中国科学技术大学
University of Science and Technology of China

《人工智能数学原理与算法》

第6章 自监督学习

6.3 自回归语言建模

凌震华

zhling@ustc.edu.cn

目录

从一个问题出发

- 机器如何判断这两句话的合理性?

我学习人工智能课程　　*我智能课程人工学习

- 可以通过检查句子是否符合语法规则、语义是否合理等方式
- 另一种方法是计算句子的概率
  - 句子的数量是无穷的，训练集无法覆盖所有可能的句子

    → 句子是单词的序列，句子概率可转化为单词的联合概率
  - 是否存在某种独立性假设能简化计算?
  - 如何评估所建立的句子概率计算模型?

语言模型(Language Model)

- 什么是语言模型?
  - 对于人类语言的内在规律建模，从而准确预测单词序列的概率
- 人类语言的内在规律
  - 单词可以任意组合吗? **No!**

    我学习人工智能课程　　*我智能课程人工学习
  - 单词之间的组合顺序是一成不变的吗? **No!**

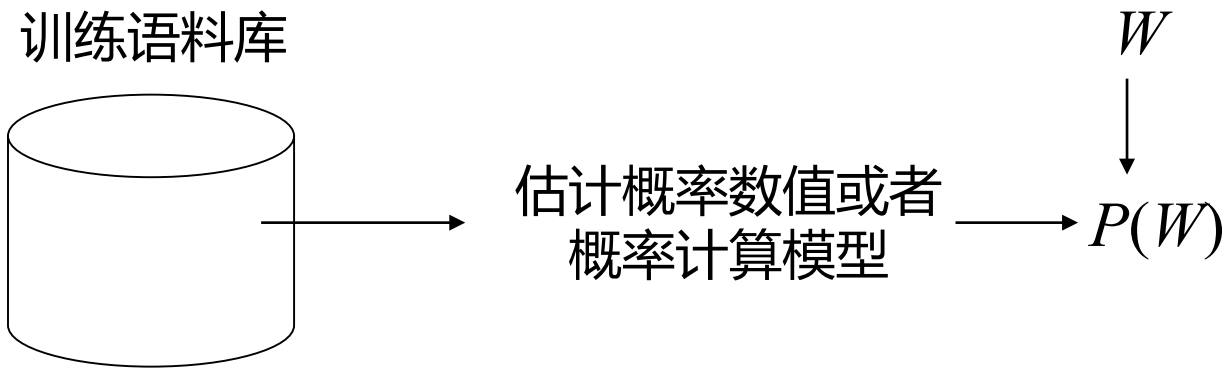
    我今天很开心　　今天我很开心
- 似乎存在一些潜在的概率形式的规则来判断句子是否规范

概率语言模型

- 目标：建立一个统计模型，用来计算语言中一个基本单元（token，通常是单词）序列 $W = w_1 w_2 \cdots w_n$ 的概率

$$P(W)$$

- 基本途径



如何计算 P(W)

- 如何计算句子概率
  - P(我学习人工智能课程)
- 由于句子是单词序列，所以等同于计算单词(token)的联合概率
  - P(我学习人工智能课程) = P(我，学习，人工，智能，课程)
- 核心思想: 使用概率的链式法则(chain rule)

链式法则

- 条件概率的定义

$$P(B|A) = \frac{P(A,B)}{P(A)}$$

$$P(A,B) = P(A)P(B|A)$$

- 更多变量

$$P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$$

- 链式法则

$$P(X_1, X_2, \cdots, X_n) = P(X_1)P(X_2|X_1)\cdots P(X_n|X_1, \cdots, X_{n-1})$$

计算 P(W)

- 应用链式法则

$$P(w_1 \cdots w_n) = \prod_{i=1}^{n} P(w_i | w_1 \cdots w_{i-1})$$

$$P(\text{我学习人工智能课程})$$

$$= P(\text{我})P(\text{学习}|\text{我})\,P(\text{人工}|\text{我学习})\,P(\text{智能}|\text{我学习人工})$$

$$P(\text{课程}|\ \text{我学习人工智能})$$

- 条件概率$P(w_i|w_1 \cdots w_{i-1})$对应自回归生成的自监督学习任务

  → **自回归语言建模**

如何获得条件概率

1. 直接存储

- 假设词汇表V中的单词个数为|V|
- 直接存储$w_1, w_2, \cdots w_{i-1}, w_i$为词汇表中每个单词时$P(w_i|w_1 \cdots w_{i-1})$的数值
- 共需要存储|V|^i个概率值

→ **n元语法语言模型**

2. 函数计算

- 使用函数$f(w_1, w_2, \cdots w_{i-1})$计算得到|V|个概率值$p_i(1 \leq i \leq |\mathrm{V}|)$
- 其中$w_1, w_2, \cdots w_{i-1}$可以是词汇表中任意单词
- p_i表示w_i为词汇表中第i个单词的概率

→ **神经网络语言模型**

目录

条件概率的直接估计

- 基于计数(count)的概率估计

$$P(\text{课程}|\text{我学习人工智能})=$$

$$\text{count}(\text{我学习人工智能课程}) \,/\, \text{count}(\text{我学习人工智能})$$

  - 对于句子中越靠后的单词，需要计数的单词序列越长
  - 越长的单词序列在训练语料库中出现的越稀疏
- 简化条件概率

$$P(\text{课程}|\text{我学习人工智能})= P(\text{课程}|\text{智能})$$

或者

$$P(\text{课程}|\text{我学习人工智能})= P(\text{课程}|\text{人工智能})$$

马尔科夫(Markov)假设

$$P(w_1 \cdots w_n) \approx \prod_{i=1}^{n} P(w_i | w_{i-k} \cdots w_{i-1})$$

- 我们将句子概率乘积中的每个条件概率简化为

$$P(w_i | w_1 \cdots w_{i-1}) \approx P(w_i | w_{i-k} \cdots w_{i-1})$$

- 即假设每个词出现的概率只受到它前面的 k 个词影响

两种最简单的情况：一元/ 二元语法模型

- 一元语法(unigram) 语言模型

$$P(w_1 \cdots w_n) \approx \prod_{i=1}^{n} P(w_i)$$

  - $k = 0$
  - 计算句子中每个单词出现的条件概率时不考虑之前单词

- 二元语法(bigram) 语言模型

$$P(w_1 \cdots w_n) \approx \prod_{i=1}^{n} P(w_i|w_{i-1})$$

  - $k = 1$
  - 计算句子中每个单词出现的条件概率时只考虑之前的一个单词

N元语法模型

- N元语法(N-gram) 语言模型

$$P(w_1 \cdots w_n) \approx \prod_{i=1}^{n} P(w_i | w_{i-N+1}, w_{i-N+2}, \ldots, w_{i-1})$$

