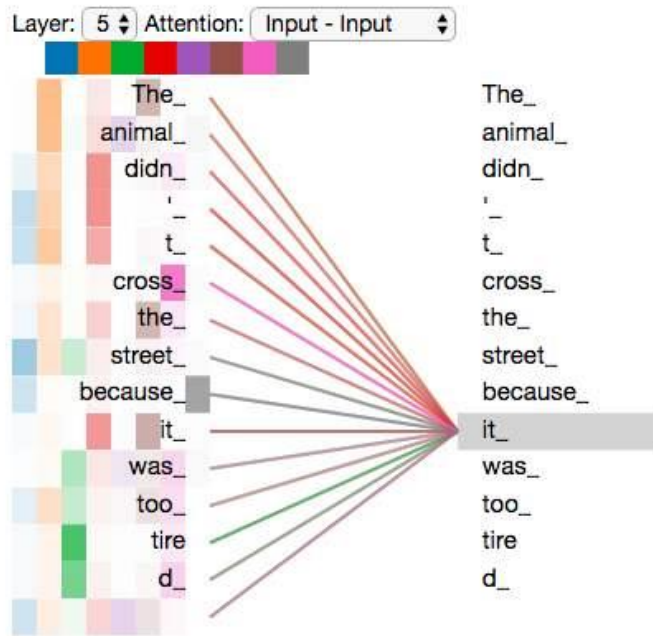
Transformer

Transformer的注意力是多头的。也就是每个词有好几组不同的值向量，键向量和查询向量。经过训练，不同的头有可能关注到不同的问题，例如，有的头关注到指代关系，有的关注到语法依赖，有的关注到语义依赖。



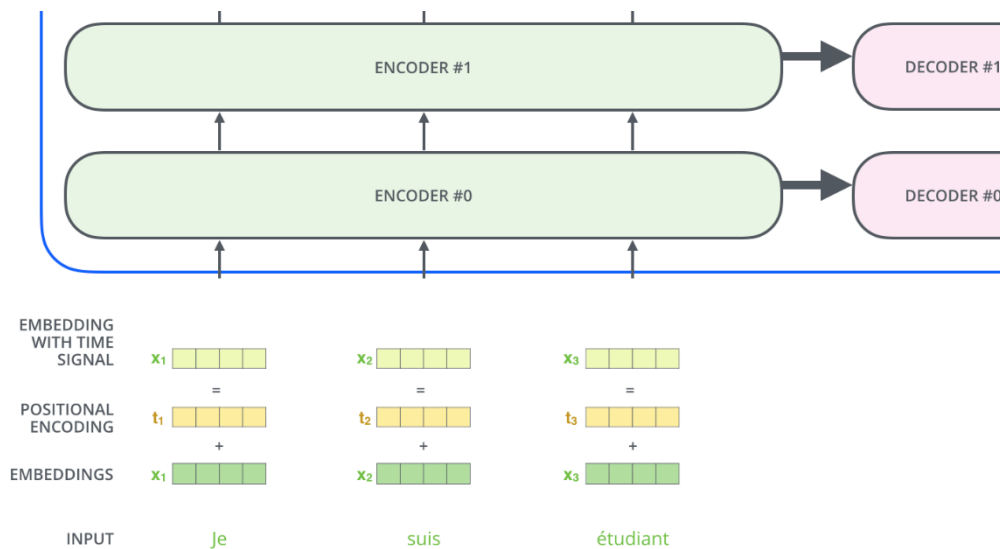
Transformer

Transformer的注意力是多头的。也就是每个词有好几组不同的值向量，键向量和查询向量。经过训练，不同的头有可能关注到不同的问题，例如，有的头关注到指代关系，有的关注到语法依赖，有的关注到语义依赖。



