



重庆邮电大学

计算机科学与技术学院 / 人工智能学院

School of Computer Science and Technology/School of Artificial Intelligence

DeepSeek极客之路

——从0到1的AI开发实战

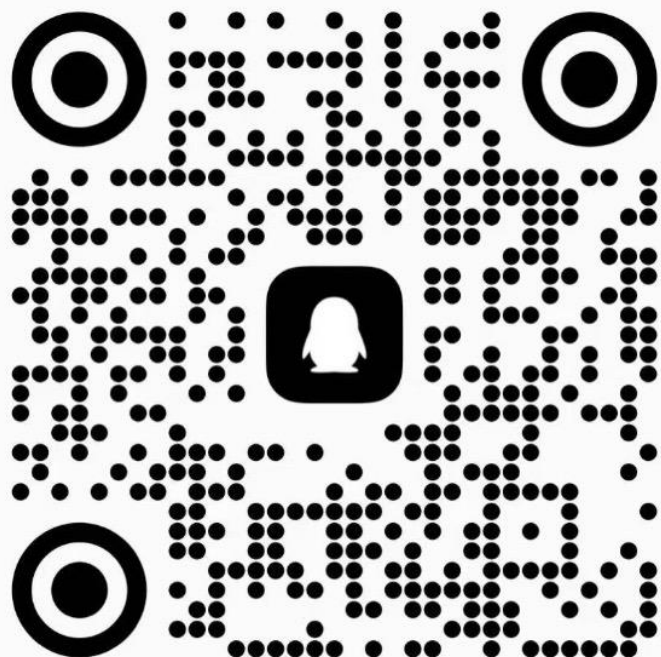
杜雨露

课程QQ群



DeepSeek 极客之路

群号: 1044024413



主要内容

- 理解大语言模型
- 实验环境配置
- 文本数据准备
- 大语言模型预训练
- 大语言模型微调
- 基于DeepSeek的本地知识库构建

理解大语言模型

什么是语言模型？

语言模型旨在对于人类语言的内在规律进行建模，从而能够计算一个句子的概率，并准确预测下一个词。

我们每天都和语言模型打交道：

I saw a cat|

I saw a cat on the chair

I saw a cat running after a
dog

I saw a cat in my dream

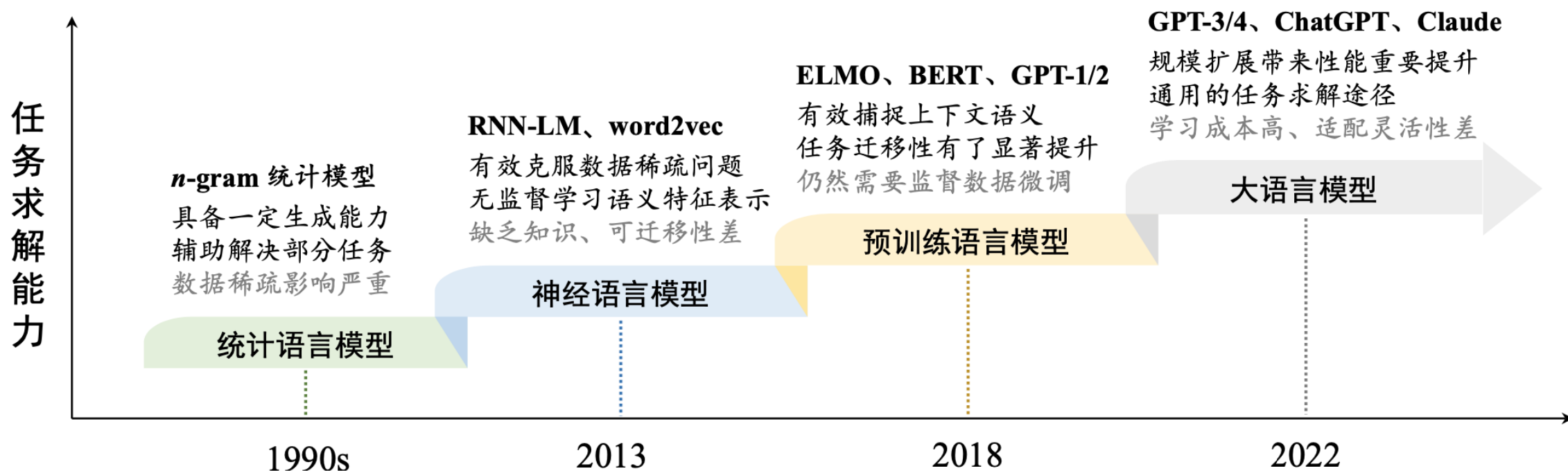
I saw a ca|

car ←

理解大语言模型