  - $k = N - 1$
  - 计算句子中每个单词出现的条件概率时，只考虑其之前的$N - 1$单词

N元语法模型概率估计

- 基于计数(count)的概率估计
  - 以2元语法模型为例

$$P(w_i|w_{i-1}) = \frac{count(w_{i-1}w_i)}{count(w_{i-1})}$$

  - 示例

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

$P(\text{I}|\text{<s>}) = \frac{2}{3} = .67$ $P(\text{Sam}|\text{<s>}) = \frac{1}{3} = .33$ $P(\text{am}|\text{I}) = \frac{2}{3} = .67$

$P(\text{</s>}|\text{Sam}) = \frac{1}{2} = 0.5$ $P(\text{Sam}|\text{am}) = \frac{1}{2} = .5$ $P(\text{do}|\text{I}) = \frac{1}{3} = .33$

N元语法模型的性能评估

- 困惑度（perplexity）
  - 困惑度是基于语言模型计算的测试集概率，再根据单词数量进行归一化得到的结果

$$\begin{aligned} \mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}} \end{aligned}$$

  - 链式法则

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$$

  - 最小化困惑度等同于最大化概率，最好的语言模型是能够最好地预测未见测试集的模型

N元语法模型的性能评估

- 困惑度（perplexity）
  - 困惑度越低，语言模型性能越好
  - 例：训练集 3800 万个单词、测试集150 万个单词（《华尔街日报》语料）

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

N元语法模型与自监督学习的关系

- **N元语法模型采用自监督方式学习**
  - 自回归生成 (autoregressive generation) 任务
  - 模型具有生成能力

1. 根据条件概率P(w|<s>)随机采样生成一个词元x_1
2. 再根据条件概率P(w| x_2)随机采样生成一个词元x_2
3. 重复上述步骤，直到生成</s>
4. 最后将这些词元串联起来

<s> I
I want
want to
to read
read a
a book
book </s>

N元语法模型与自监督学习的关系

- **N元语法模型在应用范式上与自监督学习存在差异**
  - 自监督学习通常作为前置任务，将有用信息迁移至下游任务
  - N元语法模型通常用于在文本生成任务中，对生成结果进行评分
    - 语音识别、机器翻译… …

N元语法模型的局限性

- 优点：易于构建
  - N 元语法模型中概率估计过程相对简单，并且可以在海量数据上快速进行

- 缺点：上下文描述能力有限
  - “维数灾难”（Curse of Dimensionality）
    - 直接存储N元语法的条件概率，而N元语法数目与N成指数关系
    - 随着N增加，N元语法的稀疏性越发严重
  - 实际使用中N不会太大(N≤3)
  - 模型只能描述短距离的上下文信息，对于长距离依赖关系的处理能力较弱

目录

神经网络语言模型(Neural Language Model)

- 单词 w_i 通过函数 f_e 映射到低维空间的向量z_i

$$f_e: w_i \rightarrow z_i$$

- 使用神经网络f，基于以上低维空间向量进行单词条件概率计算

$$P(w_n|w_1 \cdots w_{n-1}) = f_{w_n}(z_1, \cdots, z_{n-1})$$

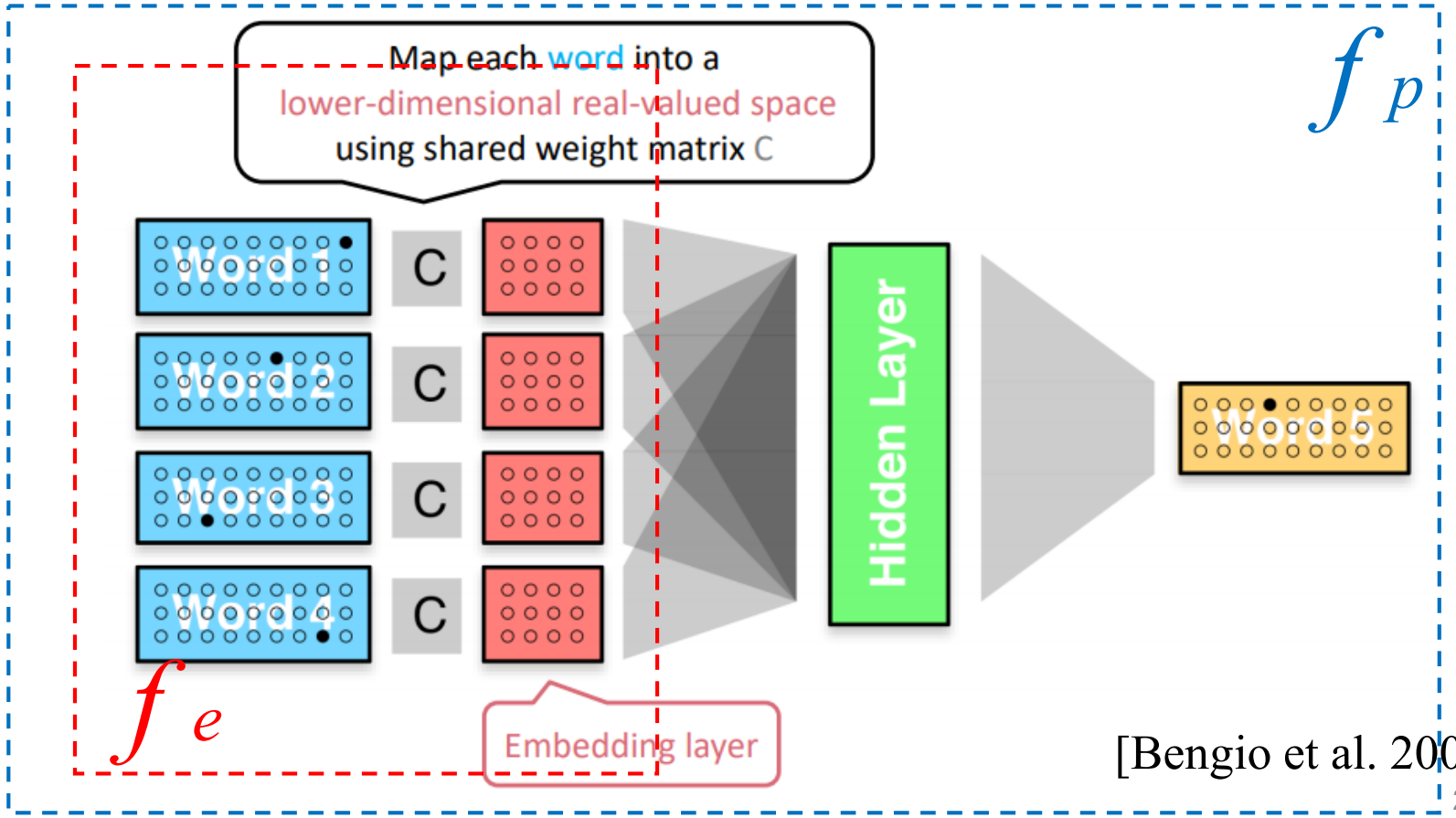
f_{w_n} 代表输出向量中与单词w_n对应的数值

- 基于神经网络语言模型的文本生成

$$\hat{w}_n = \arg\max_{w \in V} P(w|w_1 \cdots w_{n-1}) = \arg\max_{w \in V} f_w(z_1, \cdots, z_{n-1})$$

