位运算、命令行、LLM 与总结

C 语言讲座第 5 讲

代思聪

北京理工大学 | 睿信科协

2024-11-17

Outline

1. 位运算

1.1 0x5f3759df

你知道如何用一种不寻常的方法计算平方根倒数吗?

$$y = \frac{1}{\sqrt{x}}$$

1.1 0x5f3759df

你知道如何用一种不寻常的方法计算平方根倒数吗?

$$y = \frac{1}{\sqrt{x}}$$

平方根和倒数在许多计算中很常见。特别是在 3D 图形中，需要大量的法线向量的归一化计算。

1.1 0x5f3759df

你知道如何用一种不寻常的方法计算平方根倒数吗?

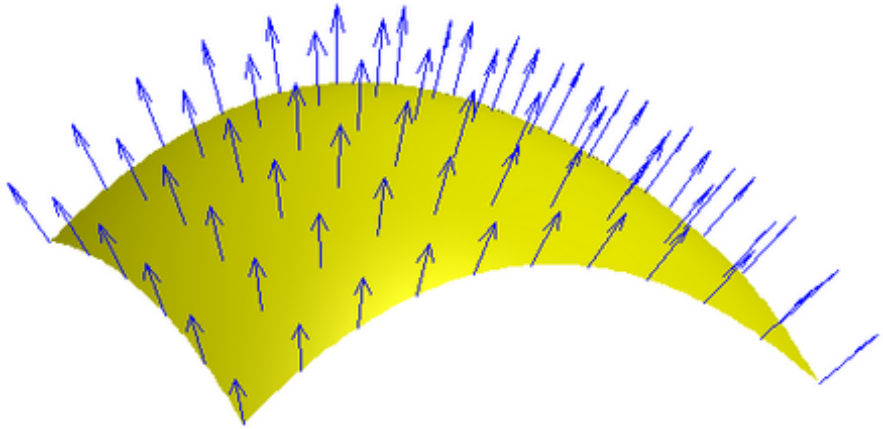
$$y = \frac{1}{\sqrt{x}}$$

平方根和倒数在许多计算中很常见。特别是在 3D 图形中，需要大量的法线向量的归一化计算。

如何在不使用 for 循环或复杂浮点运算的前提下，使用几行代码得到结果?

1.1 0x5f3759df

浮点数的平方根倒数常用于计算标准化向量。3D 图形程序需要使用正规化向量来实现光照和投影效果，因此每秒都需做上百万次平方根倒数运算，而在处理坐标转换与光源的专用硬件设备出现前，这些计算都由软件完成，计算速度亦相当之慢；在 1990 年代这段代码开发出来之时，多数浮点数操作的速度更是远远滞后于整数操作，因而针对标准化向量算法的优化就显得尤为重要。



1.1 0x5f3759df

要将一个向量标准化，首先计算向量的长度:

$$\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

1.1 0x5f3759df

要将一个向量标准化，首先计算向量的长度:

$$\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

将向量除以它的长度，得到标准化向量:

$$\hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$$

1.1 0x5f3759df

要将一个向量标准化，首先计算向量的长度:

$$\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

将向量除以它的长度，得到标准化向量:

$$\hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$$

可以看到，计算标准化向量时，计算平方根倒数是必须的，因此优化计算平方根倒数的算法相当有用。

1.1 0x5f3759df

计算的代码如下。其中，出现了 >> 运算符，这是什么？

```
float Q_rsqrt( float number )
{
  long i;
  float x2, y;
  const float threehalfs = 1.5F;

  x2 = number * 0.5F;
  y  = number;
  i  = * ( long * ) &y;      // evil floating point bit level hacking
  i  = 0x5f3759df - ( i >> 1 );  // what the freak?
  y  = * ( float * ) &i;
  y  = y * ( threehalfs - ( x2 * y * y ) );   // 1st iteration
  y  = y * ( threehalfs - ( x2 * y * y ) );   // 2nd iteration

  return y;
}
```

1.2 位运算介绍

位运算就是基于整数的二进制表示进行的运算。由于计算机内部就是以二进制来存储数据，位运算是相当快的。

基本的位运算共 6 种，分别为按位与、按位或、按位异或、按位取反、左移和右移。

1.2 位运算介绍

与、或、异或: 这三者都是两数间的运算。它们都是将两个整数作为二进制数，对二进制表示中的每一位逐一运算。

运算	运算符	解释
与	&	只有两个对应位都为 1 时才为 1
或	|	只有两个对应位都为 1 时才为 1
异或	^	只有两个对应位不同时才为 1

1.2 位运算介绍

与、或、异或: 这三者都是两数间的运算。它们都是将两个整数作为二进制数，对二进制表示中的每一位逐一运算。

运算	运算符	解释
与	&	只有两个对应位都为 1 时才为 1
或	|	只有两个对应位都为 1 时才为 1
异或	^	只有两个对应位不同时才为 1

注意，这里的运算符与逻辑的与、或含义不同，后者是 &&、||。

1.2 位运算介绍

与、或、异或运算的例子:

$$5 = (101)_2$$

$$6 = (110)_2$$

$$5 \ \& \ 6 = (100)_2 = 4$$

$$5 \ | \ 6 = (111)_2 = 7$$

$$5 \ \oplus \ 6 = (011)_2 = 3$$

1.2 位运算介绍

补码的复习：当我们想要表示一个负数的时候，比如 -5，该怎么办呢？

1.2 位运算介绍

补码的复习： 当我们想要表示一个负数的时候，比如 -5，该怎么办呢？

答案是使用一个二进制位来表示符号位，这一位通常是第一位。

原码：直接用最高位表示符号位。例如，-5 的原码是 10000101。

反码：正数的反码与原码相同，负数的反码是该数绝对值的原码按位取反（即 0 变 1，1 变 0），符号位不变。例如，-5 的反码是 11111010。

补码：正数的补码与原码相同。对于负数，取原码的反码，然后加 1 。例如，-5 的补码是 11111011。

1.2 位运算介绍

取反: 取反是一个单目运算，目的是把 num 的补码中的 0 和 1 全部取反。

例子:

$$
\begin{aligned}
5 &= (00000101)_2 \\
\sim 5 &= (11111010)_2 = -6 \\
-5 &= (11111011)_2 \qquad (\text{这里是补码}) \\
\sim (-5) &= (00000100)_2 = 4
\end{aligned}
$$

1.2 位运算介绍

左移和右移:

- `num << i` 表示将 `num` 的二进制向左移动 `i` 位得到的值。
- `num >> i` 表示将 `num` 的二进制向右移动 `i` 位得到的值。

例子:

$$11 = (00001011)_2$$

$$11 \ll 1 = (00010110)_2 = 22$$

$$11 \ll 3 = (01011000)_2 = 88$$

$$11 \gg 2 = (00000010)_2 = 2$$

1.2 位运算介绍

注意，如果 `i` 为负数，那么出现的行为是未定义的。例如 `num << -2` 没有意义。

1.2 位运算介绍

注意，如果 `i` 为负数，那么出现的行为是未定义的。例如 `num << -2` 没有意义。

如果 `i` 大于 `num` 的位数，例如对于 `int` 类型的 `num`（认为是 4 字节，具有 32 位），那么 `num << 33` 是未定义的。

1.2 位运算介绍

注意，如果 `i` 为负数，那么出现的行为是未定义的。例如 `num << -2` 没有意义。

如果 `i` 大于 `num` 的位数，例如对于 `int` 类型的 `num`（认为是 4 字节，具有 32 位），那么 `num << 33` 是未定义的。

在 C++中，右移使用的是算术右移——对于负数（在二进制中，最高位为 1），在执行右移操作时，右移后的高位也会填充 1。

如果我们有一个负数 -4，其在二进制中表示为 `11111100`，那么右移 1 位得到 `11111110`，对应的十进制数为 `-2`。

1.2 位运算介绍

符合赋值位运算符: 和 +=，-= 等运算符类似，位运算也有复合赋值运算符:

&=，|=，^= ，<<=，>>= 。（取反是单目运算，所以没有。）

1.3 位运算应用

位运算一般有三种作用:

1. 高效地进行某些运算，代替其它低效的方式。
2. 表示集合（常用于状态压缩）。
3. 题目要求位运算。

不过，也不要过度使用位运算来简化，因为编译器很多时候就会帮助你简化代码。代码的清晰性、可读性也是很重要的。

1.3 位运算应用

有关 2 的幂的应用

将一个数乘（除）2 的非负整数次幂:

```
int mulPowerOfTwo(int n, int m) {  // 计算 n*(2^m)
  return n << m;
}

int divPowerOfTwo(int n, int m) {  // 计算 n/(2^m)
  return n >> m;
}
```

1.3 位运算应用

取两个数的最大/最小值 假设 int 使用 32 位存储:

```
// 如果 a >= b, (a - b) >> 31 为 0, 否则为 -1
int max(int a, int b)
{
  return (b & ((a - b) >> 31)) | (a & (~(a - b) >> 31));
}
int min(int a, int b)
{
  return (a & ((a - b) >> 31)) | (b & (~(a - b) >> 31));
}
```

1.3 位运算应用

```
return (b & ((a - b) >> 31)) | (a & (~(a - b) >> 31));
```

max 的解释:

- (a - b) >> 31 的作用: 由于整数在内存中是以二进制补码形式存储的, 最高位 (第 31 位) 是符号位。对于 32 位整数:
  - 如果 a - b >= 0, 即 a >= b, (a - b) 的符号位是 0, 右移 31 位得到的仍然是 0。
  - 如果 a - b < 0, 即 a < b, (a - b) 的符号位是 1。右移 31 位后, 得到的值是全 1 的数, 即 −1 (在二进制补码表示中, -1 的二进制全为 1: 111...111)。
- (a - b) >> 31 的结果要么是 0 (a >= b), 要么是-1 (a < b), 这实际上是简化的条件逻辑。
- 利用 b & ((a - b) >> 31), 如果 a >= b, 那么 ((a - b) >> 31) 为 0, 表达式结果为 0; 相反, 如果 a < b, 此时 ((a - b) >> 31) 是 −1, 则表达式返回 b。

1.3 位运算应用

```
return (b & ((a - b) >> 31)) | (a & (~(a - b) >> 31));
```

max 的解释:

- 同样，对于 a，通过按位取反（ ）给出相反的判断。
- 使用按位或操作符 | 来合并两个分支的选择结果。最终会返回 a 或 b。

对于 min 来说，也是类似的。

1.3 位运算应用

例子: 设定 a = 5, b = 8:

max(5, 8):

- Step 1: a - b = 5 - 8 = -3
- Step 2: -3 >> 31, 因为 −3 是负数, 符号位是 1, 右移 31 位后, 结果为 -1。
- Step 3: 计算表达式:
  - b & ((a - b) >> 31) = 8 & (-1) = 8
  - a & (~(a - b) >> 31) = 5 & (~(-1)) = 5 & 0 = 0
- Step 4: 结果为 8 | 0 = 8。

因此, max(5, 8) 的结果是 8。

1.3 位运算应用

操作一个数的二进制位:

获取一个数二进制的某一位:

```
// 获取 a 的第 b 位，最低位编号为 0
int getBit(int a, int b) { return (a >> b) & 1; }
```

将一个数二进制的某一位设置为 0:

```
// 将 a 的第 b 位设置为 0 ，最低位编号为 0
int unsetBit(int a, int b) { return a & ~(1 << b); }
```

将一个数二进制的某一位取反:

```
// 将 a 的第 b 位取反 ，最低位编号为 0
int flapBit(int a, int b) { return a ^ (1 << b); }
```

1.3 位运算应用

交换两个数:

```
a=a^b;
b=a^b;
a=a^b;
```

1.3 位运算应用

交换两个数:

```
a=a^b;
b=a^b;
a=a^b;
```

补充知识点:

- 异或运算具有交换律、结合律。
- 0 和任意数字进行异或操作结果为数字本身。
- 两个相同的数字进行异或的结果为 0。

1.3 位运算应用

原理:

$$a = a \hat{} b$$

$$b = (a \hat{} b) \hat{} b = a \hat{} (b \hat{} b) = a \hat{} 0 = a$$

$$a = (a \hat{} b) \hat{} (a) = (a \hat{} a) \hat{} b = 0 \hat{} b = b$$

1.3 位运算应用

经典面试题: 给定一个非空整数数组，除了某个元素只出现一次以外，其余每个元素均出现两次。找出那个只出现了一次的元素。

说明: 你的算法应该具有线性时间复杂度。 你可以不使用额外空间来实现吗?