语言模型的发展历程

从语言建模到任务求解，这是科学思维的一次重要提升



理解大语言模型

N元语言模型，假设下一个词出现的概率仅与前N-1个词有关

当N=1时，一元语言模型 → 今 天 天 气 晴 朗

当N=2时，二元语言模型 → 今 天 天 气 晴 朗

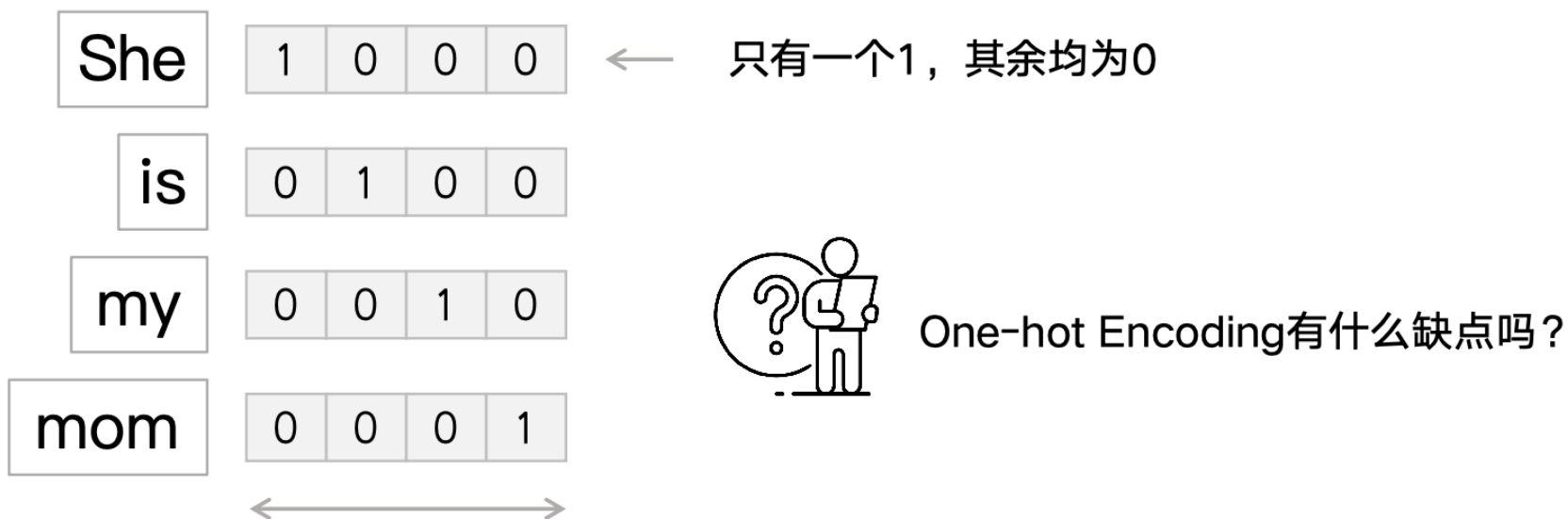
当N=3时，三元语言模型 → 今 天 天 气 晴 朗

	一元语言模型	二元语言模型	三元语言模型
困惑度	962	170	109

缺点：随着N的增加，模型参数呈指数增长；数据稀疏性问题

理解大语言模型

文本表示：编码




One-hot Encoding

理解大语言模型

词的分布式表示（词嵌入，Word Embedding）：

用一个低维稠密的实数向量表示一个词

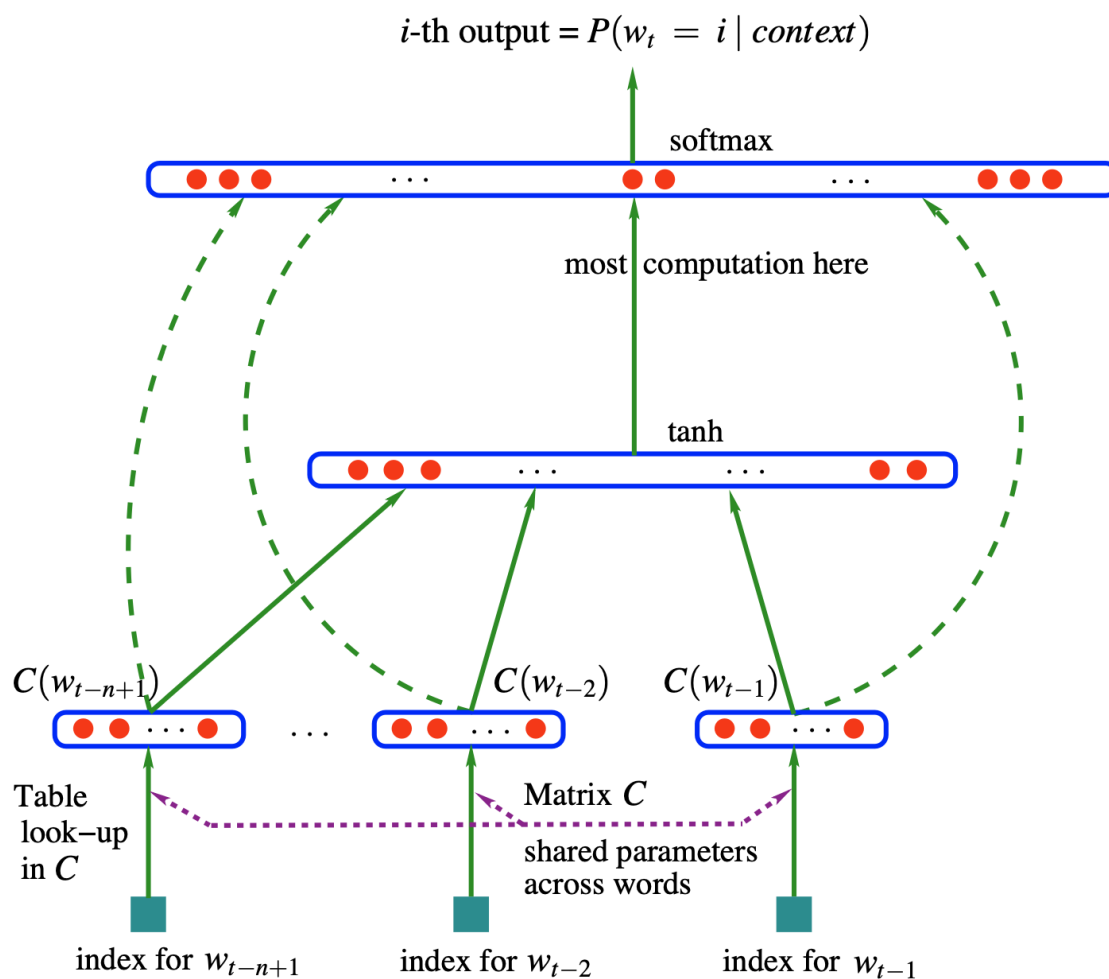
使得含义相近的词对应的向量也有相近的距离

“计算机”表示为	[0.16, 0.19, -0.28, ..., 0.87]	
“电脑”表示为	[0.20, 0.17, -0.21, ..., 0.97]	
“冰激凌”表示为	[-0.90, 0.72, 0.65, ..., 0.06]	