前馈神经网络语言模型

- 仍然使用与N-gram相同的马尔科夫假设
- f_e：将每个单词通过矩阵 C 映射到一个低维实值空间
- f：前馈神经网络



[Bengio et al. 2003]

输入层

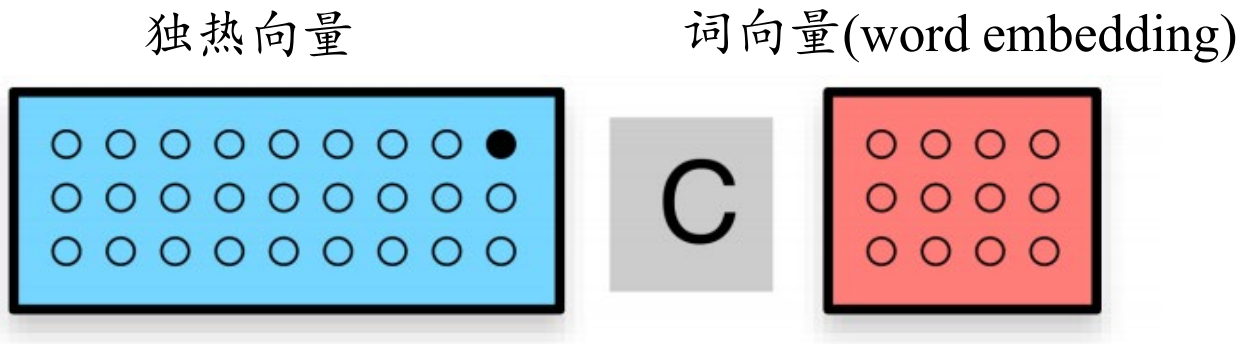
- 使用独热向量 (one-hot vector)表示单词

```
dog = [ 0, 0, 0, 0, 1, 0, 0, 0 …]
cat = [ 0, 0, 0, 0, 0, 0, 1, 0 …]
eat = [ 0, 1, 0, 0, 0, 0, 0, 0 …]
```

- 维度高，直接使用造成神经网络模型参数量大，学习困难

嵌入层

- 利用矩阵C实现
  - 将独热向量(高维、稀疏、离散)转换为词嵌入/词向量(低维、稠密、连续)

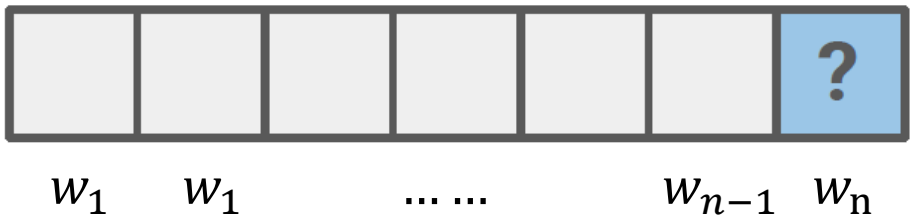


  - 矩阵C维度为$|V| \times d$
  - 每一行为一个d维向量，对应一个单词 → 词向量
  - 词向量也可以看做是神经语言模型的副产品

输出层

- 使用softmax激活函数
  - 计算w_n为词汇表中每个单词时的条件概率

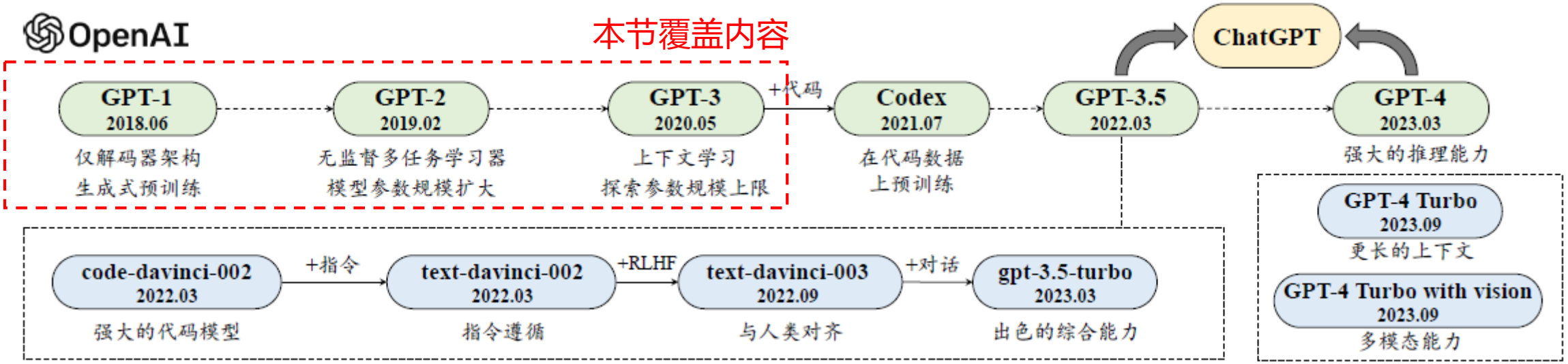
- **神经网络语言模型可以看做是一个执行分类任务的神经网络**
  - 已知历史单词，预测当前单词，类别数为|V|
  - 训练时最大化真实类别（即当前实际单词）的概率
  - **自回归生成的自监督学习任务**



目录

GPT系列模型概述

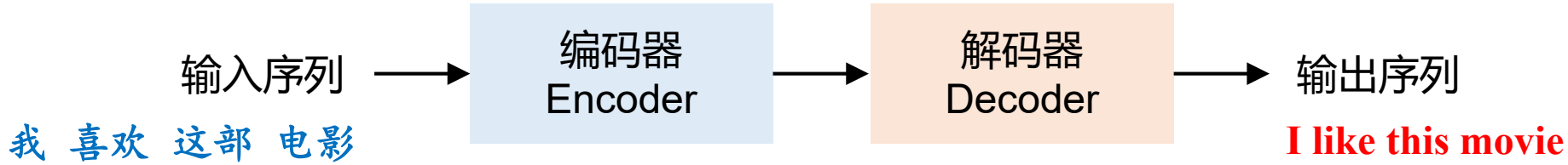
- GPT(Generative Pre-trained Transformer)
  - OpenAI自2018年开始推出的系列模型
  - 核心要素
    - 基于**Transformer**的**神经语言模型**，使用**自回归生成的自监督学习**任务
    - 扩展语言**模型规模**以及预训练**数据规模**



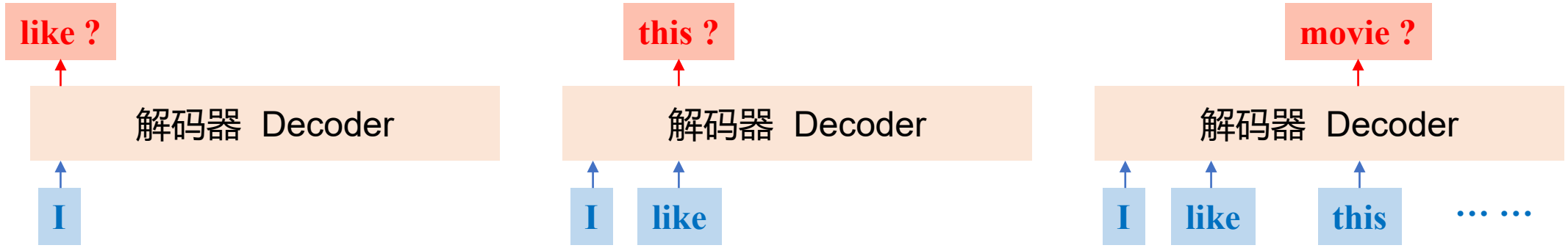
Wayne Xin Zhao et al. "A Survey of Large Language Models". In: *arXiv preprint arXiv: 2303.18223* (2023).