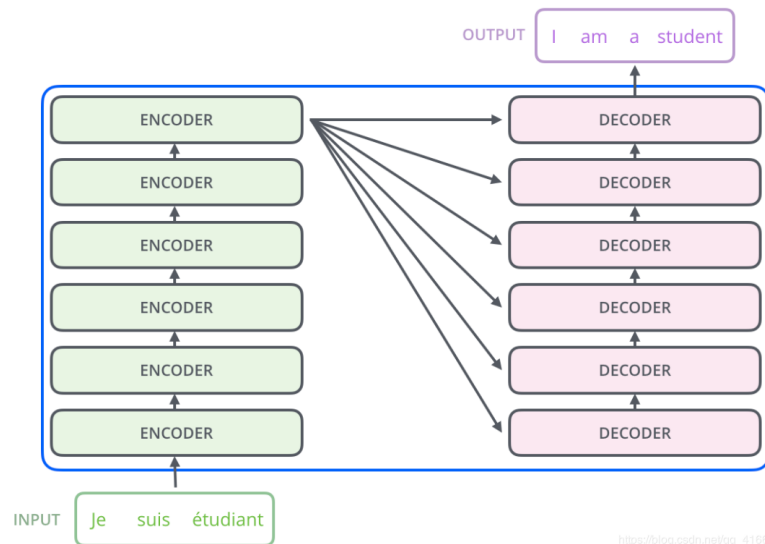
Transformer

Transformer还会将单词在句子中的位置进行编码（Positional Encoding），然后将编码与词向量相加做为输入。

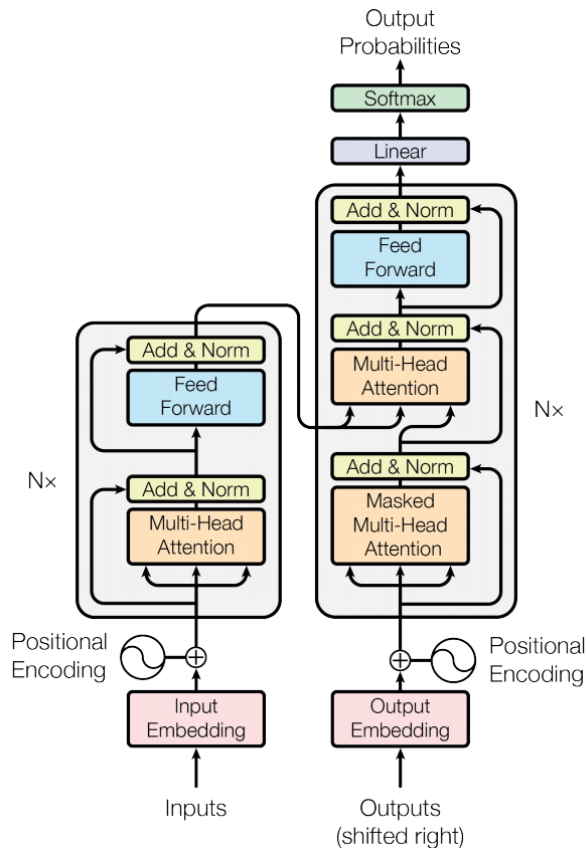


Transformer

Transformer整体结构



https://blog.csdn.net/qgq_61664845



GPT

GPT将各类典型NLP任务都转化为输入为串，输出为类别的分类任务，然后将Transformer做为预训练模型训练分类任务，得到新的预训练模型。

GPT用的是单向Transformer
(Attention不能看到后面的词)

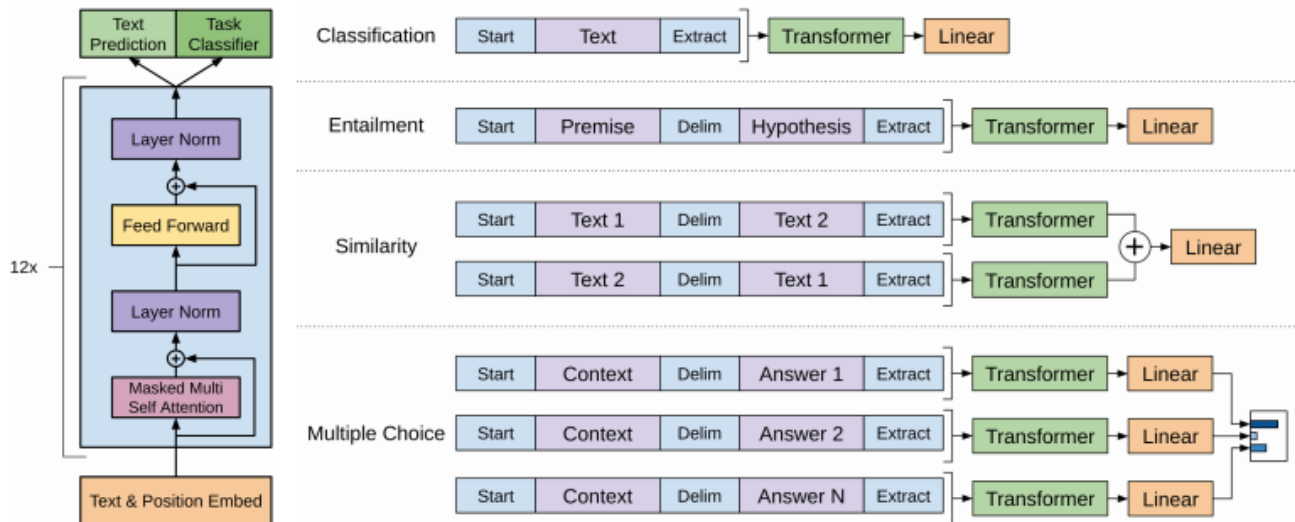


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

BERT

BERT以双向Transformer为基础。在词汇级，不做预测下一个词的语言模型任务，而是做完型填空任务。需要填写的词用[MASK]表示，称为 Masked Language Model任务。这个任务用来训练词向量。