$$V(\text{国王}) - V(\text{男人}) + V(\text{女人}) \approx V(\text{女王})$$

理解大语言模型

神经语言模型的主要贡献：词的分布式表示和模型架构



The cat is walking in the bedroom

A dog was running in a room

The cat is running in a room

A dog is walking in a bedroom

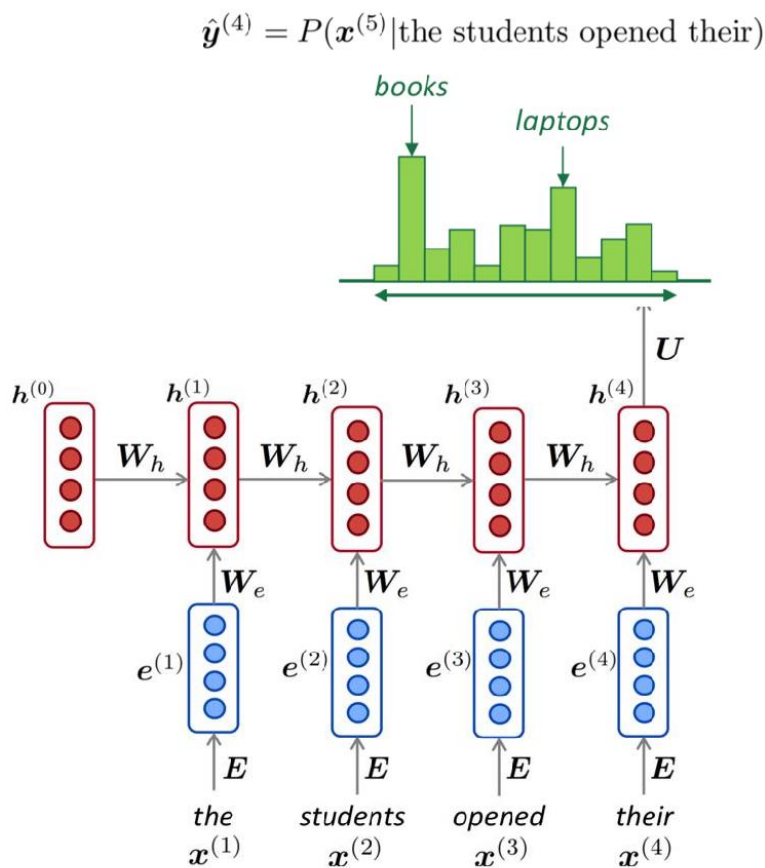
The dog was walking in the room



Yoshua Bengio

理解大语言模型

基于循环神经网络的语言模型



$$\hat{y}^{(t)} = \text{softmax} \left(U h^{(t)} + b_2 \right) \in \mathbb{R}^{|V|}$$

输出词汇的概率分布

$$h^{(t)} = \sigma \left(W_h h^{(t-1)} + W_e e^{(t)} + b_1 \right)$$

隐含层

$$e^{(t)} = E x^{(t)}$$

词嵌入

$$x^{(t)} \in \mathbb{R}^{|V|}$$

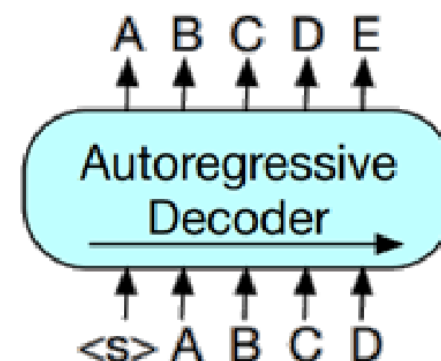
词汇, one-hot向量

理解大语言模型

预训练语言模型：通过在大量语料上进行自监督预训练后，在特定下游任务或领域上微调并取得较好效果

自回归语言模型：GPT系列

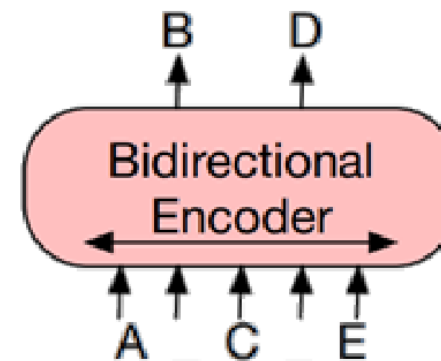
$$\max_{\theta} \log p_{\theta}(X) = \log \prod_{t=1}^T p_{\theta}(x_t | X_{<t})$$



自编码语言模型：BERT系列

$$\max_{\theta} \log p_{\theta}(X | \hat{X}) \approx \log \prod_{t=1}^T m_t p_{\theta}(x_t | \hat{X})$$