GPT-1

- 模型概述
  - 自监督学习任务与前馈神经网络语言模型相同：预测下一个词元(token)
  - 仅使用Transformer的解码器，被称为decoder-only架构
    - Transformer：基于自注意力的序列到序列模型



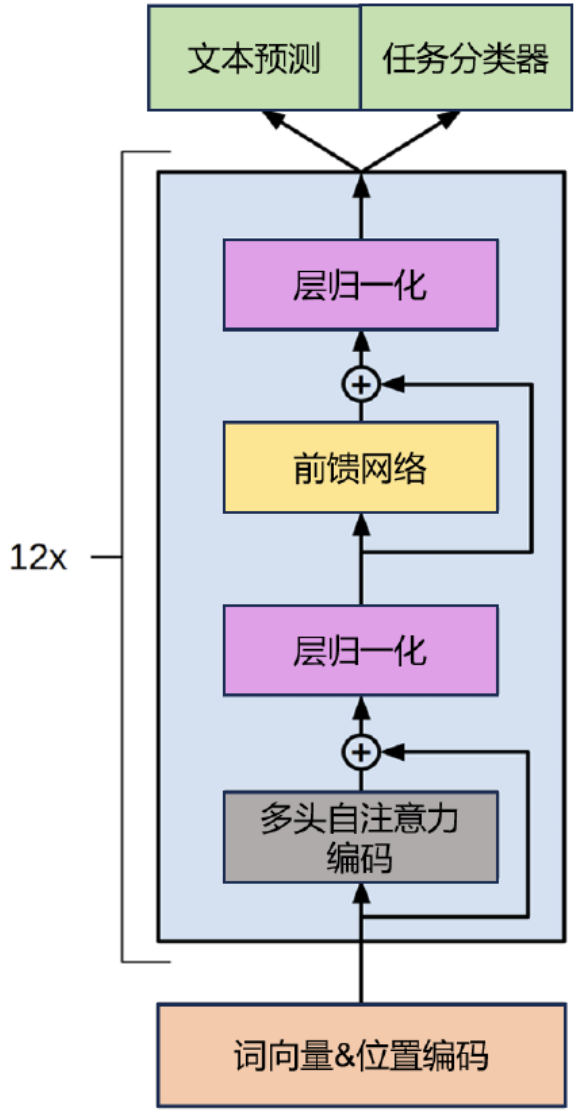
- GPT模型仅使用解码器，在序列内部进行下一个词元的预测



GPT-1

- 模型结构
  - 输入：基于BPE的词元切分；词向量叠加位置编码
  - 输出：softmax层输出概率
    - 文本预测(text prediction)：用于预训练/微调
    - 任务分类器(task classifier)：用于微调

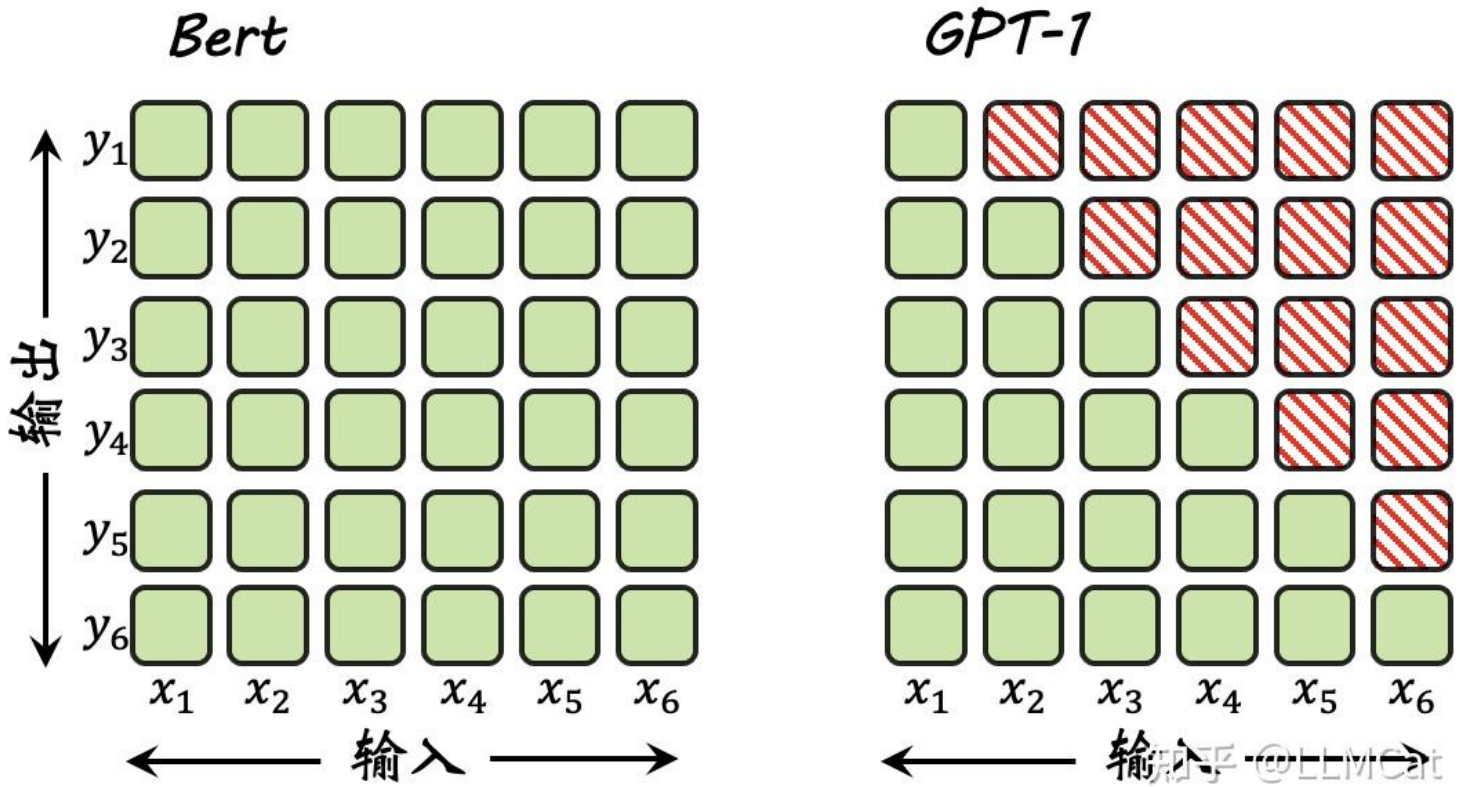
参数量	117M
Transformer层数	12
每层参数量	10M
隐层维度	768
词向量维度	512
词汇表大小	40K
预训练数据量	4.5GB