RNN模型中[MASK]标记对编码影响比较大，Transformer中的attention机制有机会自动的赋予[MASK]低权重。

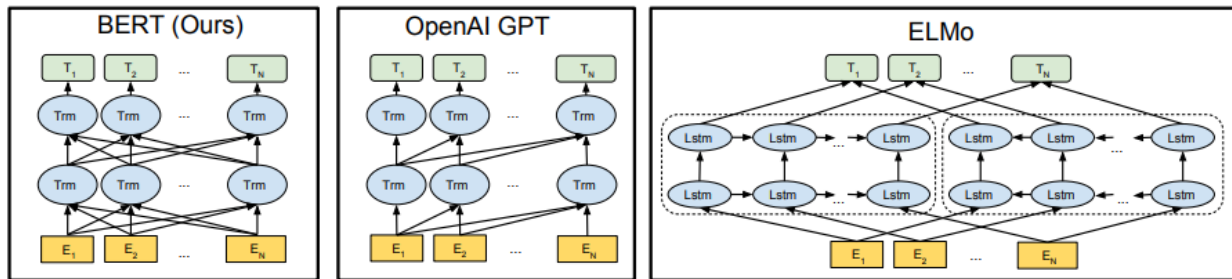


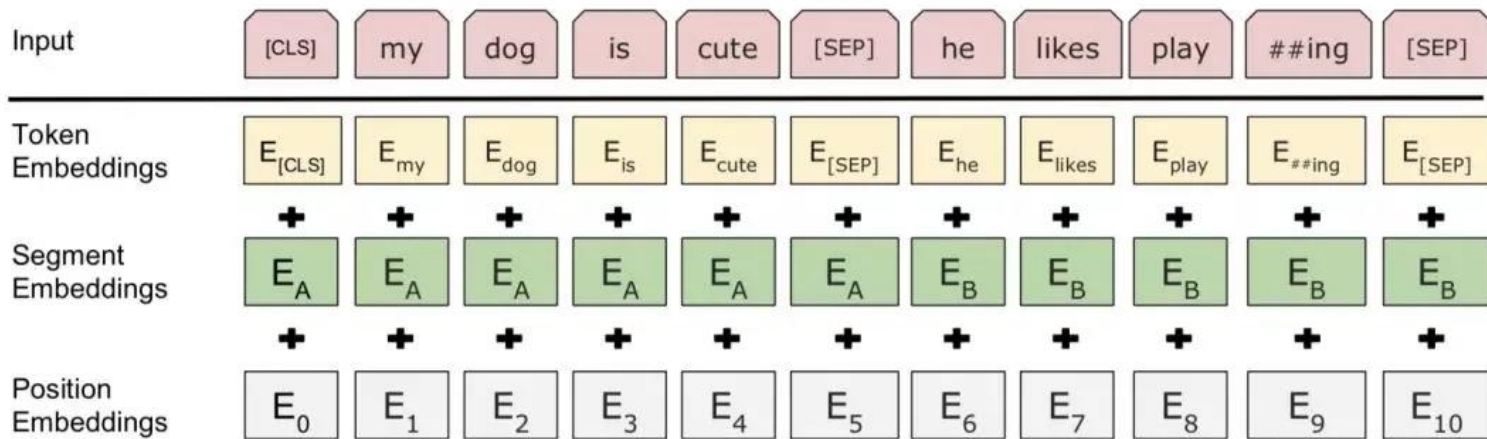
Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

<https://blog.csdn.net/liuy9803>

BERT

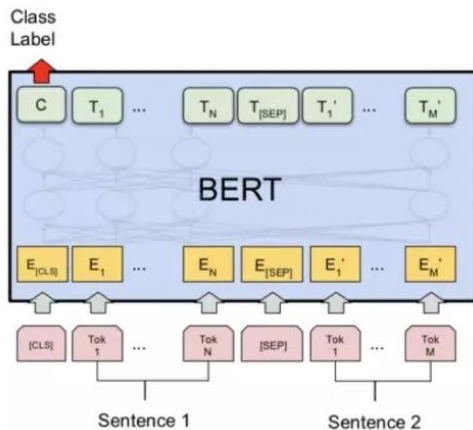
BERT的另一个预训练任务是句子向量表示，通过增加[CLS]和[SEP]标记来构建单句或句对输入。训练后，[CLS]对应的向量就是句子或句对的向量。不同的预训练任务都可以接到这个句子/句对表示上，因此BERT除了能够做完形填空的词汇级任务，还可以做很多种不同的句子级任务，是一个多任务模型（Transformer的预训练任务特定为机器翻译）。

