机器要怎么做?

Traditional Language Model

$$P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2)\ldots P(w_n \mid w_1, w_2, \ldots, w_{n-1}) = \prod_i P(w_i \mid w_1, w_2, \ldots, w_{i-1})$$

$$P(w_{next} \mid \text{圆圆，爱，吃}) = \frac{count(w_{next}, \text{圆圆，爱，吃})}{count(\text{圆圆，爱，吃})}$$

当需要预测上下文圆圆爱吃 ___ 后面跟的词时，如果词典中所有单词为圆圆，爱，吃，瓜，梨时

P(瓜|圆圆，爱，吃) > P(梨|圆圆，爱，吃)

通过计算每个词在句子中出现的概率，判断词与句子的关系

优点：比较简单直观简单，容易理解
缺点：依赖整个句子作为上下文，处理长文本时比较耗时。并且，由于模型只是基于历史语料进行概率计算。预测精度完全依赖训练模型的语料库大小。并且，如果需要预测的词在词典中没有出现过时，就无法正确预测

为了解决当词典V较大，模型训练效率低的情况时，人们引入马尔科夫链对上诉算法进行优化，也就是所谓的N-Gram算法
简单来说，就是在计算第i个词的概率时，不再参考整个语句只参考当前次前k个词即可。

机器要怎么做?

圆圆爱吃瓜

如何让机器认识一句话中的每个词

```
圆圆 -> (1, 0, 0, 0)
爱 ->  (0, 1, 0, 0)
吃 ->  (0, 0, 1, 0)
瓜 ->  (0, 0, 0, 1)
```

Vocabulary_Size = 4

编码特点:

1. 词典大小固定
2. 损失位置信息

ONE-HOT Encoding

	圆圆	爱	吃	瓜
圆圆	1	0	0	0
爱	0	1	0	0
吃	0	0	1	0
瓜	0	0	0	1