Radford et al. Improving Language Understanding by Generative Pre-Training. 2018

GPT-1

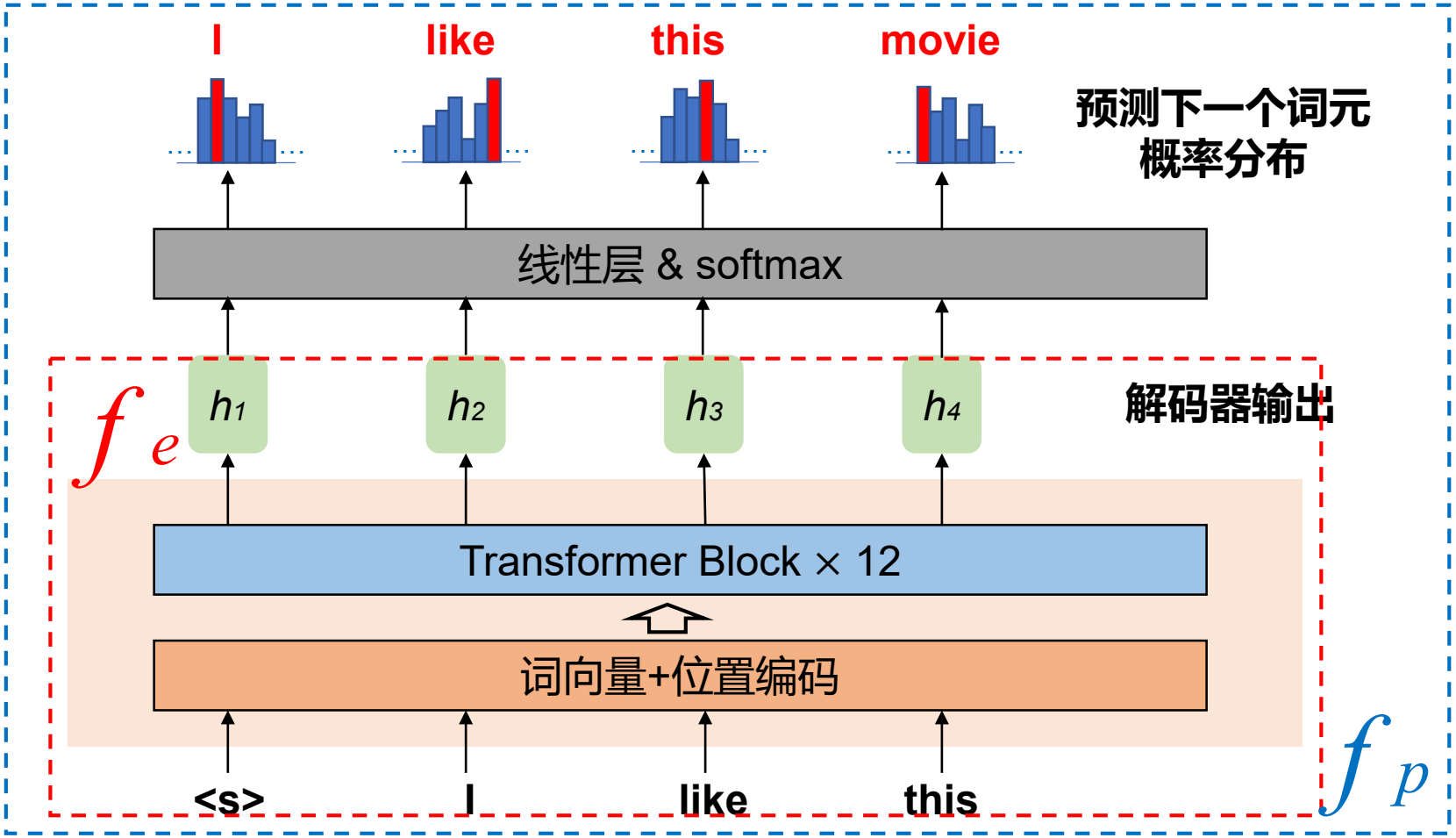
- 模型结构
  - 带因果掩码的单向自注意力机制



https://zhuanlan.zhihu.com/p/23083041150

GPT-1

- 应用范式：预训练+微调
  - 自监督预训练：最大化自监督训练样例中下一个词元的概率



GPT-1

- 应用范式：预训练+微调
  - 有监督微调：最大化有监督训练样例输出标签(label)的概率
    - 依据具体任务类别，将输入转换成词元构成的文本序列。例如：
    - **句子分类任务**：判断“I like this movie”这句话的情感极性是“正面”还是“负面”