1.3 位运算应用

经典面试题：给定一个非空整数数组，除了某个元素只出现一次以外，其余每个元素均出现两次。找出那个只出现了一次的元素。

说明：你的算法应该具有线性时间复杂度。你可以不使用额外空间来实现吗？

使用刚刚的思想，两个相同数的异或为 0，因此只用把全部数字都异或一遍，剩下的数字就是要找的数字。

1.3 位运算应用

代码:

```
int find_only_number(int[] arr, int n)  // arr 是数组，n 是数组元素的个数
{
  int value = 0;
  for (int i = 0; i < n; i++)
  {
    value ^= arr[i]; // 使用异或赋值运算
  }
  return value;
}
```

1.3 位运算应用

经典面试题 2: 给定一个非空整数数组（使用 32 位存储），除了某个元素只出现一次以外，其余每个元素均出现了三次。找出那个只出现了一次的元素。说明：你的算法应该具有线性时间复杂度。你可以不使用额外空间来实现吗?

1.3 位运算应用

经典面试题 2: 给定一个非空整数数组（使用 32 位存储），除了某个元素只出现一次以外，其余每个元素均出现了三次。找出那个只出现了一次的元素。说明：你的算法应该具有线性时间复杂度。你可以不使用额外空间来实现吗?

判断整除求余操作，按位进行。

1.3 位运算应用

代码实现:

```
int find_only_number(int[] arr, int n)  // arr 是数组，n 是数组元素的个数
{
  int value = 0;
  for (int i = 0; i < 32; i++)  // 整数位 32 位，遍历每一位
  {
    int sum = 0;
    for (int j = 0; j < n; j++)  // 遍历 arr 数组
    {
      if (((arr[j] >> i) & 1) == 1)  // arr[j] 的第 i 位为 1，就增加 sum
        sum++;
    }
    if (sum % 3 == 1)  // 对 3 求余
      value += (1 << i);  //  将 1 写入 value 的第 i 位
  }
  return value;
}
```

1.4 平方根倒数的秘密

因为时间所限，无法完全讲解，详情请参考 B 站视频:

从 00:00 开始分享　获取视频分享链接

手机扫码观看/分享

1.5 参考资料

1. Wikipedia. 平方根倒数速算法.[1]
2. OI WIKI. 位运算.[2]
3. 量子位. 什么代码让程序员之神感叹“卧槽”？改变游戏行业的平方根倒数算法.[3]

[1] https://zh.wikipedia.org/wiki/%E5%B9%B3%E6%96%B9%E6%A0%B9%E5%80%92%E6%95%B0%E9%80%9F%E7%AE%97%E6%B3%95

[2] https://oi-wiki.org/math/bit/#%E5%B7%A6%E7%A7%BB%E5%92%8C%E5%8F%B3%E7%A7%BB

[3] https://www.bilibili.com/video/BV18j411i7bp

Q&A

2. 命令行

2.1 引入

为什么需要学习命令行?

2.1 引入

为什么需要学习命令行?

下面许多内容基于类 Unix 系统的命令，例如 Linux 与 macOS。对于 Windows 的 PowerShell（注意，不是 **cmd**）当中的命令，如果和类 Unix 命令相同，不会特殊标注；如果不同，会使用括号标注。

2.2 简介

命令行界面（英语: Command-line interface，缩写: **CLI**）是在图形用户界面得到普及之前使用最为广泛的用户界面，它通常不支持鼠标，用户通过键盘输入指令，计算机接收到指令后，予以执行。

2.2 简介

命令行界面（英语: Command-line interface，缩写: **CLI**）是在图形用户界面得到普及之前使用最为广泛的用户界面，它通常不支持鼠标，用户通过键盘输入指令，计算机接收到指令后，予以执行。

与之相对的，图形用户界面（英语: Graphical User Interface，缩写: **GUI**）是指采用图形方式显示的计算机操作用户界面。目前，各种电脑和手机，主要都使用 GUI 作为交互方式，降低了大众的学习成本。不过，相比 CLI，GUI 的拓展性较弱。

2.2 简介

命令行界面（英语：Command-line interface，缩写：**CLI**）是在图形用户界面得到普及之前使用最为广泛的用户界面，它通常不支持鼠标，用户通过键盘输入指令，计算机接收到指令后，予以执行。

与之相对的，图形用户界面（英语：Graphical User Interface，缩写：**GUI**）是指采用图形方式显示的计算机操作用户界面。目前，各种电脑和手机，主要都使用 GUI 作为交互方式，降低了大众的学习成本。不过，相比 CLI，GUI 的拓展性较弱。

应用程序接口（英语：application programming interface，缩写：**API**）是一种计算接口，它定义多个软件中介之间的交互。它还可以提供扩展机制，以便用户可以通过各种方式对现有功能进行不同程度的扩展。这是最难使用的方式，不过对于程序员来说，是定制程度最高的方式。广义地讲，你现在使用的各种 C 语言库函数，也可以看作是 API。

2.3 常见命令

- cd: 用于更改当前目录（文件夹）
- cp: 用于复制文件和目录
- ls: 用于列出目录中的文件
- mkdir: 用于创建一个目录
- mv: 用于移动（重命名）文件和目录
- rm: 用于删除文件
- rmdir: 用于删除目录

（此处有 demo）

2.4 命令的组成

```
command param1 -o arg1 --option arg2
```

命令的不同的组成部分之间使用空格分割，空格的个数是任意的。 第一个字符串是要使用的命令。

2.4 命令的组成

```
command param1 -o arg1 --option arg2
```

命令的不同的组成部分之间使用空格分割，空格的个数是任意的。第一个字符串是要使用的命令。

后面可以带有一个或者多个位置参数，例如 `cp source.txt destination.txt` 当中的 `source.txt` 和 `destination.txt` 就是位置参数。

2.4 命令的组成

```
command param1 -o arg1 --option arg2
```

命令的不同的组成部分之间使用空格分割，空格的个数是任意的。第一个字符串是要使用的命令。

后面可以带有一个或者多个位置参数，例如 `cp source.txt destination.txt` 当中的 `source.txt` 和 `destination.txt` 就是位置参数。

短选项：使用一个短横线（`-`）开头，后跟一个单字母。短选项后面的下一个字符串就是该选项的参数。例如，在上面的命令中，对于短选项`-o`，`arg1` 就是该选项的参数。

2.4 命令的组成

```
command param1 -o arg1 --option arg2
```

命令的不同的组成部分之间使用空格分割，空格的个数是任意的。第一个字符串是要使用的命令。

后面可以带有一个或者多个位置参数，例如 `cp source.txt destination.txt` 当中的 `source.txt` 和 `destination.txt` 就是位置参数。

短选项：使用一个短横线（`-`）开头，后跟一个单字母。短选项后面的下一个字符串就是该选项的参数。例如，在上面的命令中，对于短选项`-o`，`arg1` 就是该选项的参数。

长选项：使用两个短横线（`--`）开头，后跟完整的单词或单词组合。长选项后面的下一个字符串就是该选项的参数。例如，在上面的命令中，对于长选项`--option`，`arg2` 就是该选项的参数。

2.4 命令的组成

标志（Flags）：类似于选项，但不接受参数，通常用于开关某个功能。例如，`-h` 或 `--verbose` 可以作为标志。

带有等号的长选项：某些长选项可以用等号来直接赋值。例如，`--output=file.txt`。

复合命令和子命令：某些命令行工具支持复合命令，其中主命令后面跟随一个子命令。例如，`git commit` 中的 `commit` 就是一个子命令。

2.5 更多的命令

- date
- echo
- $PATH（$env:path）
- which（Get-Command）
- man
- …

（此处有 demo）

2.6 参考资料

1. 计算机教育缺失的一课(2020) - 第 1 讲 - 课程概览与 shell.[1]
2. 课程概览与 shell.[2]
3. 中国科学技术大学 Linux 用户协会. Linux 101.[3]
4. 命令行的艺术.[4]
5. Wikipedia. Command-line interface.[5]

[1] https://www.bilibili.com/video/BV1uc411N7eK
[2] https://missing-semester-cn.github.io/2020/course-shell/
[3] https://101.lug.ustc.edu.cn
[4] https://github.com/jlevy/the-art-of-command-line/blob/master/README-zh.md
[5] https://en.wikipedia.org/wiki/Command-line_interface

Q&A

3. LLM

3.1 什么是大语言模型

一段很官方的解释:

大型语言模型，也称为 LLM，是基于大量数据进行预训练的超大型深度学习模型。底层转换器是一组神经网络，这些神经网络由具有自注意力功能的编码器和解码器组成。编码器和解码器从一系列文本中提取含义，并理解其中的单词和短语之间的关系。

3.1 什么是大语言模型

一段很官方的解释:

大型语言模型，也称为 LLM，是基于大量数据进行预训练的超大型深度学习模型。底层转换器是一组神经网络，这些神经网络由具有自注意力功能的编码器和解码器组成。编码器和解码器从一系列文本中提取含义，并理解其中的单词和短语之间的关系。

这都是什么意思???

3.1 什么是大语言模型

看一段视频的前 10 分钟，简单了解一下什么是 LLM:

【渐构】万字科普 GPT4 为何会颠覆现有工作流；为何你要关注微软 Copilot、文心一言等大模型

https://www.bilibili.com/video/BV1MY4y1R7EN/

3.2 实际例子

假设 LLM 接受到了以下输入:

Massachusetts is a state in the New England region of the Northeastern United States. It borders on the Atlantic Ocean to the east. The state's capital is...