实际输入由三部分构成：Token词向量，Segment向量表示词汇属于那句话，Position向量表示词汇的具体位置。Segment和Position向量也是初始化输入一个序号，然后由模型训练出向量表示。

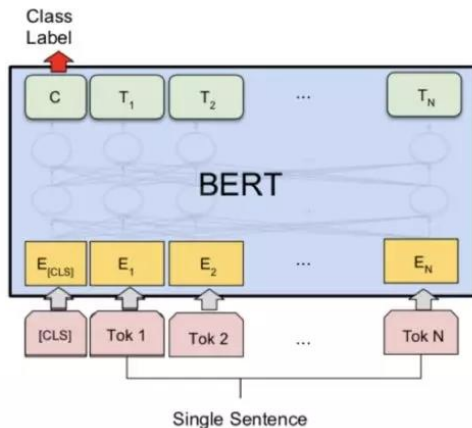


BERT

BERT的另一个预训练任务是句子向量表示，通过增加[CLS]和[SEP]标记来构建单句或句对输入。训练后，[CLS]对应的向量就是句子或句对的向量。不同的预训练任务都可以接到这个句子/句对表示上，因此BERT除了能够做完形填空的词汇级任务，还可以做很多种不同的句子级任务，是一个多任务模型（Transformer的预训练任务特定为机器翻译）。



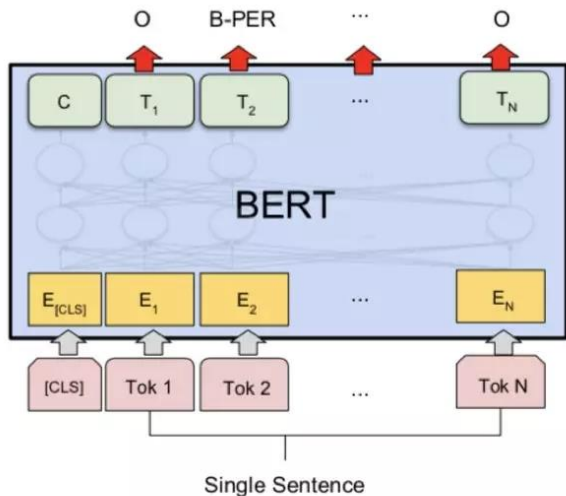
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



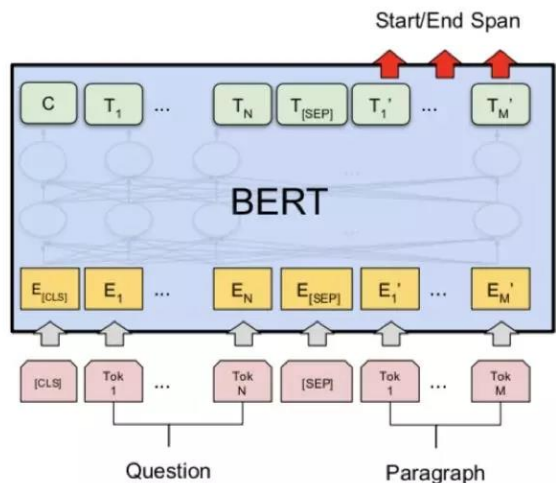
(b) Single Sentence Classification Tasks:
SST-2, CoLA

BERT

BERT的另一个预训练任务是句子向量表示，通过增加[CLS]和[SEP]标记来构建单句或句对输入。训练后，[CLS]对应的向量就是句子或句对的向量。不同的预训练任务都可以接到这个句子/句对表示上，因此BERT除了能够做完形填空的词汇级任务，还可以做很多种不同的句子级任务，是一个多任务模型（Transformer的预训练任务特定为机器翻译）。



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER



(c) Question Answering Tasks:
SQuAD v1.1

ERNIE 1.0

BERT是随机MASK词汇或字，没有考虑到粗粒度的语义单元。比如“我要买苹果手机”，BERT模型将“我”，“要”，“买”，“苹”，“果”，“手”，“机”每个字都统一对待，随机mask，丢失了“苹果手机”是一个很火的名词这一信息。ERNIE基于外部知识库（如词典）来规范MASK。

Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]
Phrase-level Masking	Harry	Potter	is	[mask]	[mask]	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]

Figure 2: Different masking level of a sentence

ERNIE 1.0

ERNIE还针对对话任务做了一些改进，不再构建如同 “[CLS] + Sentence_A + [SEP] + Sentence_B + [SEP]” 的句子对，而是如同 “[CLS] + Query + [SEP] + Response_A + [SEP] + Response_B + [SEP]” 的对话三元组，是否上下文连续的二分类训练目标转为预测该对话是否真实 (real/fake)。三元组随机地采用 QRQ、QRR、QQR 其中一种构建形式，上面的例子便是其中的 QRR。