<s> I like this movie <e>

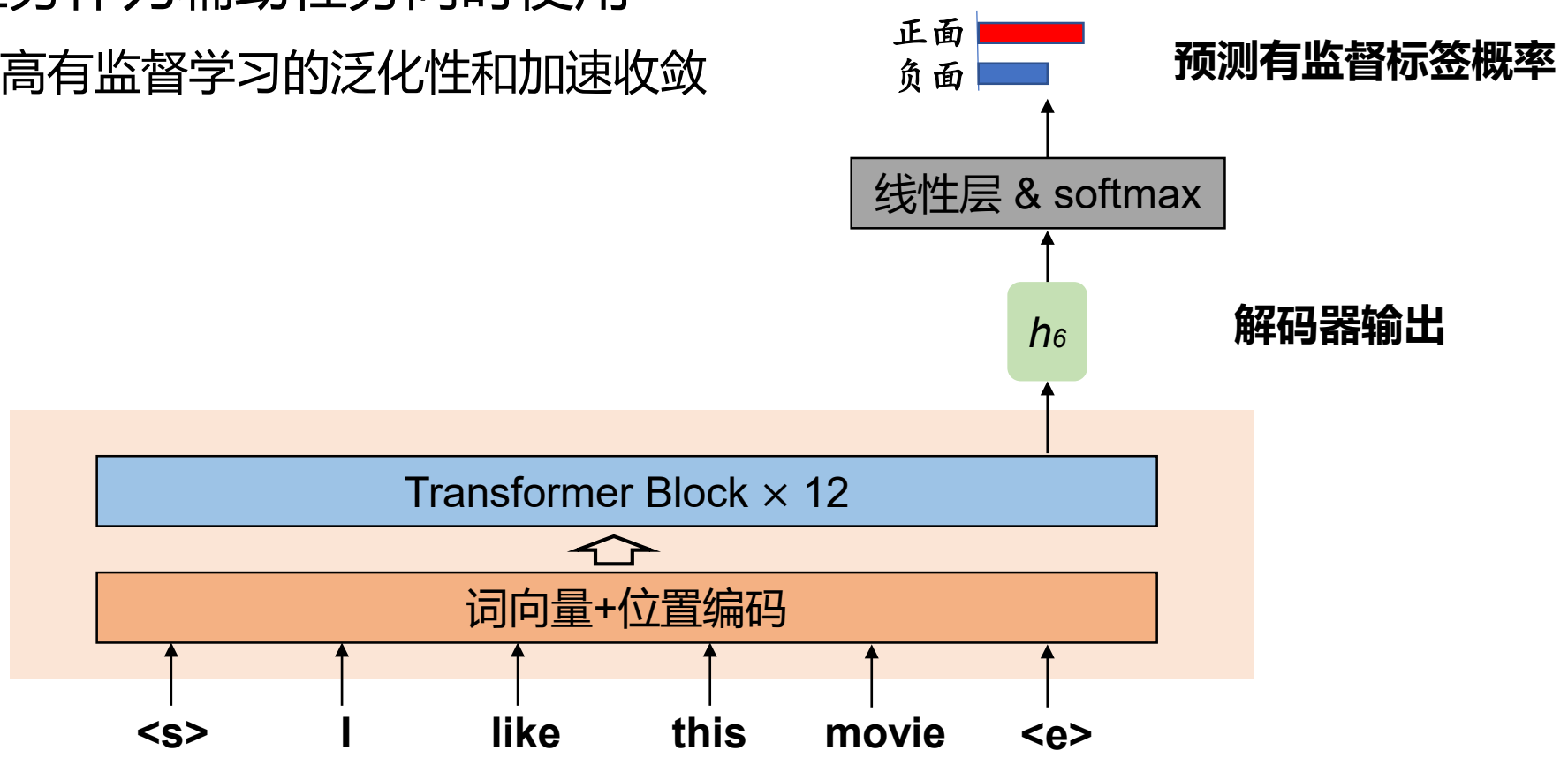
    - **文本蕴含任务：**判断“I like this movie”和“I have seen this movie”这两句话之间是“蕴含”、“矛盾”还是“中立”关系

<s> I like this movie $ I have seen this movie <e>

这里<s> <e> $ 分别为表示序列开始、结束、分隔的词元符号

GPT-1

- 应用范式：预训练+微调
  - 有监督微调：最大化有监督训练样例输出标签(label)的概率
  - 将预训练任务作为辅助任务同时使用
    - 有助于提高有监督学习的泛化性和加速收敛



GPT-1

- 对比BERT模型

	BERT	GPT-1
模型结构	Transformer编码器	Transformer解码器
词元化方法	WordPiece	BPE
自监督预训练	掩码语言模型+下一句预测	下一词元预测
有监督微调	目标任务	目标任务+下一词元预测
模型尺寸	110M(base) / 340M (large)	117M
词汇表大小	30K	40K

- GPT-1在公开评测数据集合上的性能没有优势，没有引起学术界的足够关注

GPT-2

- **主要动机**
  - 尝试去除针对特定任务所需要的微调环节，微调后模型只能执行特定任务
  - 希望建立一个能够**处理多个任务的通用语言模型** $p(output|input, task)$
  - 将**输入**、**输出**和**任务信息**都通过文本序列的形式进行描述，例如

> 机器翻译: (*"Translate to French"*, English text, French text)
>
> 阅读理解: ("*Answer the question*", document, question, answer)

  - 后续任务的求解过程就可以看作是任务方案（或答案）的文本生成问题

Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.

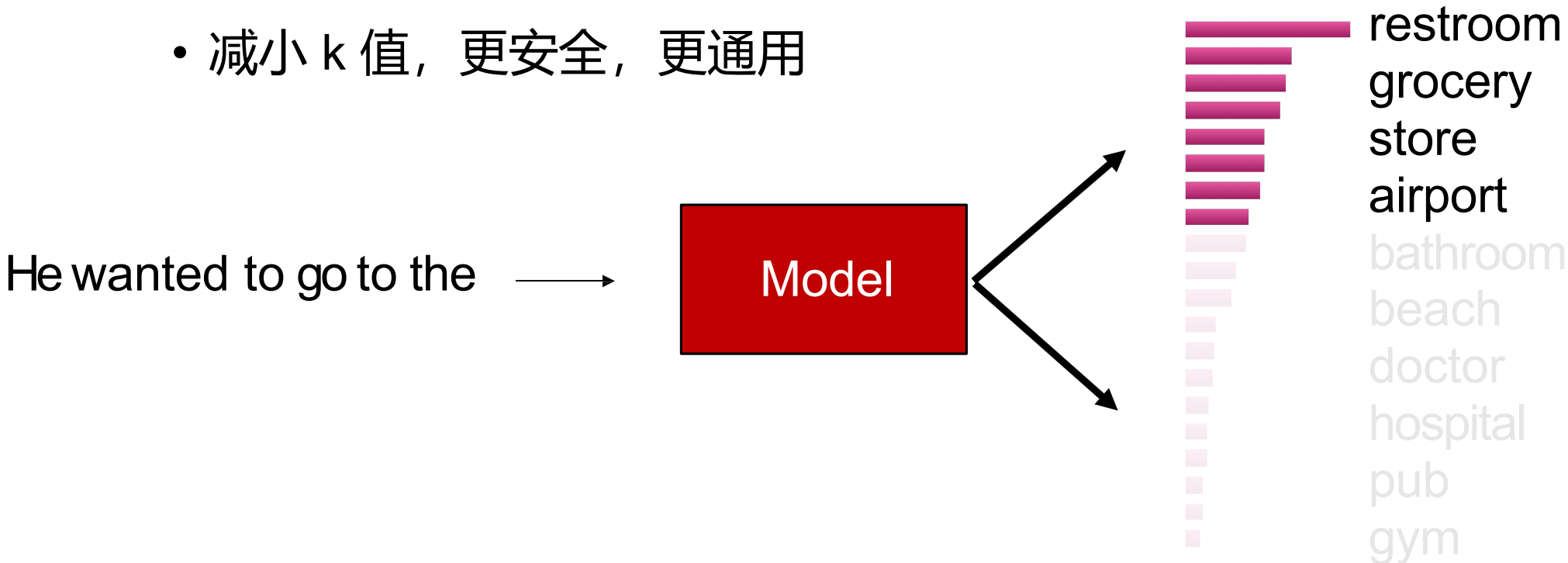
GPT-2

- 使用与GPT-1相同的模型结构
- 显著增加模型参数规模与训练数据规模

	GPT-1	GPT-2 117M	GPT-2 345M	GPT-2 762M	GPT-2 (full)
参数量	117M	117M	345M	762M	1.5B
Transformer层数	12	12	24	36	48
每层参数量	10M	10M	14M	21M	31M
隐层维度	768	768	1024	1280	1600
词向量维度	512	1024			
词汇表大小	40K	50K			
预训练数据量	4.5GB	40GB			