3.2 实际例子

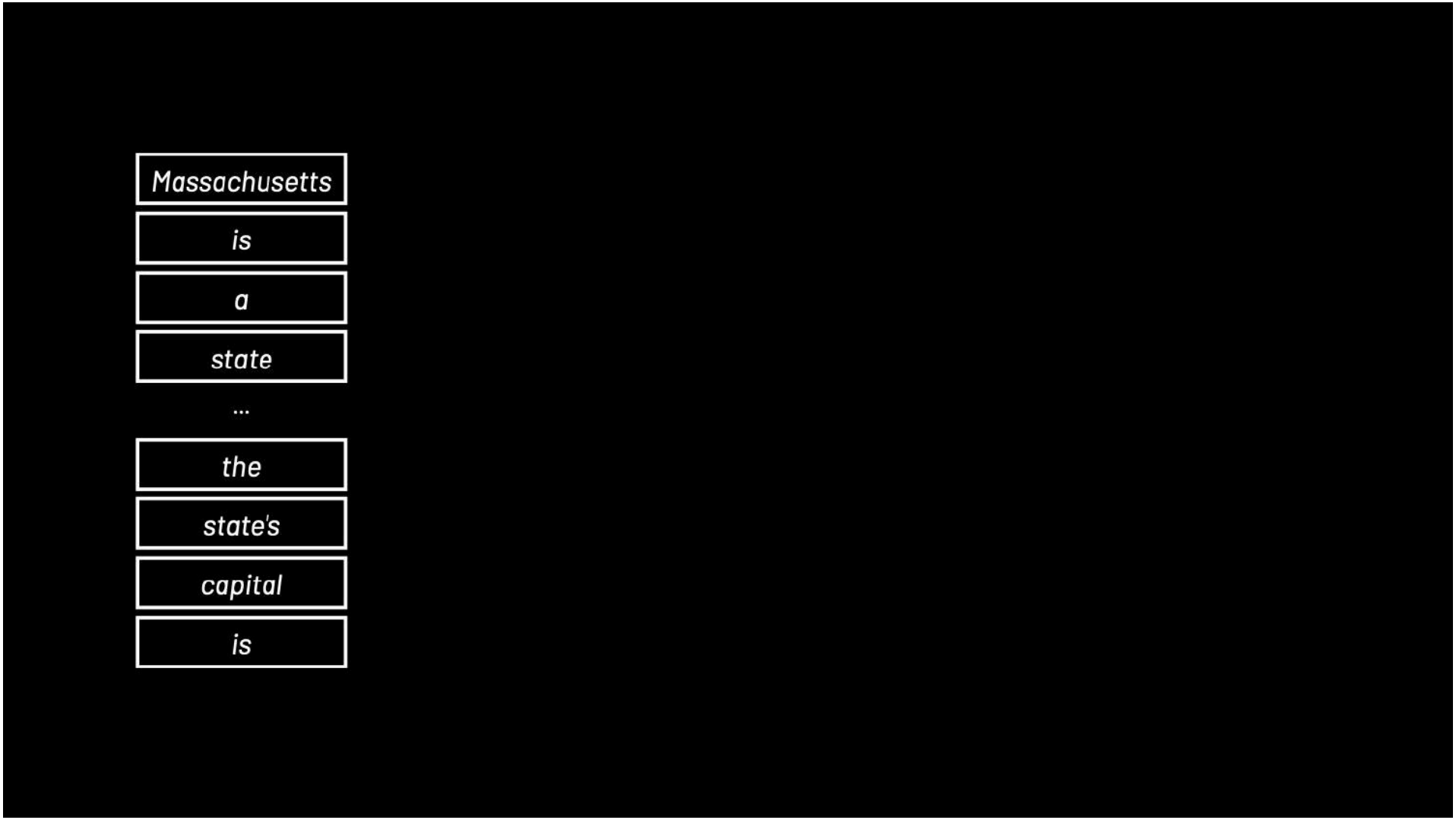
假设 LLM 接受到了以下输入:

Massachusetts is a state in the New England region of the Northeastern United States. It borders on the Atlantic Ocean to the east. The state's capital is...

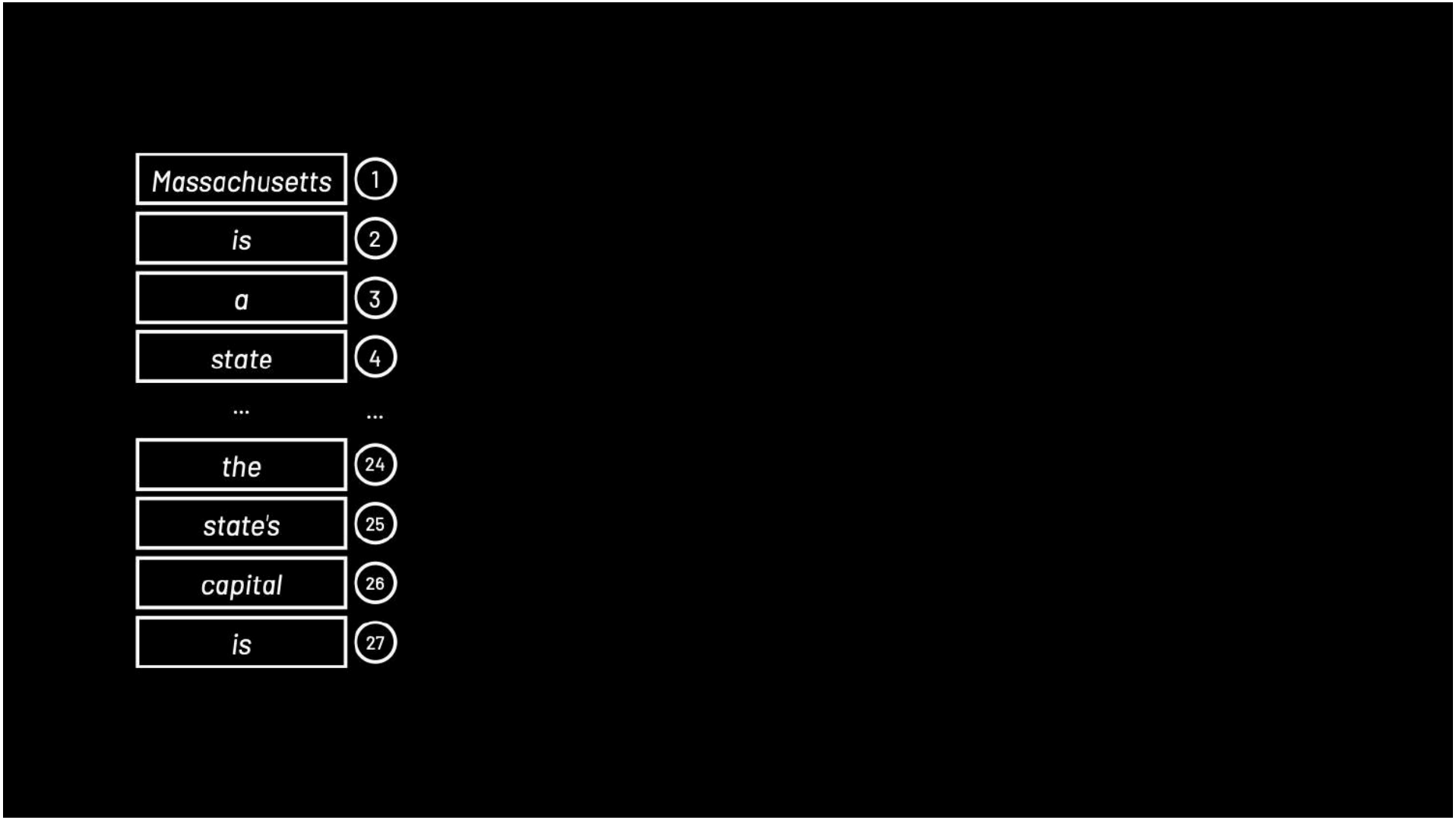
其中，模型应该注意到最后的 state 指的是麻省:

Massachusetts is a state in the New England region of the Northeastern United States. It borders on the Atlantic Ocean to the east. The state's capital is...