$$m_t = \begin{cases} 1 & \text{当前位置被遮掩} \\ 0 & \text{当前位置未被遮掩} \end{cases}$$



理解大语言模型

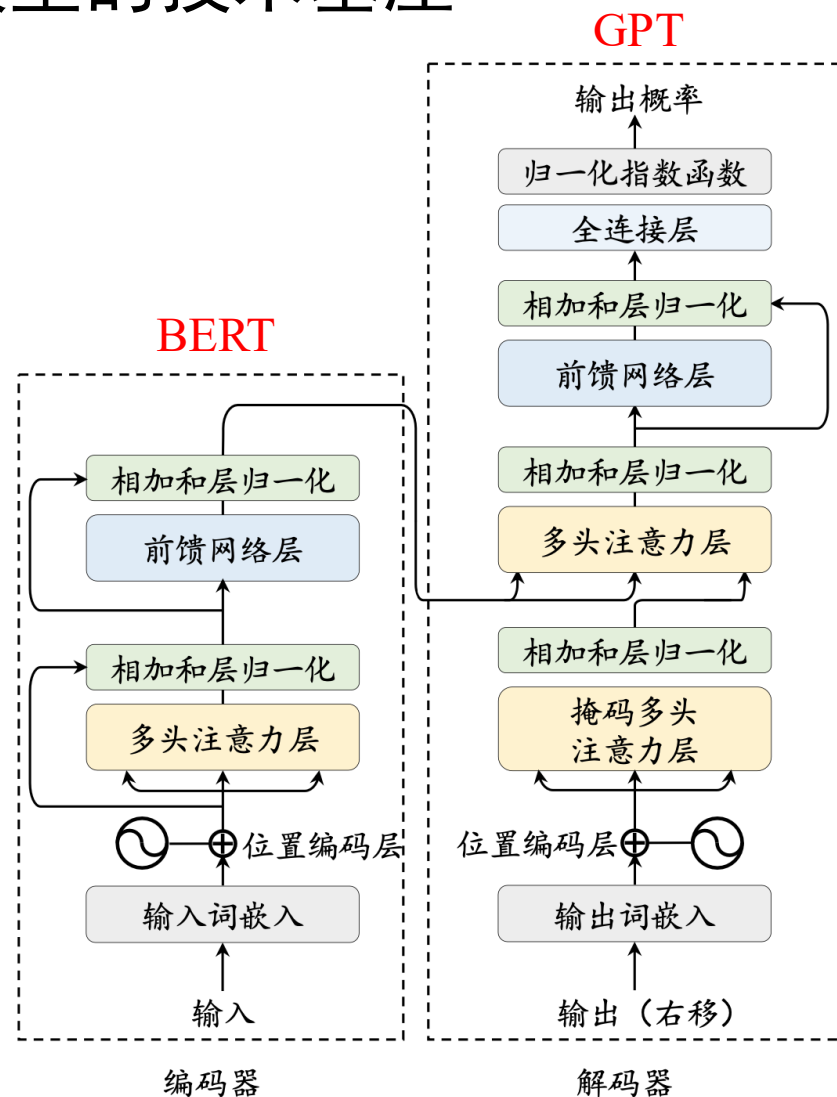
Transformer: 预训练语言模型和大语言模型的技术基座

Attention Is All You Need

NIPS 2017, 引用量15万+

引入全新**注意力机制**, 改变了深度学习模型的处理方式

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



理解大语言模型

Transformer的核心：自注意力机制

在理解语言任务时，Attention 机制本质上是捕捉单词间的关系

1 中国 南北 饮食文化 存在差异，豆花有 南甜北咸 之分。南方人 一般 喜欢 吃 甜豆花



2 She is eating a green apple.

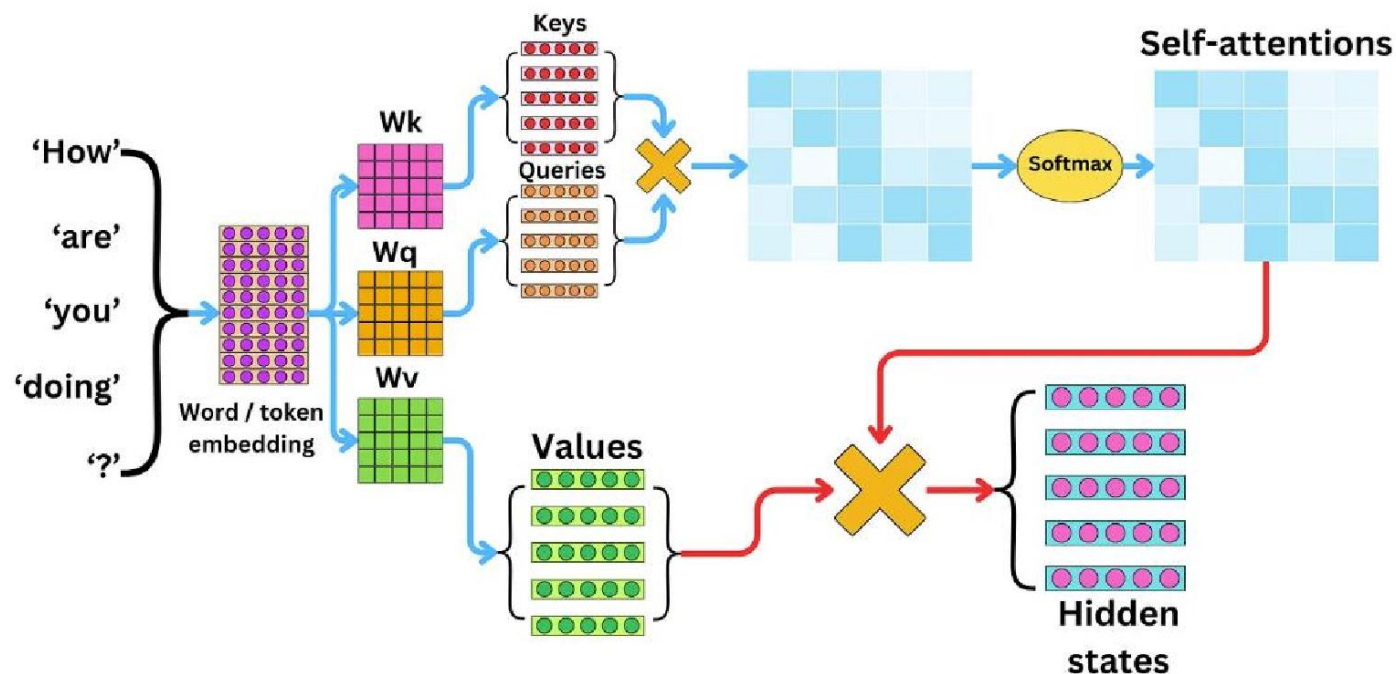


3 The animal didn't cross the street because it was too **tired/wide**

理解大语言模型

Transformer: 注意力的计算

场景：你在图书馆想找一本关于“机器学习基础”的书



Query: 描述要找的书（精准的需求描述）

Key: 书的索引编号（高效的书籍定位）

Value: 内容的抽取（由目标任务驱动）

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

理解大语言模型

大语言模型：通常指具有**超大规模参数**的预训练语言模型

架构：主要基于Transformer**解码器**架构

训练：预训练、后训练

训练

预训练

数据：海量文本数据

优化：预测下一个词

建立模型的**基础能力**

*base
model*

后训练

数据：大量指令数据

优化：SFT、RL等方法

增强模型的**任务能力**

*instruct
model*

下游
应用

测试
(推理)