GPT-2

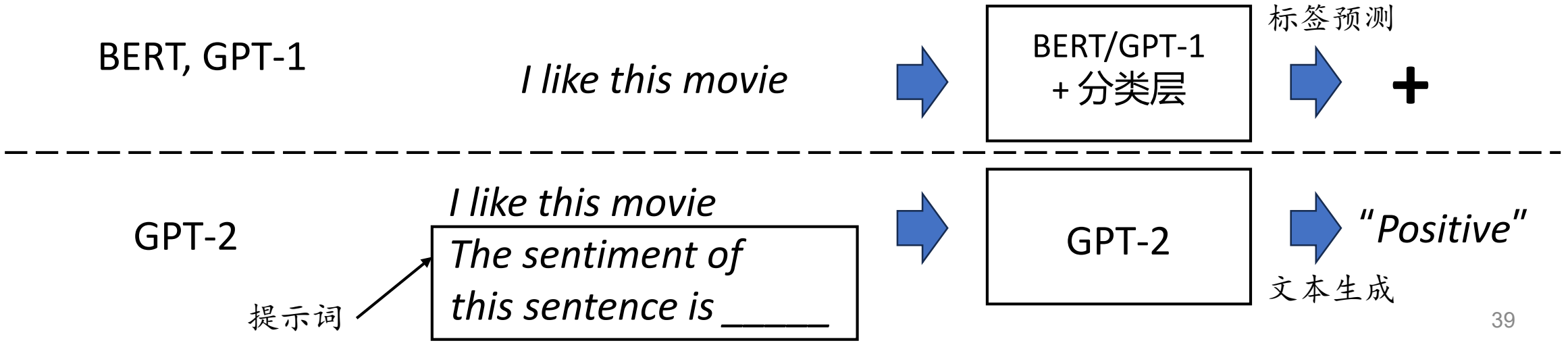
- 为进一步提升生成文本的多样性与合理性，采用top-k采样方法生成输出
  - 对每个词元位置，从输出概率分布中概率最高的前k个单词中采样生成
    - 增大 k 值，更多样化，更多风险
    - 减小 k 值，更安全，更通用



(Fan et al., ACL 2018; Holtzman et al., ACL 2018)

GPT-2

- 应用范式：**零样本(zero-shot)学习**
  - 模型在没有任务相关的训练样例的情况下有能力执行多种任务，不需要经过任务相关的有监督参数更新
- GPT-2涌现出零样本学习能力
  - 输入文本和任务指令(instruction)都以文本形式输入到模型中
  - 这些任务指令被称为 “提示词(prompt)”；模型以文本形式生成相应的输出



GPT-2

- GPT-2在多个自然语言处理下游任务上显著提升零样本学习性能
- OpenAI团队的解释
  - 特定任务的有监督学习目标与自监督学习目标在本质上是相同的（预测下一个词元），主要区别就在于它们只是在全部训练数据的子集上进行
  - 因此对于特定下游任务而言，优化自监督的全局学习目标本质上也是在优化有监督的任务学习目标
- 通俗理解
  - 语言模型将每个自然语言处理任务视为基于世界文本子集的下一个词预测问题
  - 如果自监督语言建模经过训练后具有足够的能力复原全部世界文本，那么本质上它就能够解决各种任务

GPT-3

- **主要动机**
  - 在GPT-2基础上，显著提升模型参数规模与训练数据规模
  - 提出了“上下文学习”(in-context learning)的新范式
    - 与GPT-2相同，通过语言模型的文本生成实现多种下游任务
    - 与GPT-2不同，支持单样本(one-shot)与少样本(few-shot)学习
    - 模型理解自然语言文本形式描述的新任务样本，无需针对新任务进行微调

Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

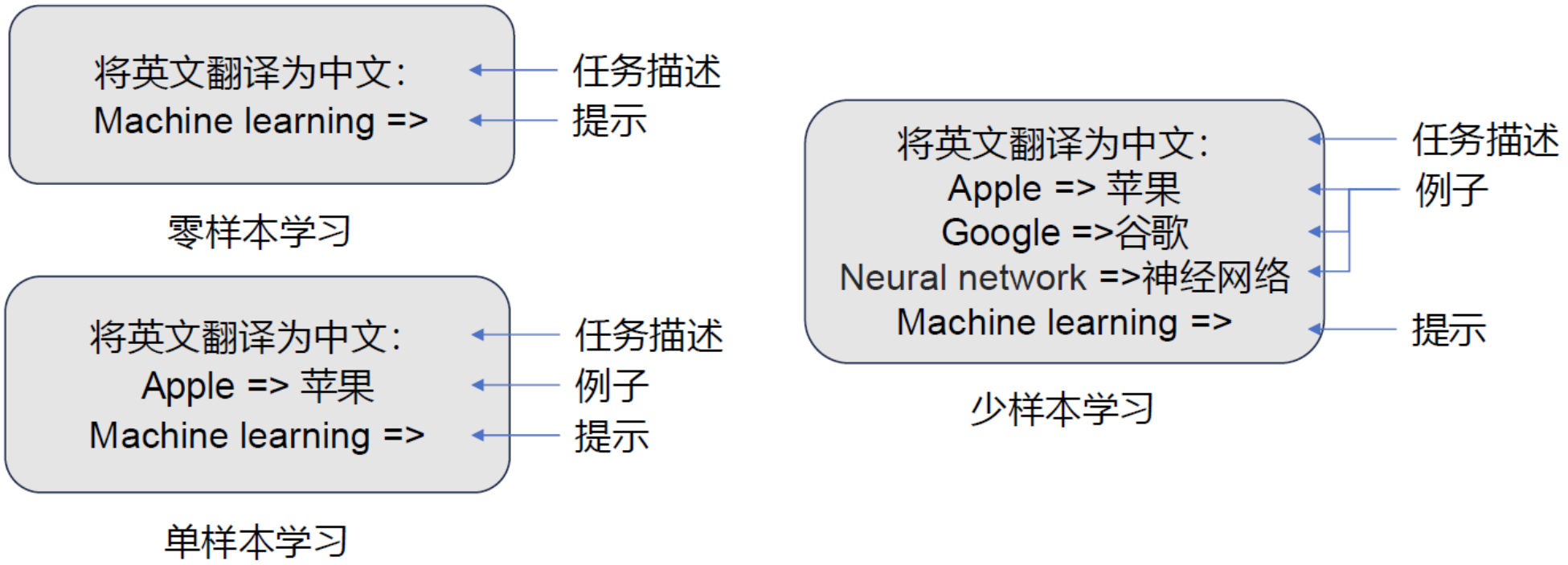
GPT-3

- 使用与GPT-1/2相同的模型结构
- 进一步显著增加模型参数规模与训练数据规模

	GPT-1	GPT-2	GPT-3
参数量	117M	1.5B	175B
Transformer层数	12	48	96
每层参数量	10M	31M	1.8B
隐层维度	768	1600	12288
词向量维度	512	1024	2048
词汇表大小	40K	50K	50K
预训练数据量	4.5GB	40GB	>600GB