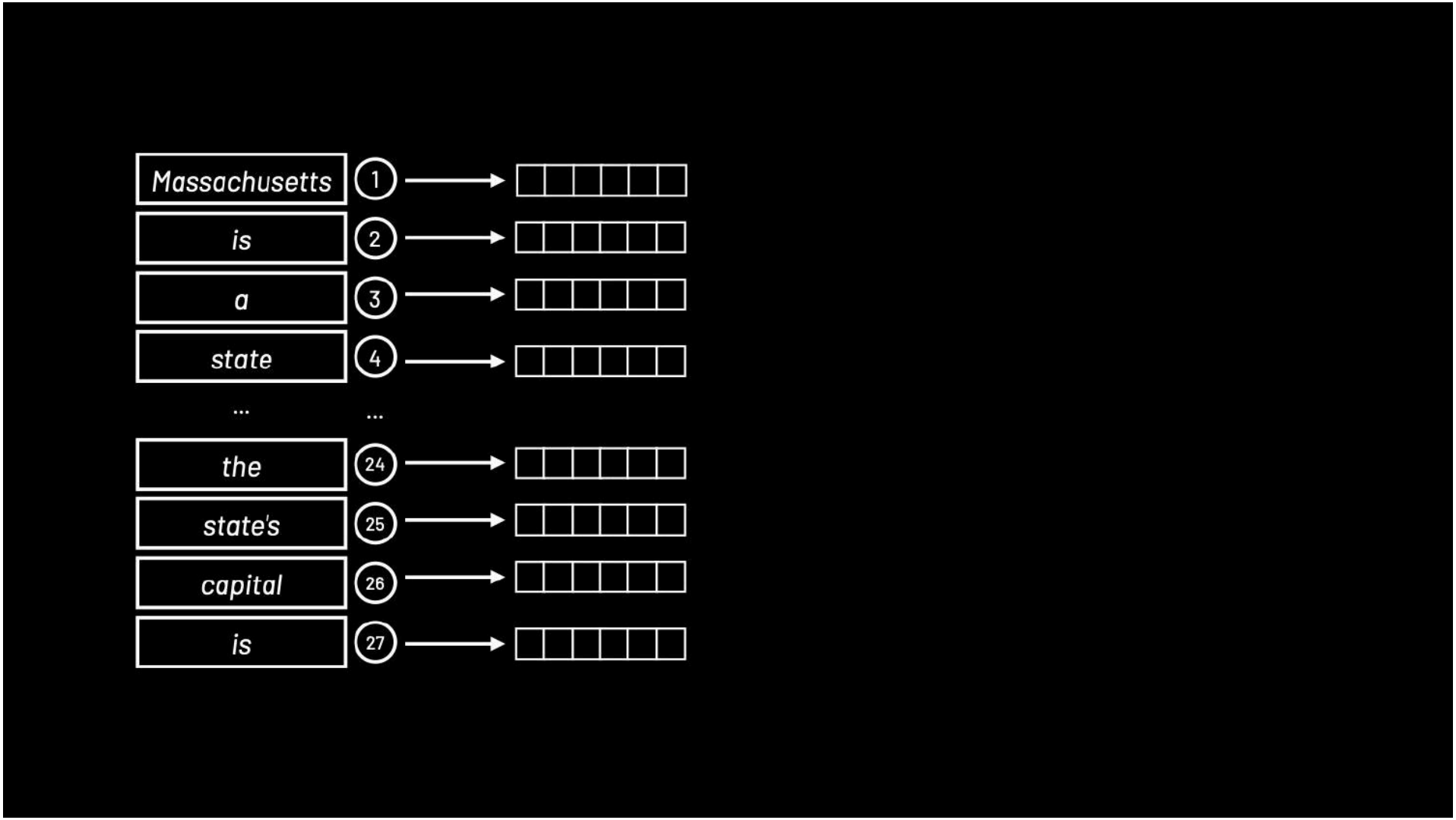
3.2 实际例子



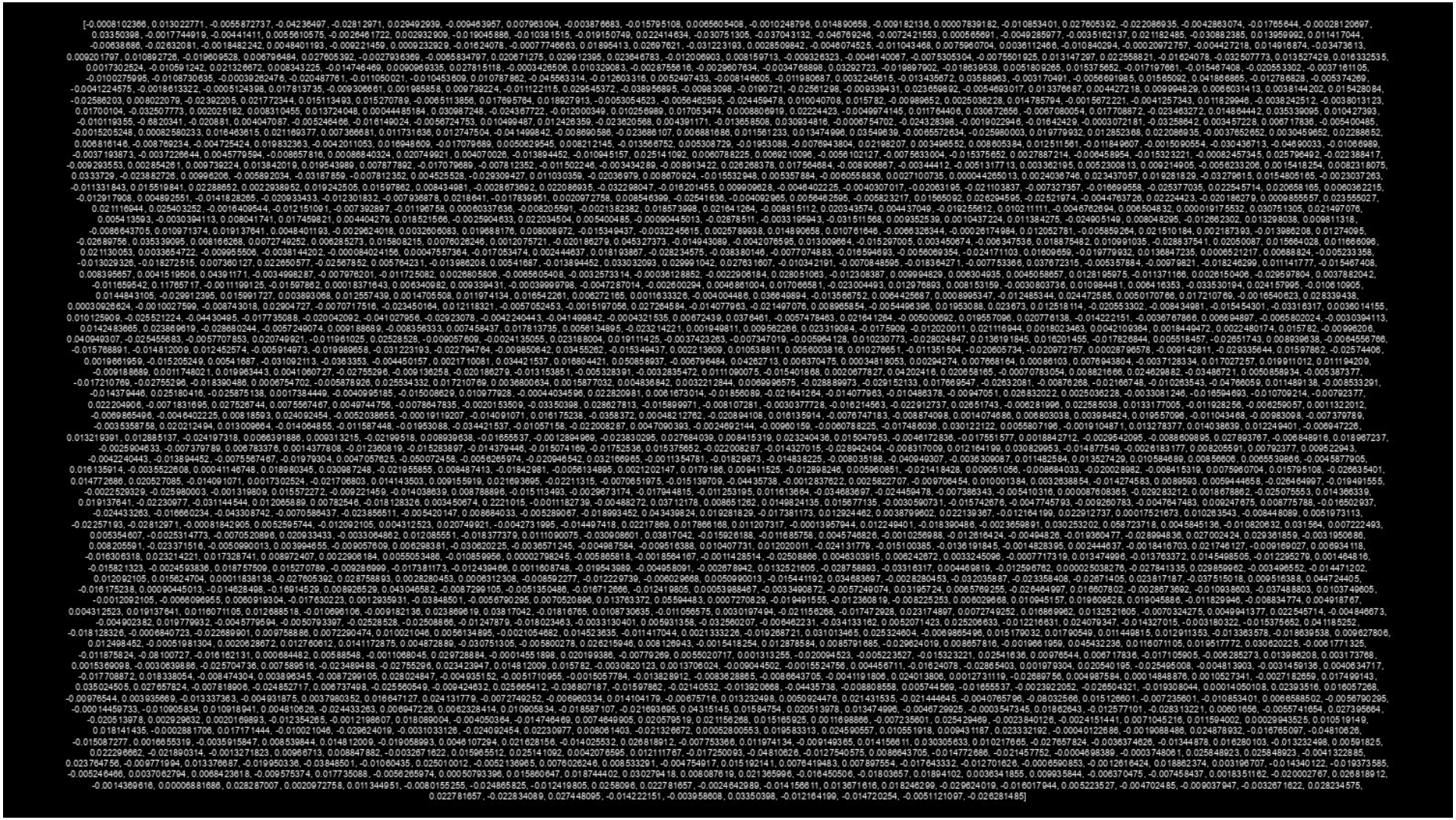
3.2 实际例子


Massachusetts
1
is
2
a
3
state
4
...
...
the
24
state's
25
capital
26
is
27

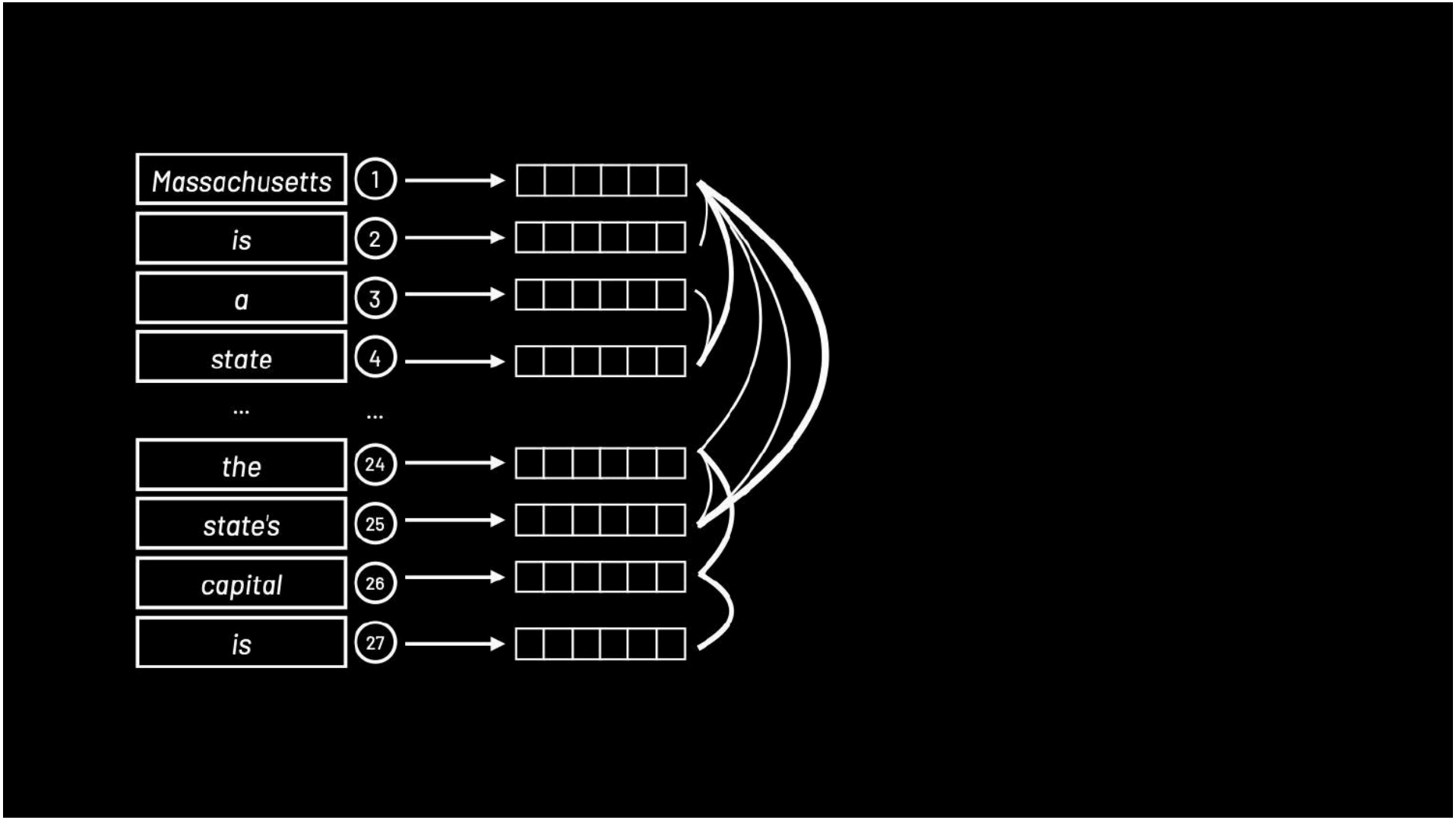
3.2 实际例子


Massachusetts
1
is
2
a
3
state
4
...
...
the
24
state's
25
capital
26
is
27

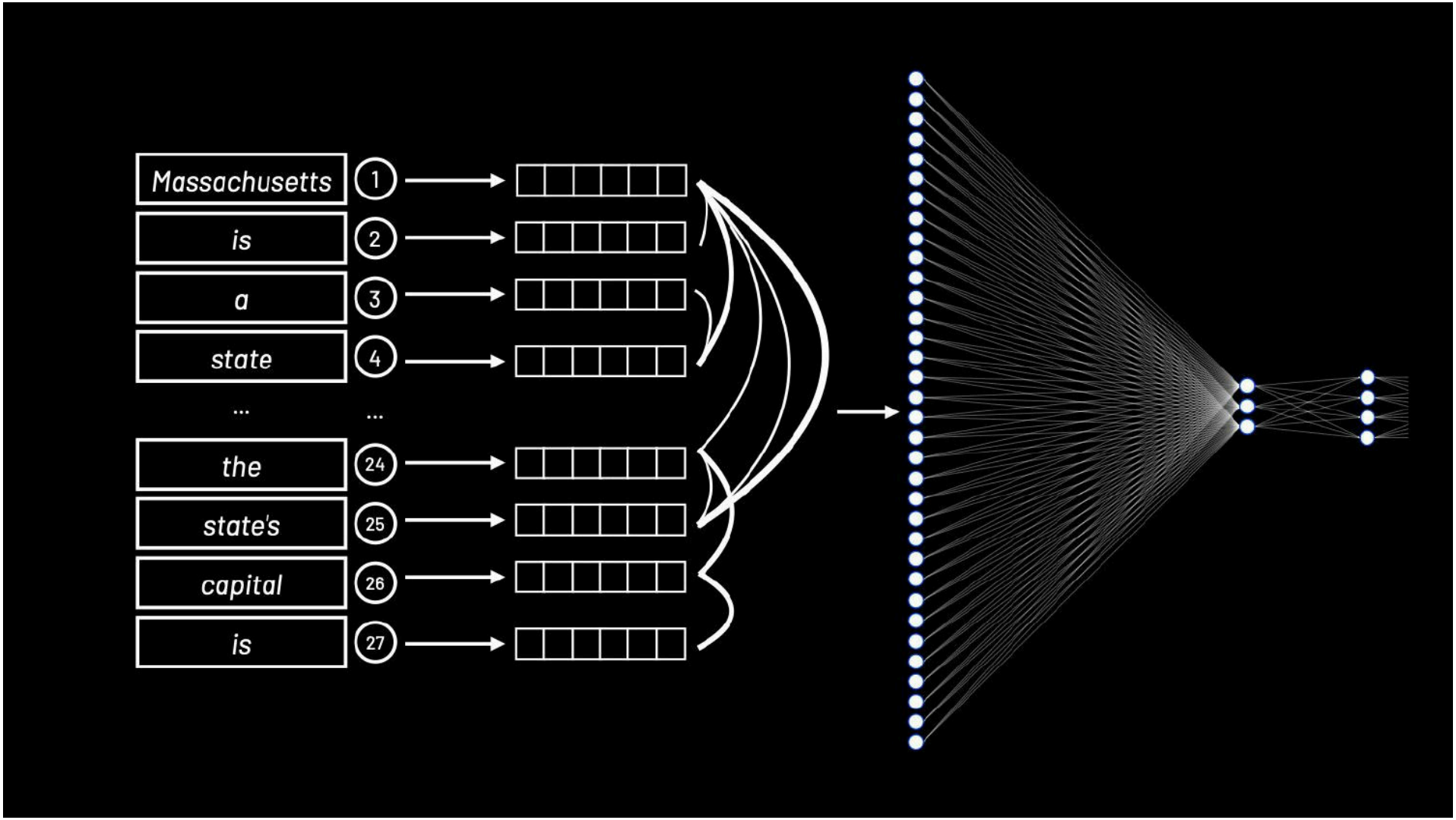
3.2 实际例子


[-0.0008102366, 0.013022771, -0.0055872737, -0.04236497, -0.02812971, 0.029492939, -0.009463957, 0.007963094, -0.003876683, -0.015795108, 0.0065605408, -0.0010248796, 0.014890658, -0.009182136, 0.00007839182, -0.010853401, 0.027605392, -0.022086935, -0.0042863074, -0.01765644, -0.00028120697,
0.03350398, -0.0017744919, -0.00441411, 0.0055610575, -0.002646722, 0.002932909, -0.019045886, -0.010381515, -0.019150749, 0.022414634, -0.030751305, -0.037043132, -0.046769246, -0.0072421553, 0.0005565691, -0.0049285977, -0.0035162137, 0.021182485, -0.030882385, 0.013959992, 0.011417044,
-0.00638686, -0.02632081, -0.0018482242, 0.0048401193, -0.009221459, 0.0009232929, -0.01624078, -0.00077746663, 0.01895413, 0.02897621, -0.031223193, 0.0028509842, -0.0046074525, -0.011043468, 0.0075960704, 0.0036112466, -0.010840294, -0.00020972757, -0.004427218, 0.014916874, -0.0347361 3,
0.009201797, 0.010892726, -0.019609528, 0.006796484, 0.027605392, -0.0027936369, -0.0065834797, 0.020671275, 0.02991239 5, 0.023646783, -0.012006903, 0.008159713, -0.009326323, -0.0046140067, -0.0075305304, -0.0075501925, 0.013147297, 0.022558821, -0.01624078, -0.032507773, 0.013527429, 0.016332535,
0.0017302524, -0.010591242, 0.021326672, 0.008343225, -0.014746469, 0.0090969335, 0.027815118, -0.0003426506, 0.010329083, -0.002875516 16, -0.0029607634, -0.0034768898, 0.03292723, -0.019897902, -0.018639538, 0.0051809265, 0.015375652, -0.017197661, -0.015467408, -0.020553302, -0.0037161105,
-0.0100275995, -0.0108730635, -0.0003926247 6, -0.020487761, -0.011050021, -0.010453609, 0.010787862, -0.045563314, -0.012603316, 0.0052497433, -0.008146605, -0.011980687, 0.0032245615, -0.013435872, 0.03588963, -0.003170491, -0.0056691985, 0.01565092, 0.041866865, -0.012786828, -0.005374269,
-0.0041224575, -0.0018613322, -0.0005124398, 0.017813735, -0.009306661, 0.001985858, 0.009739224, -0.011122115, 0.029545372, -0.038956895, -0.00983098, -0.0190721, -0.025612 98, -0.009339431, 0.023659892, -0.0054693017, 0.013376887, 0.004427218, 0.009994829, 0.0086031413, 0.003814420 2, 0.015428084,
-0.02586203, 0.008022079, -0.02392205 0, 0.021772344, 0.015113493, 0.015270789, -0.0065113856, 0.017695764, 0.018927913, -0.0053054523, -0.0056462595, -0.024459478, 0.010040708, 0.015782, -0.00989852 6, 0.0025036228, 0.014785794, -0.0051672221, -0.0041257343, 0.011829948, -0.0038242512, -0.003801312 3,
0.01700104, -0.032507773, 0.002025182, 0.008310455, 0.013724048, 0.00044485184, 0.030987248, -0.024367722, -0.012000349, 0.010256989, 0.017053474, 0.0008806919, 0.02224423, -0.0049974145, 0.011764406, 0.030672658, -0.0067080054, 0.017708872, -0.023463272, 0.014884442, 0.035339095, 0.010427393,
-0.010119355, -0.6820341, -0.020881, 0.004047087, -0.005248466, -0.016149024, -0.0056724753, 0.010499487, 0.012426359, -0.023620568, 0.004391171, -0.013658508, 0.030934816, -0.0006754702, -0.024328398, -0.0019022946, -0.01642429, -0.0003072181, -0.03258842, 0.003457228, 0.008717836, -0.005400485,
-0.0015205248, 0.00082580233, 0.016463615, 0.021169377, 0.007366681, 0.011731636, 0.012747504, -0.04149984 2, -0.008890586, -0.023686107, 0.006881686, 0.011561233, 0.013474996, 0.03549639, -0.0065572634, -0.025980003, 0.019779932, 0.012852368, 0.022086935, -0.037652652, 0.003049652, 0.022888652,
0.006816146, -0.008769234, -0.004725424, 0.019832363, -0.0042011053, 0.016948609, -0.017079689, 0.0050629545, 0.008212145, -0.013566752, 0.005308729, -0.019153088, -0.007694384, 0.02198207, 0.003496552, 0.008605384, 0.012511561, -0.011849607, -0.0015090554, -0.030436713, -0.04690033, -0.01066989,
-0.0037193873, -0.0037226644, 0.004577959 4, -0.008657816, 0.0008884 0324, 0.020749921, 0.004070026, -0.013894452, -0.010945157, 0.025141092, 0.0060788225, 0.006921096, -0.0056102127, -0.007583 3004, -0.015375652, 0.027887214, -0.006458954, -0.015323221, -0.0008245734 5, 0.025796492, -0.022388417,
-0.009293553, 0.002854281, 0.009739224, 0.013842019, 0.019543989, 0.007877892, -0.017079689, -0.007812352, -0.011502246, -0.003434289, -0.008913422, 0.026268378, 0.017564884, -0.008906867, -0.003444 12, -0.005131777 3, 0.003362195, 0.005230083 3, 0.009214905, -0.0056233206, 0.001541825 4, 0.008231807 5,
0.033729, -0.023882728, 0.00996206, -0.005892034, -0.03187859, -0.007812352, 0.004525528, -0.029309427, 0.011303059, -0.02036979, 0.008670924, -0.015532948, 0.005357884, -0.0060555836, 0.0027100735, 0.000044265013, 0.024036746, 0.023437057, 0.019281829, -0.03279615, 0.015480516 5, -0.023037263,
-0.011331843, 0.011519841, 0.02288652, 0.002293895 2, 0.019242505, 0.01597862, 0.008434981, -0.0028673692, 0.022086935, -0.032298047, -0.016201455, 0.009909628, -0.0046402225, -0.0040307017, -0.02063195, -0.021103837, -0.007327357, -0.016699558, -0.025377035, 0.022545714, 0.020658165, 0.0060362215,
-0.012917908, 0.004892551, -0.0141828265, -0.020933433, -0.012301832, -0.007936878, 0.0218641, -0.017839951, 0.0020972758, 0.008546399, -0.02541636, -0.004092965, 0.005646259 5, -0.005823217, 0.01565092, 0.026294595, -0.02521974, -0.004476372 6, 0.02224423, -0.020186279, 0.009985555 7, 0.023555027,
0.021116944, 0.025403252, -0.0016409544, -0.012151091, -0.007392897, -0.011967 58, 0.000603376 36, -0.008205591, -0.002138238 2, 0.018573998, 0.021641264, -0.008815112, 0.020343574, 0.004437049, -0.019255512, 0.010211111, -0.004676269 4, 0.006504832, 0.000019175 532, 0.030751305, 0.021497076,
0.005413593, -0.0030394113, 0.008041741, 0.017459821, 0.004404279, 0.018521566, -0.0025904633, 0.022034504, 0.005400485, -0.009004450 13, -0.02878511, -0.003319594 3, -0.031511 568, 0.009352539, 0.0010437224, 0.011384275, -0.024905149, 0.008048295, -0.012662302, 0.013298038, 0.009811318,
-0.0086643705, 0.010971374, 0.019137641, 0.004840119 3, -0.0029624018, 0.003260608 3, 0.019688176, 0.008008972, -0.015349437, -0.003224565 15, 0.002578993 8, 0.014890658, 0.010761646, -0.0066326344, -0.00026174984, 0.012052781, -0.005859264, 0.021510184, 0.002187393, -0.013986208, 0.01274095,
-0.02689756, 0.035539095, 0.008166268, 0.007274925 2, 0.006285273, 0.015808215, 0.007602824 6, 0.0012075721, -0.020186279, 0.045327373, -0.014943089, -0.0042076595, 0.013009864, -0.015297005, 0.003450674, -0.006347536, 0.018875482, 0.010991035, -0.028837541, 0.02050087, 0.015664028, 0.011666096,
0.021130053, 0.0033654722, -0.009955508, -0.003814420 2, -0.000804024156, 0.000475573 64, -0.017053474, 0.002444637, 0.018193867, -0.028234575, -0.038380146, -0.007770478 3, -0.016594693, 0.005609354, -0.024171103, 0.01809659, -0.019779932, 0.013684725, 0.0006521217, 0.00688824, -0.005233358,
-0.013029326, -0.018272515, 0.007360127, 0.022650577, -0.02567852, 0.005764231, -0.013986208, 0.00541687, -0.013894452, 0.033302093, 0.029991042, 0.027631607, -0.010342191, -0.0070848595, -0.018364271, -0.007753366, 0.037672315, -0.005357884, -0.00979821, -0.018246299, 0.011141777, -0.015467408,