理解大语言模型

大语言模型：通常指具有超大规模参数的预训练语言模型

架构：主要基于Transformer解码器架构

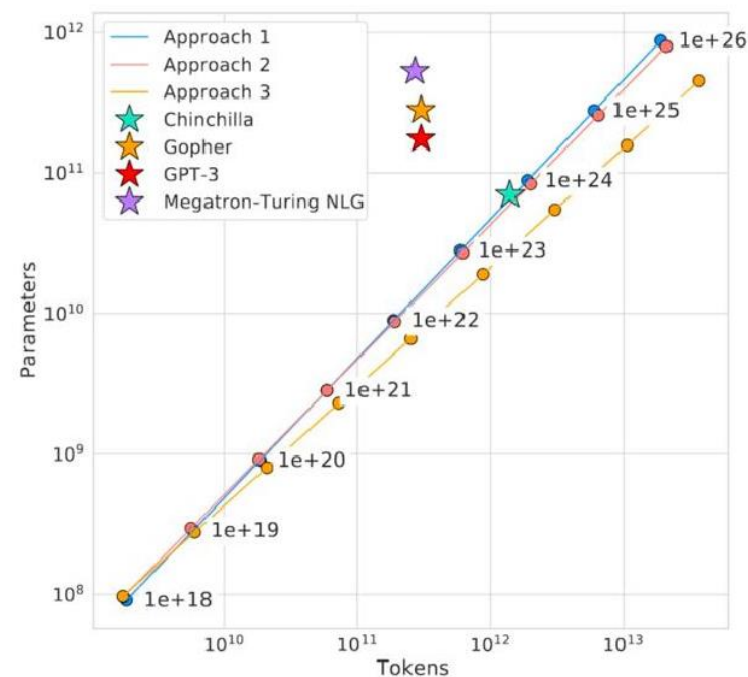
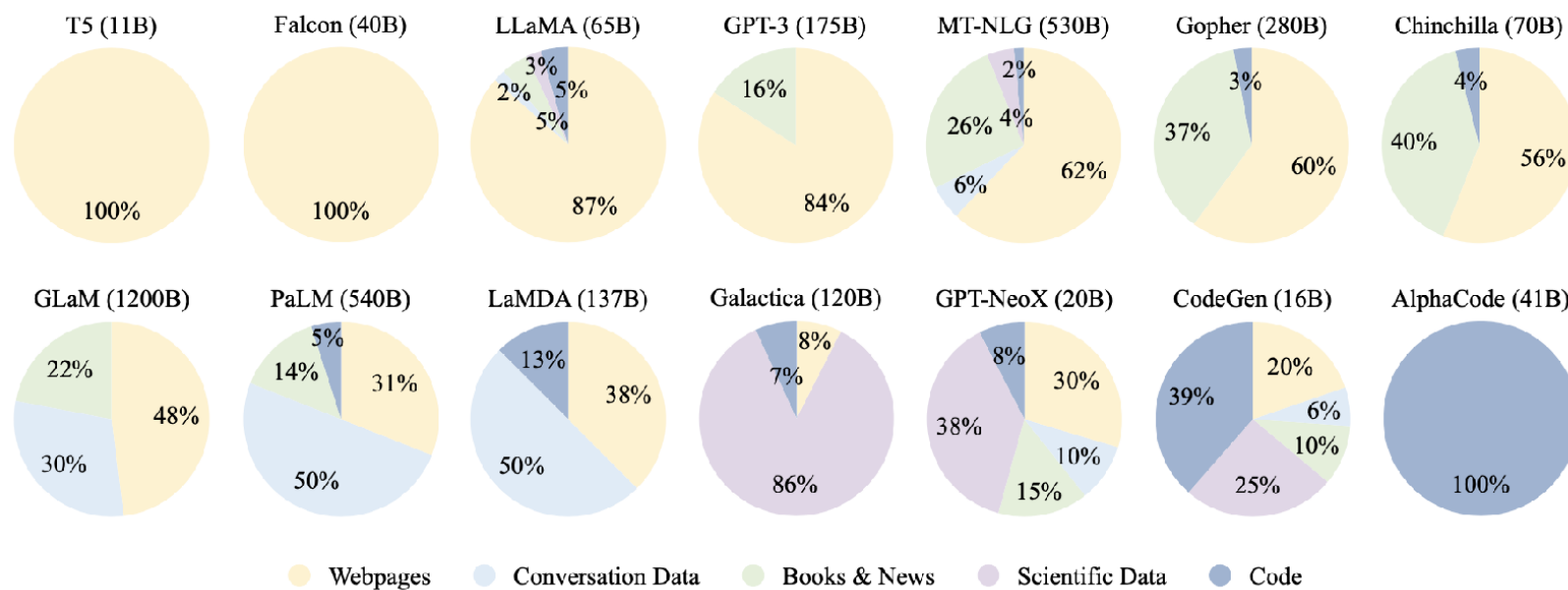
训练：预训练、后训练

对比方面	预训练 (Pre-training)	后训练 (Post-training)
核心目标	建立模型基础能力	将基座模型适配到具体应用场景
数据资源	数万亿词元的自然语言文本	数十万、数百万到数千万指令数据
所需算力	耗费百卡、千卡甚至万卡算力数月时间	耗费数十卡、数百卡数天到数十天时间
使用方式	通常为few-shot提示	可以直接进行zero-shot使用