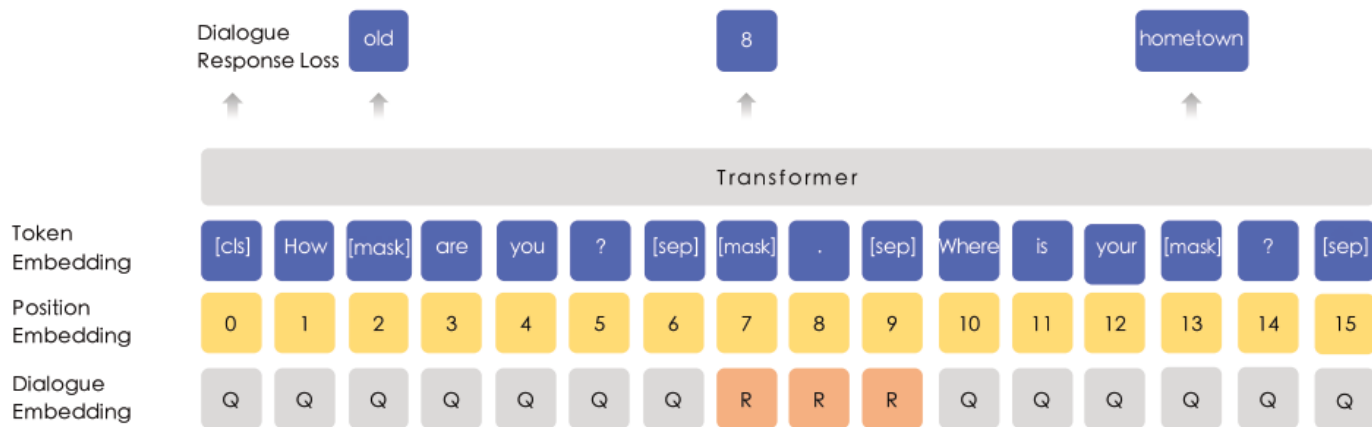


Figure 3: Dialogue Language Model. Source sentence: [cls] How [mask] are you [sep] 8 . [sep] Where is your [mask] ? [sep]. Target sentence (words the predict): old, 8, hometown)

ERNIE 2.0

ERNIE2.0 在1.0 的基础上构建了更加丰富的预训练任务，并且通过多任务的连续训练来提高模型的泛化能力。

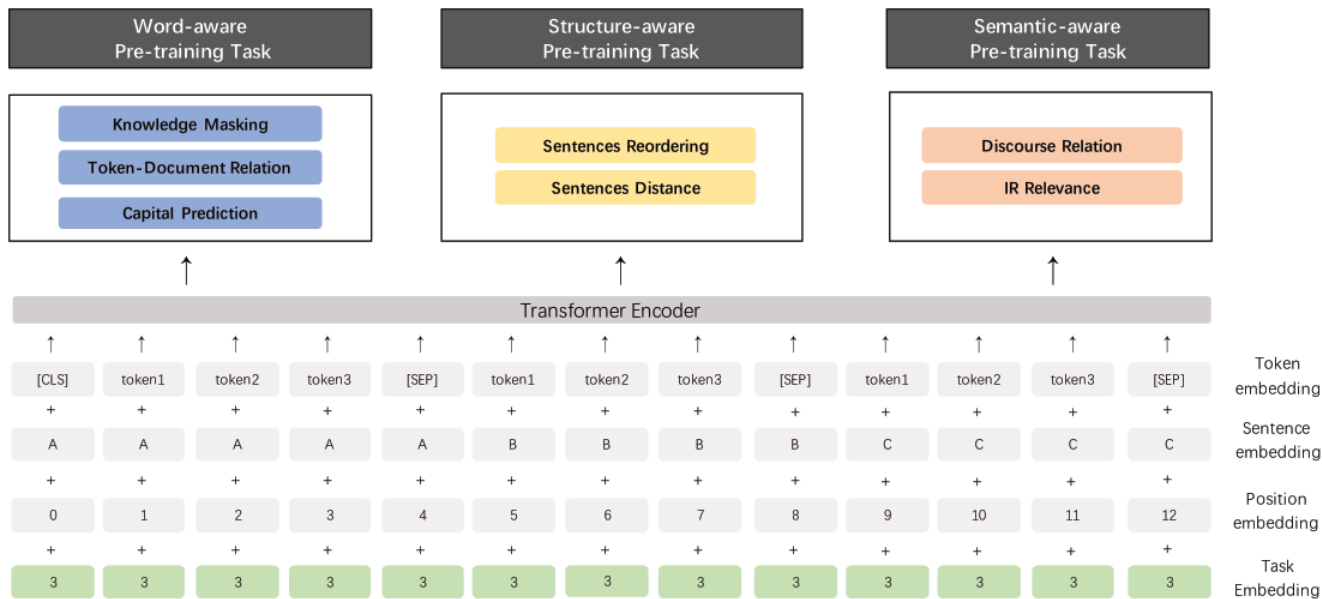


Figure 3: The structure of the ERNIE 2.0 model. The input embedding contains the token embedding, the sentence embedding, the position embedding and the task embedding. Seven pre-training tasks belonging to different kinds are constructed in the ERNIE 2.0 model.