0.008395657, 0.0041519506, 0.04391171, -0.0034998287, -0.007976201, -0.011725082, 0.002680580 6, -0.0055605408, -0.003257331 4, -0.00036128852, -0.0022906184, 0.028051063, -0.012308387, 0.0099994829, 0.008304935, 0.0045058857, 0.012819597 5, -0.011371166, 0.0026150406, -0.029597804, 0.003788204 2,
-0.011659542, 0.11765717, -0.00 11199125, -0.01597862, 0.000183771643, 0.006340982, 0.009339431, -0.0039999798, -0.0047287014, -0.002600294, 0.0046861004, 0.017066581, -0.023004493, 0.012976893, 0.008153159, -0.0033003736, 0.010988481, 0.006416353, -0.033530194, 0.024157995, -0.010610905,
0.014484310 5, -0.029912395, 0.015991727, 0.003893068, 0.012557439, 0.0014705508, 0.011974134, 0.016542261, 0.006272165, 0.0011633326, -0.004004486, 0.036649894, -0.013566752, 0.006442568 7, 0.0008995347, -0.012485344, 0.024472585, 0.0050170766, 0.017210769, -0.016540623, 0.028339438,
0.00030926624, -0.0010027599, -0.008743018, 0.02904727, -0.0070717516, -0.023450164, 0.012118321, -0.0057052453, -0.0015197056, 0.027264584, -0.014077963, -0.021497076, 0.008965854, -0.0054496396, 0.0195308 8, 0.023673, 0.012518114, -0.020553302, -0.008434981, -0.015454301, -0.03316317, 0.003601415 5,
0.010125909, -0.025521224, -0.04430495, -0.017735088, -0.020042092, -0.041027956, -0.02923078, -0.0042240443, -0.041499842, -0.0004321535, 0.00672439, 0.0376461, -0.0057478463, 0.021641264, -0.005000692, 0.019557096, 0.020776138, -0.014222151, -0.0036767866, 0.006694897, -0.0065802024, -0.003039411 3,
0.0142483665, 0.023889619, -0.028880244, -0.005724907 4, 0.009188689, -0.008356333, 0.007458437, 0.017813735, 0.005613489 5, -0.023214221, 0.001949811 1, 0.009562266, 0.023319084, -0.0175909, -0.012020011, 0.021116944, 0.0018023463, 0.0042109364, 0.0018449472, 0.0022480174, 0.015782, -0.00996206,
0.040949307, -0.025455683, -0.0057707853, 0.020749921, -0.011961025, 0.02528528, -0.009057609, -0.0024135055, 0.023188004, 0.019111425, -0.0037423263, -0.007347019, -0.005964 128, 0.010230773, -0.028024847, 0.013619185, 0.016201455, -0.017826844, 0.005518457, -0.026517483, 0.008939638, -0.0064556766,
-0.015768891, -0.014812009, 0.012452574, -0.005914973, -0.019989658, -0.031223193, -0.022794764, -0.009850842, 0.03455282, -0.015349437, 0.002213609, 0.010538811, 0.005600381 6, 0.010278651, -0.011351504, -0.020605734, -0.020972757, 0.0002879658, -0.009142811, -0.029335644, 0.01597862, -0.02574406,
0.0019661959, -0.015205249, 0.0054 1687, -0.031092113, -0.0363353, -0.004450157, 0.002171 0081, 0.034421537, 0.016804421, 0.050858937, -0.006796484, 0.0426273 13, 0.006370475, 0.000348 1853, 0.00294274, 0.007668164, 0.00086103, 0.007694384, -0.003712833 4, 0.017027257, 0.019911012, 0.011194209,
-0.009188689, 0.001174802 1, 0.019963443, 0.004106072 7, -0.02755296, -0.009136258, -0.020186279, -0.013153851, -0.005328391, -0.003283547 2, 0.011190075, -0.015401868, 0.002067787 2, 0.04202416, 0.020658165, -0.000707835 4, 0.008821666, 0.024629882, 0.03486721, 0.005085834, -0.005387377,
-0.017210769, -0.02755296, -0.018390486, 0.006754702, -0.005878926, 0.025534332, 0.017210769, 0.003680063 4, 0.001587703 2, 0.004836842, 0.003221284 4, 0.0069996575, -0.028889973, -0.029152133, 0.017669547, -0.02632081, -0.008762 68, -0.021667 48, -0.010283543, -0.04766059, 0.011489138, -0.008533291,
-0.014379446, 0.025180416, -0.025875138, 0.001738444 9, -0.004099518 5, -0.015008629, 0.010977928, -0.000440345 96, 0.022820981, 0.006 1673 014, -0.01856089, -0.021641264, -0.014077963, -0.010486378, -0.00947051, 0.026832022, 0.0025036228, -0.0033081246, -0.016594693, -0.010709214, -0.007923 77,
0.022204906, -0.007 1831695, 0.027526744, 0.007556746 7, 0.004974475 6, -0.007864783 5, -0.0020153509, -0.033550398, 0.028627813, -0.015899971, -0.008107281, -0.0030377728, -0.016214563, -0.022912737, 0.026517483, -0.006281996, 0.022585038, 0.013177005, -0.011928256, -0.006259057, 0.001132012,
-0.006986549 6, -0.0046402225, 0.008 18593, 0.024092454, -0.005203865 5, -0.000191192 07, -0.014091071, 0.016175238, -0.0358372, 0.0004821276 2, -0.020894108, 0.016135914, -0.007674718 3, -0.008874098, 0.001407468 6, 0.006803038, 0.003984824, 0.019557096, -0.011043468, -0.0098309 8, -0.007379789,
-0.0035358758, 0.020212494, 0.013009664, -0.014064855, -0.011587448, -0.01953088, -0.034421537, -0.01057158, -0.022008287, 0.004709039 3, -0.002469214 4, -0.00960159, -0.006078822 5, -0.017486036, 0.030122122, 0.005580719 6, -0.009101487 1, 0.013278377, 0.014038639, 0.012249401, -0.006947228,
0.013219391, 0.012885137, -0.024197318, 0.006639188 6, 0.009313215, -0.02199518, 0.008939638, -0.01655537, -0.001289499 9, -0.023830295, 0.027684039, 0.008415319, 0.023240436, 0.015047953, -0.004617283 6, -0.017551577, 0.0018842712, -0.0029542095, -0.008860898 5, 0.027893767, -0.006848916, 0.018967237,
-0.0025904633, -0.007379789, 0.006783376, 0.014377808, -0.012360819, -0.015283897, -0.014379446, -0.015074169, -0.01752356, 0.015375652, -0.022008287, -0.014327015, -0.028942404, -0.008317009, 0.012164199, 0.030829953, -0.014877549, -0.002618317 7, 0.008205591, 0.00792377, 0.009522943,
-0.0042240443, -0.013894452, -0.007556746 7, -0.01979304, 0.004705762 5, -0.050072458, -0.005626597 4, -0.020946542, 0.032166965, -0.001135478 1, -0.01829873, -0.014838225, -0.008035188, -0.040949307, -0.003630908 7, 0.011482584, 0.013527429, 0.010584689, 0.00856606, 0.006553986 6, -0.004587905,
0.016135914, -0.0035522608, 0.0004114674 8, 0.018980345, 0.030987248, -0.021955855, 0.008487413, -0.018429 81, -0.005613489 5, 0.002120214 7, 0.017918 6, 0.009411525, -0.012898246, 0.005960851, -0.021418428, 0.009051056, -0.008684033, -0.020028982, -0.008415319, 0.007596070 4, 0.015795108, -0.026635401,
0.014772686, 0.020527085, -0.014091071, 0.001730252 4, -0.021706803, 0.014143503, 0.009155919, 0.021693695, -0.02211315, -0.007065197 5, -0.015139709, -0.04435738, -0.001283762 2, 0.002582270 7, -0.009706454, 0.010001384, 0.003263885 4, -0.014274583, 0.0089593, 0.005944465 8, -0.026464997, -0.019491555,
-0.0022529329, -0.025980003, -0.001319809, 0.015572272, -0.009221459, -0.014038639, 0.008788896, -0.015113493, -0.002967317 4, -0.017944815, -0.011253195, 0.011613664, -0.034683697, -0.024459478, -0.007386343, -0.005410316, 0.000087608385, -0.029283212, 0.001867886 2, -0.025075553, 0.014366339,
0.019137641, -0.02230977, -0.031144544, 0.012065889, 0.00782546, -0.018128326, 0.003450674, 0.2221015, -0.001118273 9, -0.00488272, 0.03712178, 0.008651262, 0.014982415 3, 0.015677135, -0.0030590731, -0.015742676, -0.004774579 3, -0.009260783, -0.004764783, 0.009247675, 0.008775788, -0.016502937,
-0.024433263, -0.016660234, -0.043308742, -0.007058643 7, -0.023856511, -0.005420147, 0.008684033, -0.005289067, -0.018993452, 0.043439824, 0.019281829, -0.017381173, 0.012924462, 0.003879602, 0.022139367, -0.012164199, 0.022912737, 0.000175216 73, 0.010263543, -0.008448089, 0.005197313,
-0.02257193, -0.02812971, -0.000818429 05, 0.005259574 4, -0.012092105, 0.004312523, 0.020749921, -0.004273199 5, -0.014497418, 0.02217889, 0.017866168, 0.011207317, -0.000139579 44, 0.012249401, -0.018390486, -0.002385898 1, 0.030253202, 0.058723718, 0.004584513 6, -0.018020832, 0.031564, 0.007222493,