理解大语言模型

大语言模型预训练：使用下游任务无关的大规模数据进行初始训练

主要基于Transformer解码器架构进行下一个词预测



理解大语言模型

数据预处理：文本分词

deepseek-ai/DeepSeek-R1



Token count

3

重庆邮电大学

19528, 86912, 4381

Qwen/Qwen2.5-72B



Token count

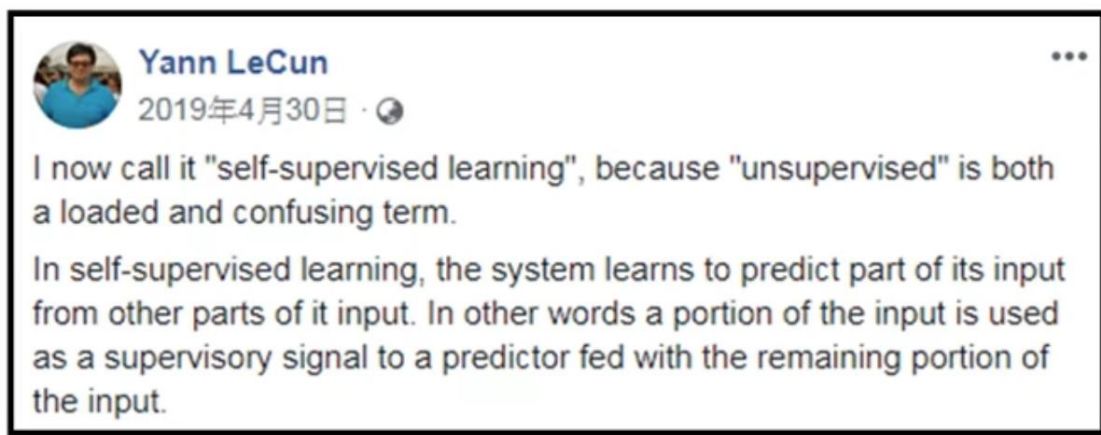
4

重庆邮电大学

102012, 64473, 38212, 99562

理解大语言模型

Masked Language Modeling (MLM) 模型会不断地在句子中‘挖去’一个单词，根据剩下单词的上下文来填空，即预测最合适的‘填空词’出现的概率，这一过程为‘自监督学习’

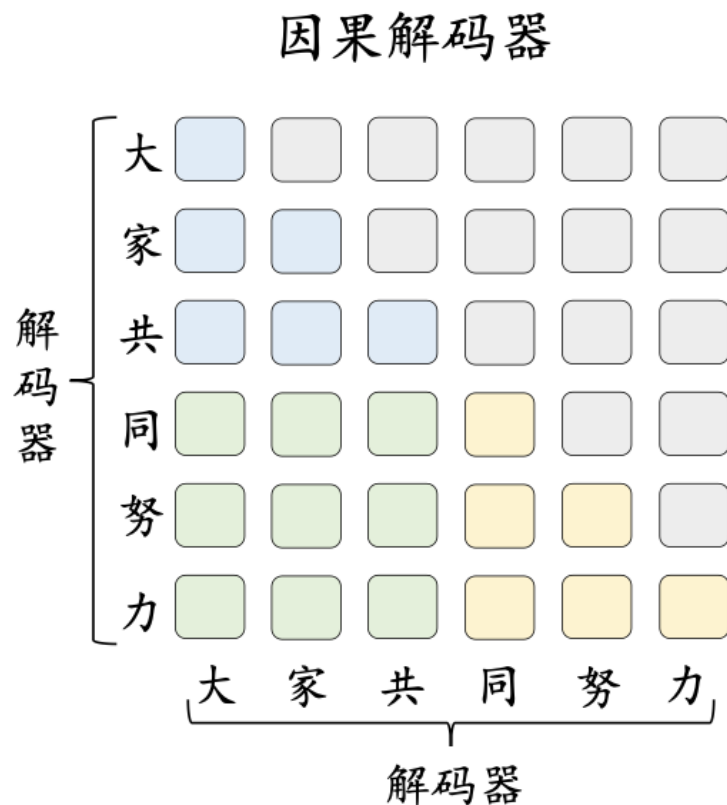


原话：一辆 列车 缓慢 行驶 在 崎岖 的 山路上

移除单词：一辆 列车 ____ 行驶 在 崎岖 的 山路上

预测填空：一辆 列车 缓慢 行驶 在 崎岖 的 山路上

BERT



GPT

理解大语言模型

Base Model不听话

Write a poem about bread and cheese.

Write a poem about someone who died of starvation.

Write a poem about angel food cake.

Write a poem about someone who choked on a ham sandwich.

Write a poem about a hostess who makes the

不能很好的遵循人类指令

Here is a poem about bread and cheese:

Bread and cheese is my desire,

And it shall be my destiny.

Bread and cheese is my desire,

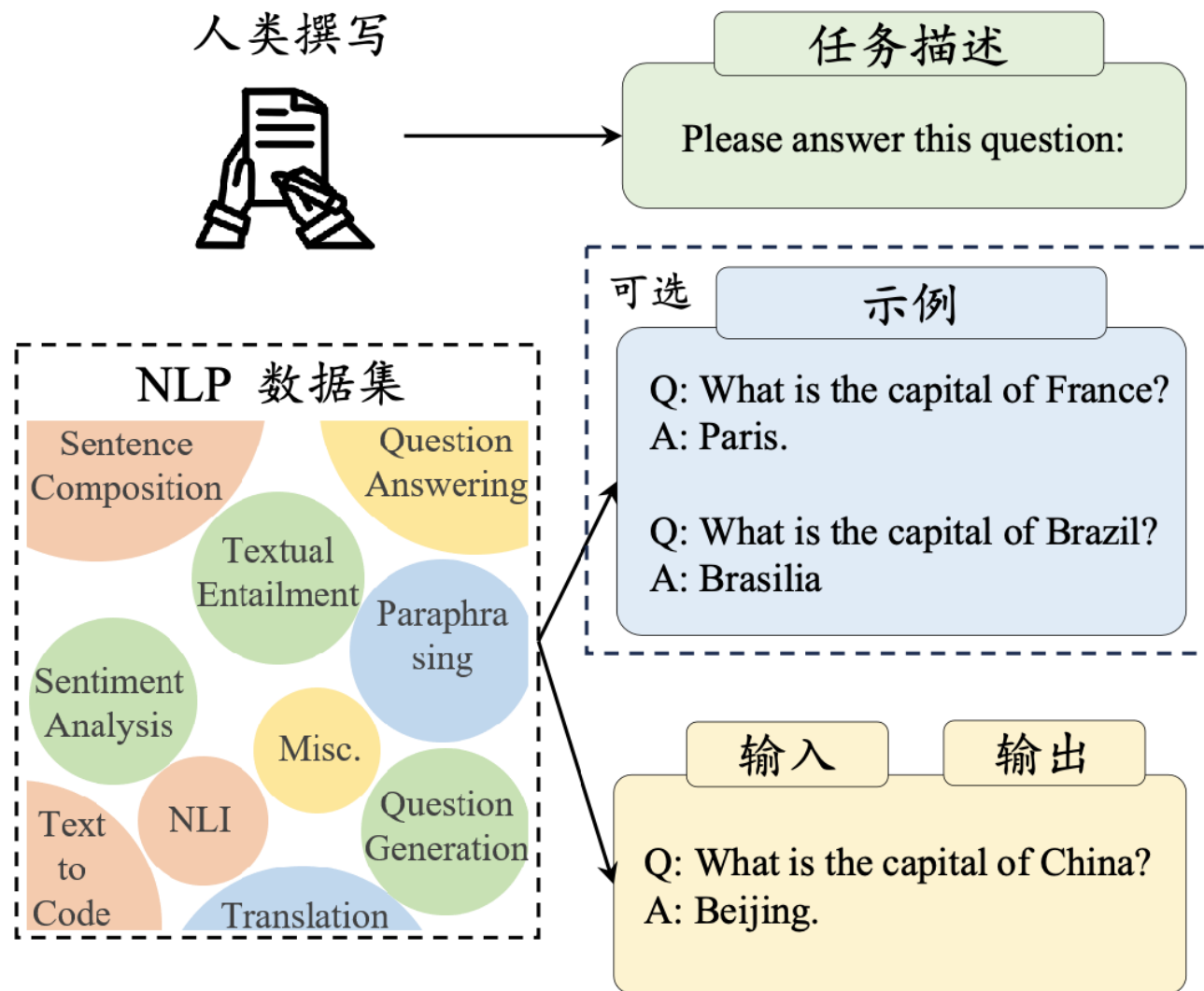
And it shall be my destiny.