ERNIE 2.0

ERNIE2.0 在1.0 的基础上构建了更加丰富的预训练任务，并且通过多任务的连续训练来提高模型的泛化能力。

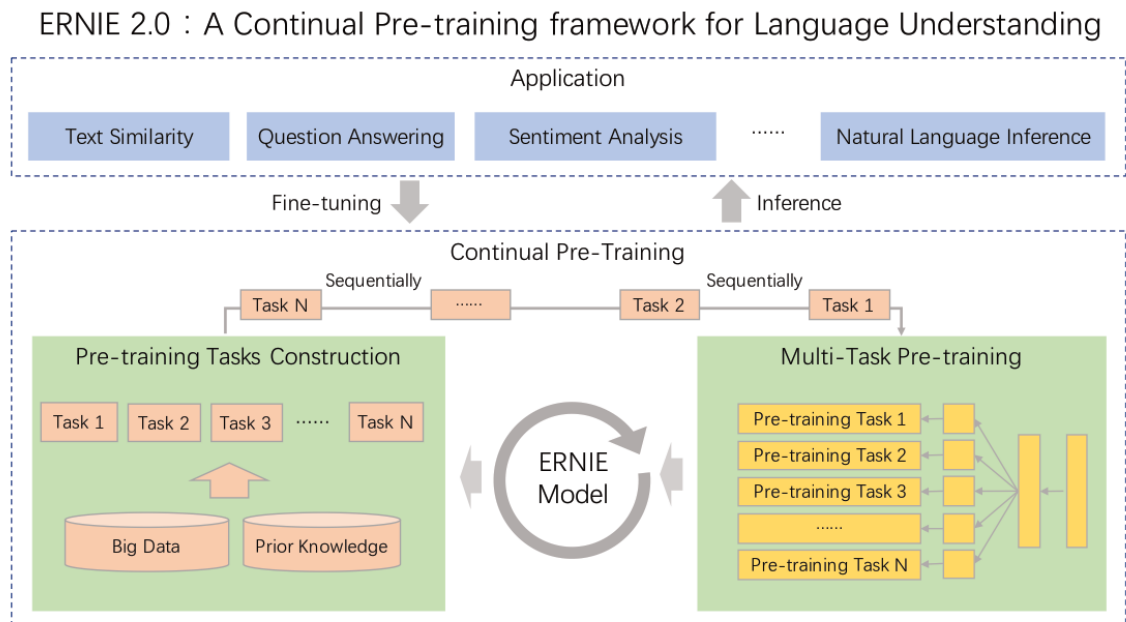
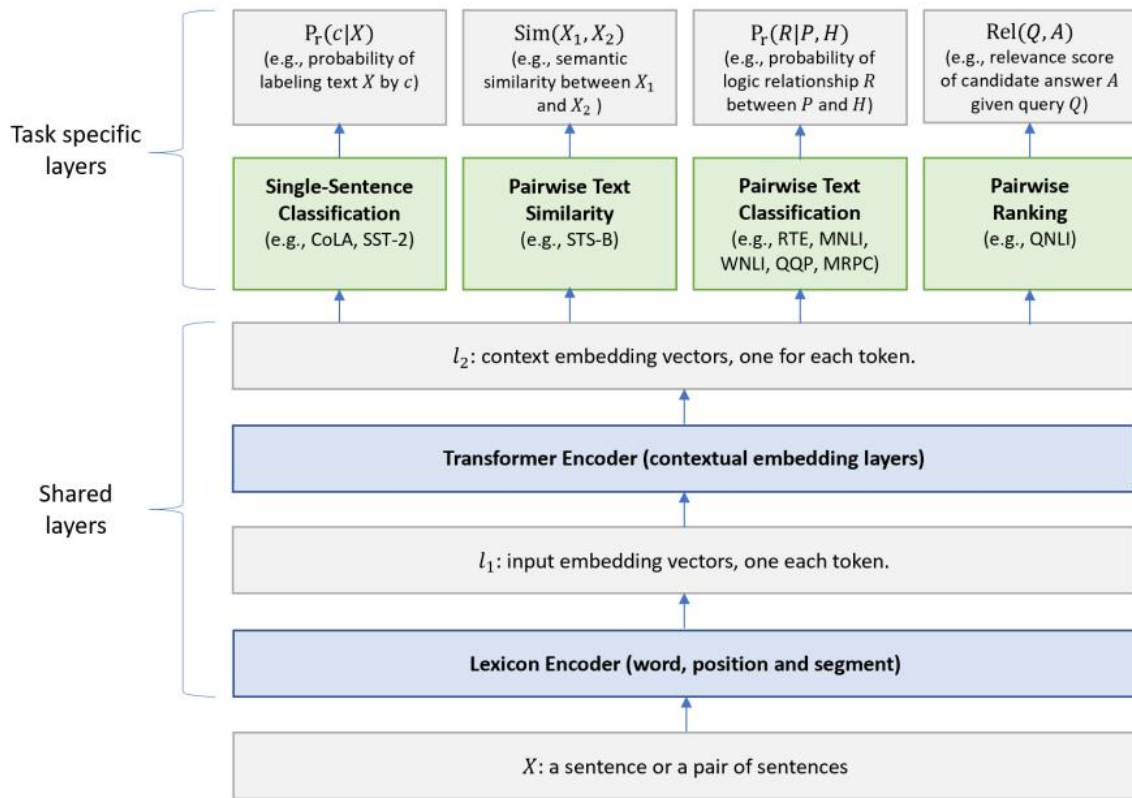