0.005354607, -0.0025314773, -0.007052089 6, 0.020933433, -0.003306486 2, 0.012085551, -0.018377379, 0.011190075, -0.030908801, 0.03817042, -0.015926188, -0.011685758, 0.004574682 6, -0.004948 26, -0.019360477, -0.028994836, 0.027002424, 0.029361859, -0.003195068 6,
0.008205591, -0.023371516, -0.005099003, 0.003994655, -0.009057609, 0.006298381, -0.030620225, -0.036571245, -0.004987584, -0.009516388, 0.010407731, 0.012020011, -0.024131779, -0.015100385, -0.013619185, -0.001482839 5, 0.002444837, -0.001841670 3, 0.021746127, -0.009169027, 0.006934118,
-0.016306318, 0.023214221, 0.017328741, 0.008972407, 0.022906184, 0.005505348 6, -0.010859956, 0.00002798245, -0.005865818, -0.001856416 7, -0.011428514, -0.025088866, 0.004630391 5, 0.014548505, -0.012295279, 0.001464816,
-0.015821323, -0.0024593836, 0.018757509, 0.015270789, -0.009286999, -0.017381173, -0.012439466, 0.001160874 8, -0.019543989, -0.004958091, -0.002678942, 0.013252160 5, -0.028758893, -0.03316317, 0.004469819, -0.012596762, 0.000025038276, -0.027841335, 0.029859962, -0.003496552, -0.014471202,
0.012092105, 0.015624704, 0.000118381 38, -0.027605392, 0.028758893, 0.002820845 3, 0.000631230 8, -0.008592277, -0.012229739, -0.006029668, 0.0050990013, -0.015441192, 0.034683697, -0.022820845 3, -0.023035887, -0.023358408, -0.026714057, 0.023187187, -0.037515018, 0.009516388, 0.044724405,
-0.016175238, 0.00090445013, -0.014628498, -0.16914529, 0.008926529, 0.043046582, -0.008729910 5, -0.0051350486, -0.016712666, -0.012419805, 0.0053988467, -0.003349087 2, -0.005724907 4, 0.03195724, 0.006576925 5, -0.026464997, 0.016078 02, -0.002867 3692, -0.010938603, -0.037488803, 0.010374960 5,
-0.0012092105, -0.0066096955, 0.006091934 0, -0.017630223, 0.001293593 1, -0.03848501, -0.005679029 5, 0.007052089 6, 0.013763372, 0.05594483, 0.000727082 9, -0.019491555, -0.012306819, -0.008225253, 0.006029668, 0.010945157, 0.019609528, 0.019045886, -0.011829946, -0.008834774, 0.004918767,
0.004312523, 0.019137641, 0.011607110 5, 0.012688518, -0.010696106, -0.009182136, 0.023869619, 0.038170 42, -0.018167 65, 0.010873063 5, -0.011056575, 0.003019794, -0.021156268, -0.017472928, 0.023174897, 0.007274925 2, 0.016869962, 0.013252160 5, -0.007032427 5, 0.004994137 7, 0.022545714, -0.004846673,
-0.004902382, 0.019779932, -0.004577959 4, -0.005079399 7, -0.02528528, -0.025088666, -0.012478879, -0.018203463, -0.003313040 1, 0.005931358, -0.032560207, -0.006462231, -0.034133162, 0.052071423, 0.025206633, -0.012216631, 0.024079347, -0.014327015, -0.003180322, -0.015375652, 0.04185252,
-0.018128326, -0.0006840723, -0.022689901, 0.009758886, 0.007229047 4, 0.010021046, 0.005613489 5, -0.0021054682, 0.014523635, -0.011417044, 0.002133326, -0.019268721, 0.031013465, 0.025324604, -0.006986596, 0.015179032, 0.01790549, 0.011449815, 0.012911353, -0.013363578, -0.018639538, 0.009627806,
0.012498452, -0.00051981304, 0.002062867 2, 0.012760612, 0.014117287 5, 0.004872899, -0.030751305, -0.005800278, 0.026215946, 0.008126943, -0.001541825 4, 0.012878584, 0.008579168 5, -0.019966 1959, 0.045432236, 0.011607110 5, 0.019517772, 0.030620225, -0.006 1771 325,
-0.011875824, -0.08 100727, -0.016162131, 0.000684482, 0.005885 48, -0.001106804 5, 0.029728884, -0.000145518 98, 0.020199386, -0.00779289, 0.005502071 7, 0.001313255, -0.020094523, -0.005223527, -0.015323221, 0.025416 36, 0.00976544, 0.008717836, -0.017105905, -0.006285273, 0.013986208, 0.003173768,
0.0015369098, -0.0030639888, -0.025704736, 0.007589516, -0.023489488, -0.02755296, 0.023423947, 0.014812009, 0.015782, -0.003082012 3, 0.013706024, -0.009004502, -0.001552475 6, 0.004456711, -0.01624078, -0.02865403, 0.001979304, 0.020540195, -0.025495008, -0.004813903, -0.003145936 13, 0.004063417 7,
-0.017708872, 0.018338054, -0.008474304, 0.003896345, -0.008729910 5, 0.028024847, -0.004935152, -0.005171095 5, -0.001505778 4, -0.013828912, -0.008362886 5, -0.008664705, -0.004119 1806, 0.024013806, 0.001273 1119, -0.02689756, 0.004987584, 0.000148488 76, 0.001052734 1, -0.002718265 9, 0.017499143,
0.035024505, 0.027657824, -0.007818906, -0.024852717, 0.006737498, -0.025560549, -0.009424632, 0.025665412, -0.036807187, -0.01597862, -0.02140532, -0.013920668, -0.004357 38, -0.008808558, 0.005744569, -0.01655537, -0.002392205 2, -0.026504321, -0.019308044, 0.000141050108, 0.023935 16, 0.016057268,
-0.00976544, 0.003935669, -0.013337363, -0.004931875, 0.003798035 2, 0.016647127, 0.024131779, -0.0072749252, -0.006960334, 0.014104179, -0.006757 16, 0.013232498, 0.005092447 6, 0.021431535, -0.021444645, -0.004076579 6, -0.08032566, 0.015126601, -0.007235601, -0.010853401, 0.006658502, -0.005679029 5,
-0.000144597 33, -0.010905834, 0.010918941, 0.004810626, -0.024433263, 0.006947226, 0.006232841 4, 0.010905834, -0.018587107, -0.021693695, 0.043151405, 0.015847 54, 0.020513978, 0.013474996, -0.004672992 5, -0.000354734 5, 0.018626 43, -0.012577101, -0.028313221, 0.00601656, -0.005574165 4, 0.027395664,
-0.020513978, 0.002929632, 0.002016989 3, -0.012354265, -0.001219860 7, 0.018089004, -0.004050364, 0.014746469, 0.007464990 5, 0.020579519, 0.021156268, 0.015165925, 0.001698886, -0.007235601, 0.025429489, -0.002384012 6, -0.002415144 1, 0.007104521 6, 0.011594002, 0.00029943525, 0.010519149,
0.018141435, -0.0002881706, 0.017171444, -0.010021046, -0.029624019, -0.003103312 8, -0.024092454, 0.02230977, 0.008061403, -0.021326672, 0.000528005 3, 0.019583313, 0.024590557, 0.010551918, 0.009431187, 0.023332192, -0.00040122686, -0.019088486, 0.024878932, -0.016765097, -0.04810626,
-0.015087277, 0.001655519, -0.003591584 7, 0.008539844, 0.014812009, -0.019058993, 0.004610729 4, 0.021628156, 0.014025532, 0.026818912, -0.007753366, 0.011974134, -0.009149365, 0.014158611, 0.030305633, 0.010217665, -0.027657824, -0.003637462 6, -0.013448 78, 0.016280103, -0.013232498, 0.00591825,
0.022296862, -0.021890314, -0.0013271823, 0.00966713, 0.008847882, -0.003267 1622, 0.015965512, 0.025141092, 0.004207 6595, 0.012111767, -0.017250093, -0.04810626, -0.012754075 6, 0.008664 3705, -0.014772686, -0.021457752, -0.004698389, -0.0003748061, 0.025848923, 0.025848923, -0.004132285
0.023764756, -0.0097719944, 0.013376687, -0.019950336, -0.03848501, -0.01060435, 0.025010012, -0.005213696 5, 0.007602624 6, 0.008533291, -0.004754917, 0.015192141, 0.007641948 3, 0.007897554, -0.017643332, -0.012701626, -0.000659085 3, 0.001281642 4, 0.018862374, 0.003196707, -0.014340122, -0.019373585,
-0.005264666, 0.003706279 4, 0.008842361 8, -0.009575374, 0.017735088, -0.005626597 4, 0.000507939 6, 0.015860647, 0.018744402, 0.030279418, 0.008087619, 0.021365996, -0.018450506, -0.018036 57, 0.018941202, 0.003634185 5, 0.009935844, -0.006370475, -0.007458437, 0.001835116 2, -0.020002767, 0.026818912,
-0.0014389616, 0.000068 81686, 0.028287007, 0.002097275 8, 0.011344951, -0.008015525 5, -0.024865825, -0.012419805, 0.025809 6, 0.022781657, -0.0024642989, -0.014156611, 0.013671616, 0.018246299, -0.029824019, -0.016017944, 0.005223527, -0.047020485, -0.009037947, -0.003267 1622, 0.028234575,
0.022781657, -0.022834089, 0.027448095, -0.014222151, -0.003958608, 0.03350398, -0.012164199, -0.014720254, -0.0051121097, -0.026281485]