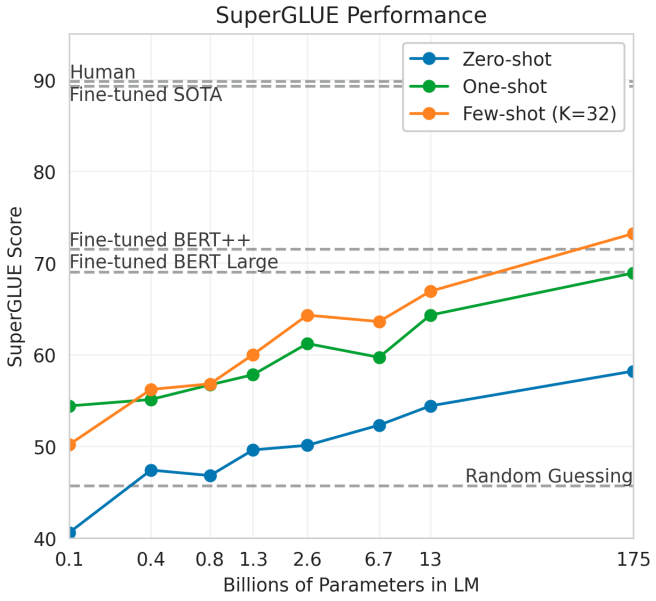
GPT-3

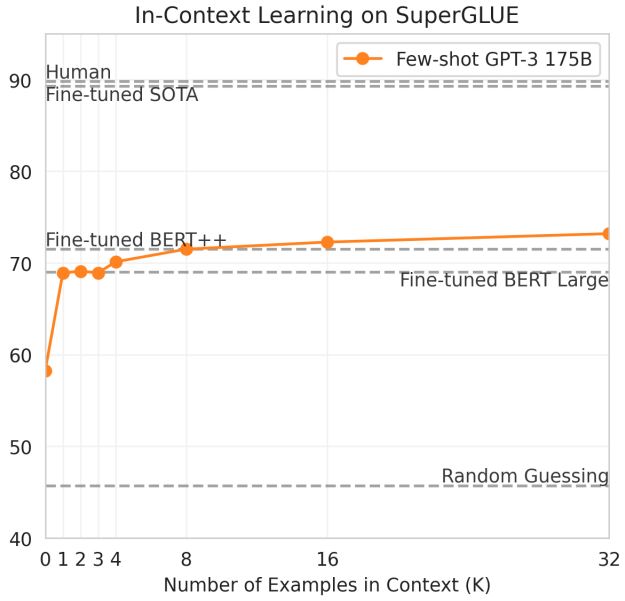
- 应用范式：上下文学习 (in-context learning)
  - 提示学习(prompt learning)的一种类型
  - 使用由任务描述和（或）示例所组成的自然语言文本作为提示(prompt)
  - 模型无需显式的梯度更新即可识别和执行新的任务

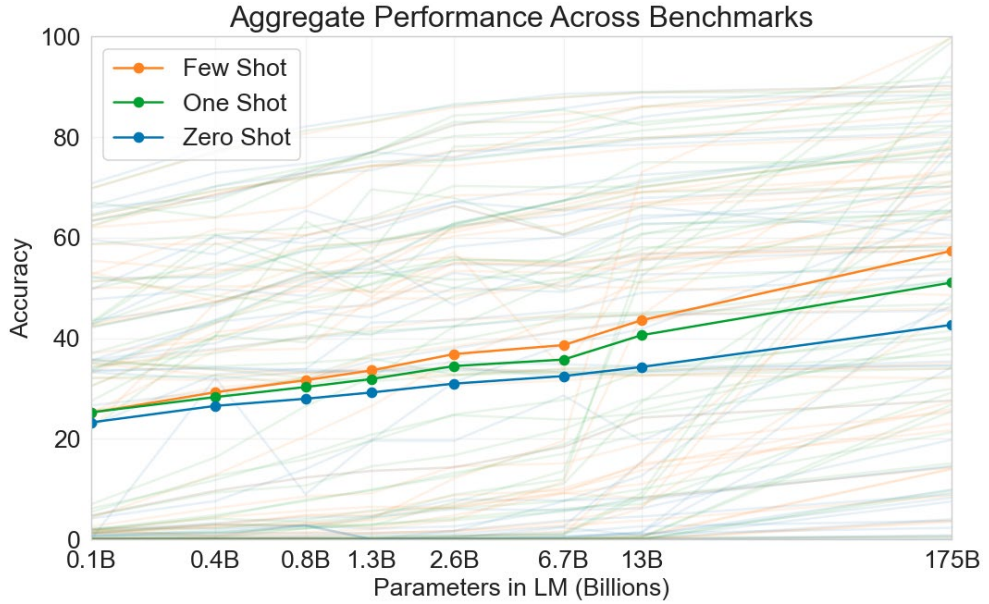


GPT-3

- 模型性能
  - SuperGLUE测试集上，GPT-3使用32样本的少样本训练性能超过BERT-large，后者使用数十万任务相关样本进行微调
  - 在42个不同任务上的测试结果体现了零样本/单样本/少样本学习性能模型规模增大持续提升







GPT-3

- GPT-3的重要意义
  - 可以被看作从预训练语言模型到大语言模型演进过程中的一个重要里程碑
  - 验证了将神经网络扩展到超大规模可以带来大幅的模型性能提升
  - 建立了以提示学习(prompt learning)方法为基础技术路线的任务求解范式

本节小结

<table>
<tr><th></th><th>N元语法语言模型</th><th>前馈神经网络
语言模型</th><th>GPT-1</th><th>GPT-2</th><th>GPT-3</th></tr>
<tr><td>自监督
学习目标</td><td colspan="5">自回归生成（下一词元预测）$P(w_i|w_1 \cdots w_{i-1})$</td></tr>
<tr><td>模型结构</td><td>马尔科夫假设+
直接存储条件概率</td><td>马尔科夫假设+
前馈神经网络</td><td colspan="3">Transformer解码器</td></tr>
<tr><td rowspan="2">应用范式</td><td rowspan="2">句子概率评分</td><td rowspan="2">句子概率评分
词向量（副产品）</td><td rowspan="2">预训练+微调</td><td colspan="2">以文本形式生成输出</td></tr>
<tr><td>零样本学习</td><td>上下文学习</td></tr>
</table>

课后思考

- 任选一本中文书籍，记录书名以及其在版权页上标注的字数；假设中文常用汉字有3500个，以单字作为词元(token)单位，以这本中文书籍为训练语料库建立1元/2元/3元语法模型，计算每个1元/2元/3元语法平均会有多少个计数。