Figure 1: The framework of ERNIE 2.0, where the pre-training tasks can be incrementally constructed, the models are pre-trained through multi-task learning, and the pre-trained model is fine-tuned to adapt to various language understanding tasks.

MTDNN

以BERT作为共享层，叠加具体任务
多任务训练。



XLnet

GPT, Elmo: 自回归语言模型 (AR, Autoregressive LM), 不能使用双向上下文, 但适合生成式任务

BERT: 自编码语言模型 (AE, Autoencoder LM), 可以使用双向上下文, 但与生成任务匹配不好。

如何两全其美?

- 将词汇序列随机重拍, 使得后面的词可以换到前面
- 借用attention机制的便利, 通过关注随机序列中排在目标词前面的词, 来实现这种排列变换

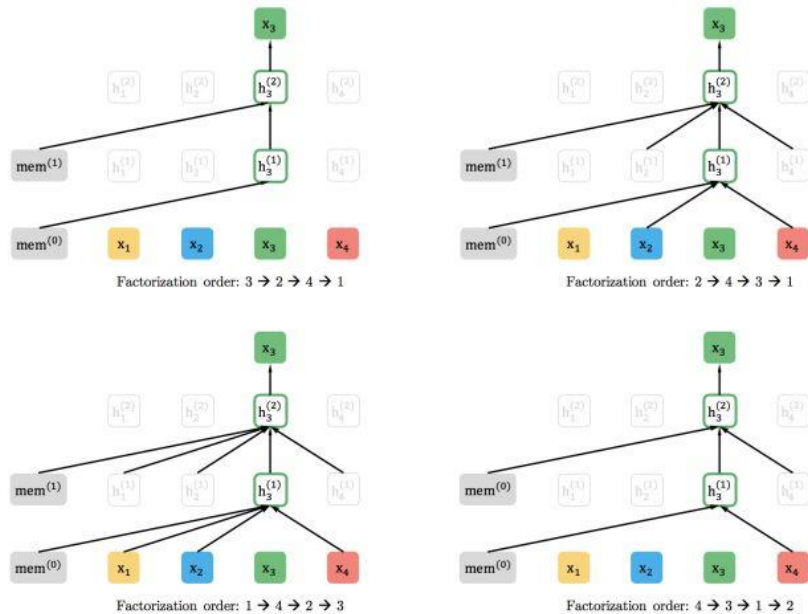
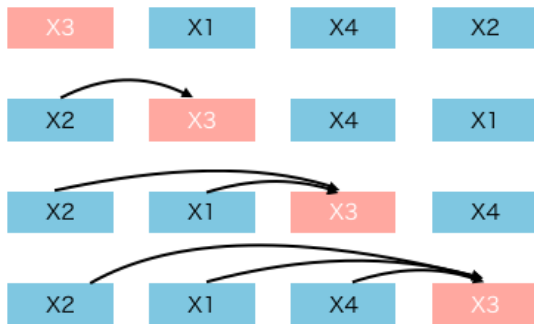


Figure 1: Illustration of the permutation language modeling objective for predicting x_3 given the same input sequence x but with different factorization orders.