3.2 实际例子


Massachusetts
1
is
2
a
3
state
4
...
...
the
24
state's
25
capital
26
is
27

3.2 实际例子


Massachusetts
1
is
2
a
3
state
4
...
...
the
24
state's
25
capital
26
is
27

3.3 提示词工程

基本的预测下文。尝试“The sky is ”。

3.3 提示词工程

基本的预测下文。尝试“The sky is ”。

更加明确的指令。尝试

“Complete the sentence:

The sky is ”

3.3 提示词工程

基本的预测下文。尝试“The sky is ”。

更加明确的指令。尝试

“Complete the sentence:

The sky is ”

这种设计有效的提示词，以指导模型执行期望任务的方法被称为**提示词工程（Prompt Engineering）**。

3.3 提示词工程

提示词的格式:

<问题>?

或者

<指令>

或者

Q: <问题>?
A:

3.3 提示词工程

刚才的提示词，都是零样本（zero-shot）提示，即你直接提示模型给出一个回答，而没有提供任何关于你希望它完成的任务的示例或示范。

3.3 提示词工程

刚才的提示词，都是零样本（zero-shot）提示，即你直接提示模型给出一个回答，而没有提供任何关于你希望它完成的任务的示例或示范。

其实，不一定非要使用问答格式。例如，使用下面的提示词，可以直接得到想要的输出:

```
This is awesome! // Positive
This is bad! // Negative
Wow that movie was rad! // Positive
What a horrible show! //
```

3.3 提示词工程

刚才的提示词，都是零样本（zero-shot）提示，即你直接提示模型给出一个回答，而没有提供任何关于你希望它完成的任务的示例或示范。

其实，不一定非要使用问答格式。例如，使用下面的提示词，可以直接得到想要的输出:

```
This is awesome! // Positive
This is bad! // Negative
Wow that movie was rad! // Positive
What a horrible show! //
```

这使用了小样本（few-shot）的提示，来让模型根据上下文学习。

3.3 提示词工程

提示词要素:

- 指令: 想要模型执行的特定任务或指令。
- 上下文: 包含外部信息或额外的上下文信息，引导语言模型更好地响应。
- 输入数据: 用户输入的内容或问题。
- 输出指示: 指定输出的类型或格式。

3.3 提示词工程

为了更好地演示提示词要素，下面是一个简单的提示，旨在完成文本分类任务:

请将文本分为中性、否定或肯定
文本: 我觉得食物还可以。
情绪:

3.3 提示词工程

为了更好地演示提示词要素，下面是一个简单的提示，旨在完成文本分类任务:

请将文本分为中性、否定或肯定
文本: 我觉得食物还可以。
情绪:

在这个任务当中，指令是“将文本分类为中性、否定或肯定”，输入数据是“我认为食物还可以”部分，使用的输出指示是“情绪: ”。这个提示词没有使用上下文，也就是没有提供示例。

3.3 提示词工程

通用技巧:

- 从简单开始: 根据大模型输出的好坏，慢慢修改提示词，而不是指望一次提问就可以得到想要的答案。

3.3 提示词工程

通用技巧:

- 从简单开始: 根据大模型输出的好坏，慢慢修改提示词，而不是指望一次提问就可以得到想要的答案。
- 指令: 建议将指令放在提示的开头，同时可以使用清晰的分隔符来分割指令和上下文。例子:

```
### 指令 ###
将下面的文本翻译成西班牙语：
文本：“hello！”
```

3.3 提示词工程

- 具体性：提示越详细，越具有描述性，结果越好。提示当中的细节应该和问题是相关的，有助于完成手头的任务。

3.3 提示词工程

- 具体性: 提示越详细，越具有描述性，结果越好。提示当中的细节应该和问题是相关的，有助于完成手头的任务。

具体性的例子:

提取以下文本中的地名。

所需格式：

地点：<逗号分隔的公司名称列表>

输入: “虽然这些发展对研究人员来说是令人鼓舞的，但仍有许多谜团。里斯本未知的香帕利莫德中心的神经免疫学家 Henrique Veiga-Fernandes 说: “我们经常在大脑和我们在周围看到的效果之间有一个黑匣子。”“如果我们想在治疗背景下使用它，我们实际上需要了解机制。””

3.3 提示词工程

- 引入场景：通常来说，提示词的“具体”需要指定场景，而不是泛泛而谈。这里非常类似于有效沟通——场景越具体，信息传达得越有效。

例如，你想要了解提示工程的概念。你可以尝试这样做:

解释提示工程的概念。保持解释简短，只有几句话，不要过于描述。

3.3 提示词工程

- 引入场景：通常来说，提示词的“具体”需要指定场景，而不是泛泛而谈。这里非常类似于有效沟通——场景越具体，信息传达得越有效。

例如，你想要了解提示工程的概念。你可以尝试这样做:

解释提示工程的概念。保持解释简短，只有几句话，不要过于描述。

从上面的提示中不清楚要使用多少句子以及什么风格。更好的提示应当是非常具体、并且使用场景的。例如:

使用 2-3 句话向高中学生解释提示工程的概念。

3.3 提示词工程

- 做什么还是不做什么? 设计提示时的另一个常见技巧是避免说不要做什么，而应该说要做什么。

提示:

以下是向客户推荐电影的代理程序。不要询问兴趣。不要询问个人信息。

客户：请根据我的兴趣推荐电影。

代理：

3.3 提示词工程

更好的提示:

以下是向客户推荐电影的代理程序。代理负责从全球热门电影中推荐电影。它应该避免询问用户的偏好并避免询问个人信息。如果代理没有电影推荐，它应该回答“抱歉，今天找不到电影推荐。”。

顾客：请根据我的兴趣推荐一部电影。

客服：

3.3 提示词工程

代码提示词

- 完整的题目描述
- 指明你使用的语言
- 充分的输入输出示例
- 各个部分使用明确的分隔符分割
- （可选）如果你正在 debug，提供完整的代码
- （可选）debug 过程中，给出具体的失败案例——输入、期望的输出、实际的输出
- （可选）编译失败的情况下，给出完整的报错信息

看一个具体例子。

3.4 参考资料

1. Yjango 的科普视频: 【【渐构】万字科普 GPT4 为何会颠覆现有工作流; 为何你要关注微软 Copilot、文心一言等大模型】.[1]
2. 3blue1brown 的深度学习系列: 【GPT 是什么? 直观解释 Transformer | 深度学习第 5 章】.[2]
3. 漫士沉思录的科普视频: 【90 分钟！清华博士带你一口气搞懂人工智能和神经网络】.[3]
4. 提示词工程指南.[4]

[1] https://www.bilibili.com/video/BV1MY4y1R7EN
[2] https://www.bilibili.com/video/BV13z421U7cs
[3] https://www.bilibili.com/video/BV1atCRYsE7x
[4] https://www.promptingguide.ai/zh

3.5 参考论文

1. Vaswani, Ashish et al. “Attention is All you Need.” Neural Information Processing Systems (2017).
2. Devlin, Jacob et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” North American Chapter of the Association for Computational Linguistics (2019).
3. Radford, Alec and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training.” (2018).
4. Radford, Alec, et al. “GPT 2; Language Models are Unsupervised Multitask Learners.” 2019 by OpenAI.
5. Brown, Tom B. “Language models are few-shot learners.” arXiv preprint arXiv:2005.14165 (2020).
6. Achiam, OpenAI Josh et al. “GPT-4 Technical Report.” (2023).

Q&A

4. 总结与其他

4.1 C 语言的知识

- 基本的字符集、变量与常量、运算符
- 输入输出、调试、分支与循环
- 函数与递归
- 数组、指针、字符串、标准库
- 位运算

[1] https://www.bilibili.com/video/BV1XF411y7iJ

4.1 C 语言的知识

- 基本的字符集、变量与常量、运算符
- 输入输出、调试、分支与循环
- 函数与递归
- 数组、指针、字符串、标准库
- 位运算

对于睿信同学的 C 语言入门，可以参考 CS50 课程，其中用生动形象的方式讲解了 C 语言。对于电信同学，可以尝试做一做 CS50 的编程作业，质量很高。[1]

[1] https://www.bilibili.com/video/BV1XF411y7iJ

4.2 资源推荐

- CSDIY。自学 CS 的最好途径之一，详细介绍了许多的课程，拥有丰富的资料推荐。[1]
- Hackway。可以看作 CSDIY 的补充，介绍了六大 CS 名校（斯坦福、MIT、UCB、普林斯顿、哈佛、CMU）的本科培养路线，可以参考。[2]
- CS50P 或者 CS61A。Python 入门课程，其中 CS50P 更偏向实用，CS61A 更偏向理论，二者都很好。[3][4]
- 我的博客《信科大一通关指南》。介绍了心态、方法，也有每门课程的简评。电信的课程差别不大，都可以参考。[5]

---

[1] https://csdiy.wiki/

[2] https://hackway.org/docs/cs/intro

[3] https://csdiy.wiki/%E7%BC%96%E7%A8%8B%E5%85%A5%E9%97%A8/Python/CS50P/

[4] https://csdiy.wiki/%E7%BC%96%E7%A8%8B%E5%85%A5%E9%97%A8/Python/CS61A/

[5] https://ovideros.github.io/p/%E4%BF%A1%E7%A7%91%E5%A4%A7%E4%B8%80%E9%80%9A%E5%85%B3%E6%8C%87%E5%8D%97/

4.3 实用的工具

- 如果你想要保存代码的每一个版本，还想和别人协作写代码——**Git**、**Github**、**Gitee**

4.3 实用的工具

- 如果你想要保存代码的每一个版本，还想和别人协作写代码——**Git、Github、Gitee**
- 如果你想要使用远程的服务器，来运行一些很复杂的运算（通常是深度学习）——**了解 Linux、CLI、SSH**

4.3 实用的工具

- 如果你想要保存代码的每一个版本，还想和别人协作写代码——**Git、Github、Gitee**
- 如果你想要使用远程的服务器，来运行一些很复杂的运算（通常是深度学习）——**了解 Linux、CLI、SSH**
- 如果你觉得 Word 很难用，希望使用更加格式化的工具来写论文、写报告、做 PPT——**Markdown、LaTeX、Typst（Typst 拥有 LaTeX 的排版能力，同时拥有 Markdown 一样简洁的语法，该演示文稿使用 Typst 编写）**

4.3 实用的工具

- 如果你想要保存代码的每一个版本，还想和别人协作写代码——**Git、Github、Gitee**
- 如果你想要使用远程的服务器，来运行一些很复杂的运算（通常是深度学习）——**了解 Linux、CLI、SSH**
- 如果你觉得 Word 很难用，希望使用更加格式化的工具来写论文、写报告、做 PPT——**Markdown、LaTeX、Typst（Typst 拥有 LaTeX 的排版能力，同时拥有 Markdown 一样简洁的语法，该演示文稿使用 Typst 编写）**
- 如果你觉得 C 语言太麻烦，想要分析数据、画图——**使用 Python 以及相关的库，例如 Numpy、Matplotlib、Pandas、Seaborn 等等**
- 如果你觉得 Dev C++太丑，或者不够智能，可以使用更先进的 IDE——**Vscode、Cursor（内部集成了 AI）**

4.3 实用的工具

- 如果你想要自定义 C 语言的编译过程，或者需要编写拥有很多文件的大型项目——使用 **GNU Make 等等**

4.3 实用的工具

- 如果你想要自定义 C 语言的编译过程，或者需要编写拥有很多文件的大型项目——使用 **GNU Make** 等等
- 如果你想要匹配某个文本的格式——正则表达式

4.3 实用的工具

- 如果你想要自定义 C 语言的编译过程，或者需要编写拥有很多文件的大型项目——使用 **GNU Make** 等等
- 如果你想要匹配某个文本的格式——正则表达式
- 如果你想要做一个网页——了解一些前端开发，例如 **html**、**css**、**javascript**

4.3 实用的工具

- 如果你想要自定义 C 语言的编译过程，或者需要编写拥有很多文件的大型项目——使用 **GNU Make 等等**
- 如果你想要匹配某个文本的格式——正则表达式
- 如果你想要做一个网页——了解一些前端开发，例如 **html、css、javascript**
- 如果你觉得配置环境很麻烦，使用虚拟环境——**Anaconda（Python 相关）、Docker（万能的环境）**

4.3 实用的工具

- 如果你想要自定义 C 语言的编译过程，或者需要编写拥有很多文件的大型项目——使用 **GNU Make 等等**
- 如果你想要匹配某个文本的格式——**正则表达式**
- 如果你想要做一个网页——**了解一些前端开发，例如 html、css、javascript**
- 如果你觉得配置环境很麻烦，使用虚拟环境——**Anaconda（Python 相关）、Docker（万能的环境）**
- 如果你不想使用鼠标，只使用键盘编辑代码——**Vim、Emacs**

4.3 实用的工具

- 如果你想要自定义 C 语言的编译过程，或者需要编写拥有很多文件的大型项目——使用 **GNU Make 等等**
- 如果你想要匹配某个文本的格式——**正则表达式**
- 如果你想要做一个网页——**了解一些前端开发，例如 html、css、javascript**
- 如果你觉得配置环境很麻烦，使用虚拟环境——**Anaconda（Python 相关）、Docker（万能的环境）**
- 如果你不想使用鼠标，只使用键盘编辑代码——**Vim、Emacs**
- 如果你希望在 Windows 上面运行 Linux——**虚拟机、WSL**

4.3 实用的工具

- 如果你希望查找报错——**Bing、Google、StackOverflow、Github issues 等等**

[1] https://csbaoyan.top/

[2] https://csdiy.wiki/%E5%BF%85%E5%AD%A6%E5%B7%A5%E5%85%B7/tools

4.3 实用的工具

- 如果你希望查找报错——**Bing、Google、StackOverflow、Github issues 等等**
- 如果你希望了解 CS 方向的保研信息——**强烈推荐 CS BAOYAN[1]，知乎、小红书上也会有零散的经验贴**

[1] https://csbaoyan.top/

[2] https://csdiy.wiki/%E5%BF%85%E5%AD%A6%E5%B7%A5%E5%85%B7/tools

4.3 实用的工具

- 如果你希望查找报错——**Bing、Google、StackOverflow、Github issues 等等**
- 如果你希望了解 CS 方向的保研信息——**强烈推荐 CS BAOYAN[1]，知乎、小红书上也会有零散的经验贴**
- 如果你希望了解 AI 领域的最新研究进展，可以看中文的**“三大会”**公众号——**机器之心、量子位、新智元**

[1] https://csbaoyan.top/

[2] https://csdiy.wiki/%E5%BF%85%E5%AD%A6%E5%B7%A5%E5%85%B7/tools

4.3 实用的工具

- 如果你希望查找报错——**Bing、Google、StackOverflow、Github issues 等等**
- 如果你希望了解 CS 方向的保研信息——**强烈推荐 CS BAOYAN[1]，知乎、小红书上也会有零散的经验贴**
- 如果你希望了解 AI 领域的最新研究进展，可以看中文的**“三大会”**公众号——**机器之心、量子位、新智元**
- 更多可以参考 CSDIY 上面的必学工具与实用工具箱[2]

[1] https://csbaoyan.top/

[2] https://csdiy.wiki/%E5%BF%85%E5%AD%A6%E5%B7%A5%E5%85%B7/tools

Q&A

THANKS!

结束后有问题也可以当面问我。