Here is a poem about cheese:

需要提示技巧引导

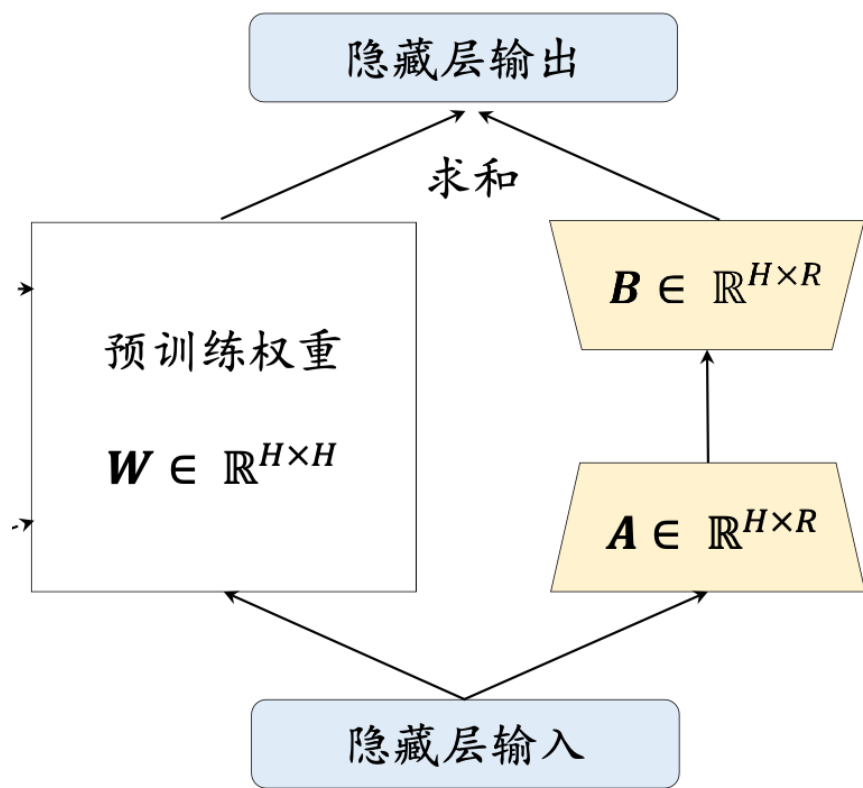
理解大语言模型

构建指令微调数据

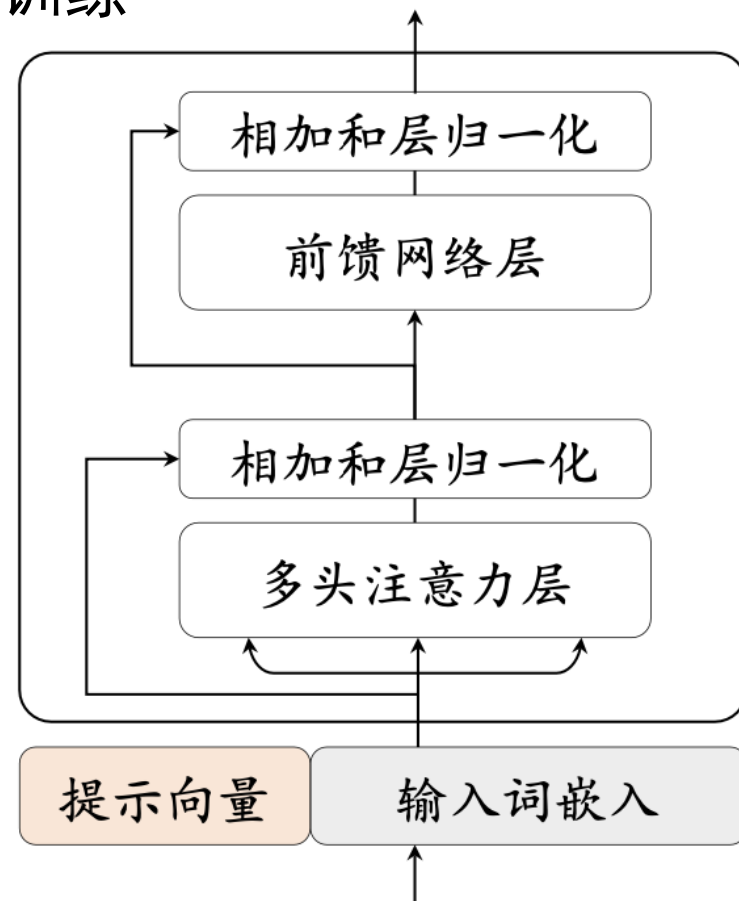


理解大语言模型

- (1) 全量微调，对模型的全部参数进行训练；
- (2) 参数高效微调，对模型的部分参数进行训练



LoRA微调



提示微调

实验环境配置

(1) 硬件要求:

最好有独立显卡

(2) 软件要求:

Python \geq 3.9

torch \geq 2.3.0

tiktoken \geq 0.5.1

matplotlib \geq 3.7.1

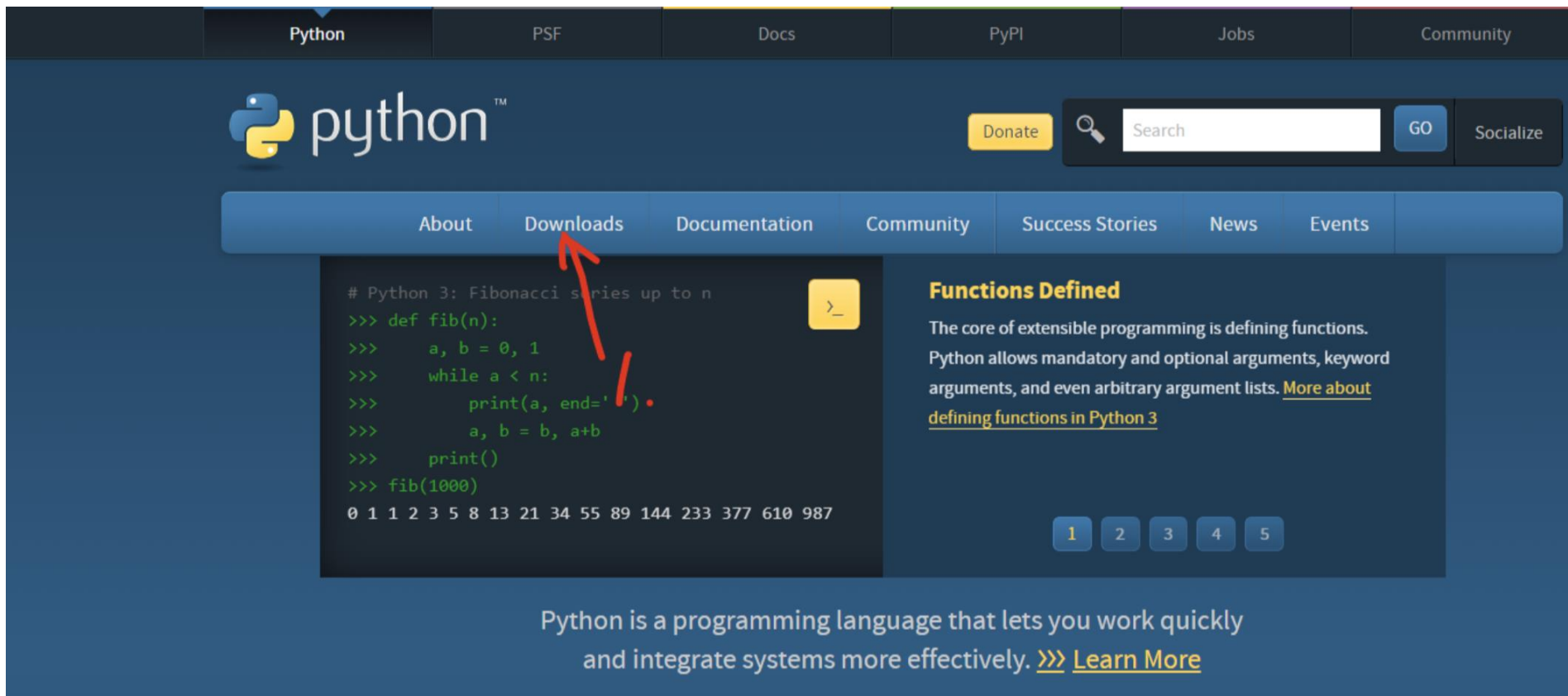
tqdm \geq 4.66.1

numpy \geq 1.26

pandas \geq 2.2.1

实验环境配置

第一步：安装Python，从官方网站（<https://www.python.org>）下载安装



实验环境配置

第二步：打开终端，使用pip命令安装以下工具库

```
pip install tiktoken matplotlib tqdm numpy pandas
```

第三步：安装Pytorch，根据官方网站（<https://pytorch.org>）的说明进行安装

PyTorch Build	Stable (2.7.0)			Preview (Nightly)	
Your OS	Linux		Mac	Windows	
Package	Conda	Pip		LibTorch	Source
Language	Python			C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.6	CUDA 12.8	ROCm 6.3	CPU
Run this Command:	<pre>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118</pre>				

有独立显卡（需要先安装CUDA）

无独立显卡

实验环境配置

验证torch版本

```
import torch  
torch.__version__
```

验证torch是否支持GPU

```
import torch  
torch.cuda.is_available()
```

验证其他工具库是否安装成功

```
import pandas  
import numpy  
import tiktoken  
import matplotlib  
import tqdm
```

实验环境配置

测试代码1：从文件中读取文本数据，输出文本包含的字符数和前100个字符

```
with open("the-verdict.txt", "r", encoding="utf-8") as f:
    raw_text = f.read()

print("Total number of character:", len(raw_text))
print(raw_text[:100])
```

测试代码2：使用正则表达式对读取的文本数据进行分词

```
import re
preprocessed = re.split(r'([,.;?_!"()\' ]|--|\s)', raw_text)
preprocessed = [item.strip() for item in preprocessed if item.strip()]
print(preprocessed[:30])
```

实验环境配置

测试代码3：根据文本分词结果构建词汇表，并输出词汇表大小和前50个词

```
all_words = sorted(set(preprocessed))
vocab_size = len(all_words)
vocab = {token:integer for integer,token in enumerate(all_words)}
print(vocab_size)
for i, item in enumerate(vocab.items()):
    print(item)
    if i >= 50:
        break
```

课后任务：学习Python正则表达式模块re的用法，实现更多的标点符号的分割