XLnet

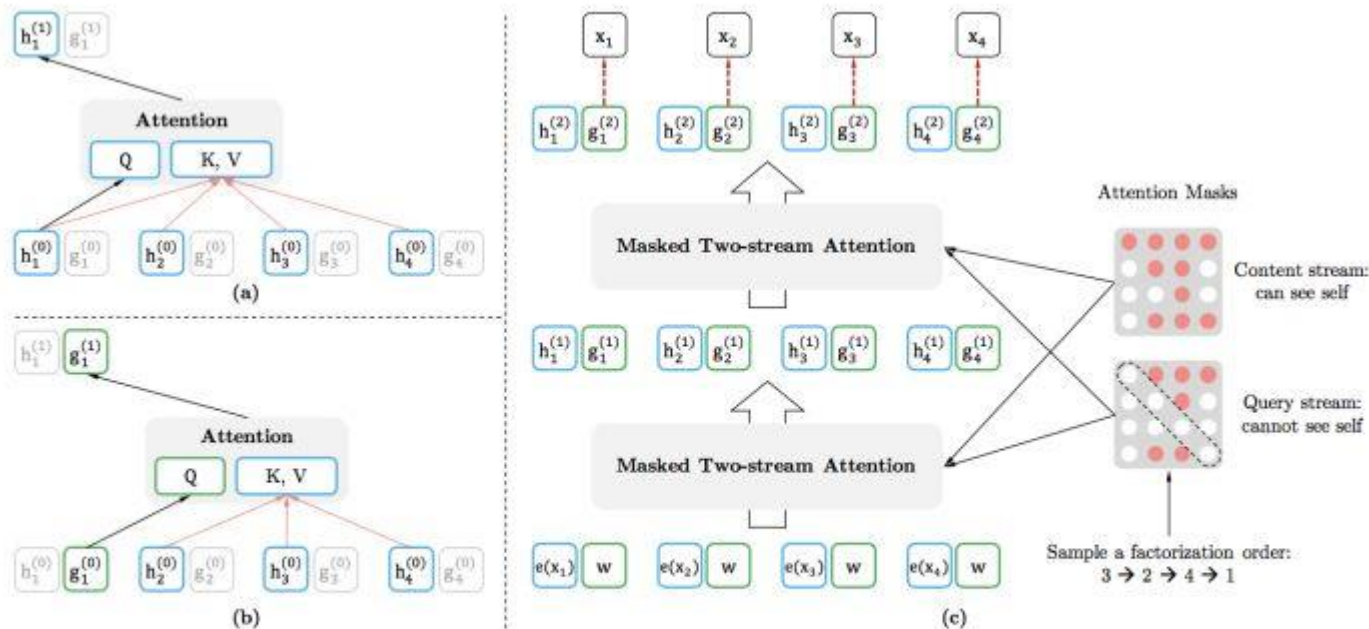


Figure 2: (a): Content stream attention, which is the same as the standard self-attention. (b): Query stream attention, which does not have access information about the content x_{z_t} . (c): Overview of the permutation language modeling training with two-stream attention.

十项全能模型 decaNLP 2018: 将所有NLP任务都转化为问答

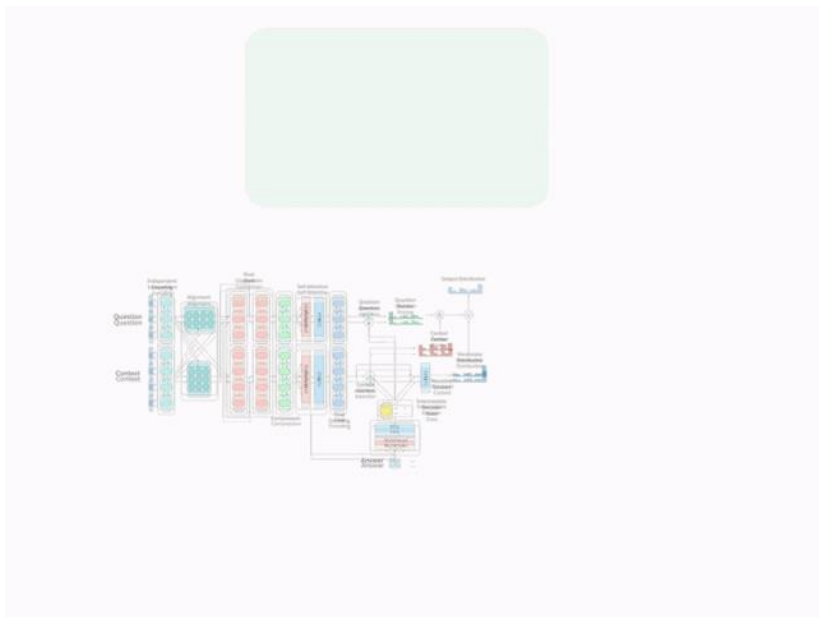


翻译一句话, 就将要翻译的句子作为语境信息,
把整个任务转化为一个问题: “这句英语翻译成
成德语是什么?”

官方介绍 (英文): <https://einstein.ai/research/the-natural-language-decathlon>

论文:
<https://einstein.ai/static/images/pages/research/decaNLP/decaNLP.pdf>

GitHub: <https://github.com/salesforce/decaNLP>





Thank You!