# 本地知识库实现的相关知识

## 分词

先分词，中文可以使用jieba分词



## 构造数据集

将词汇向量化是自然语言处理中的重要任务，它可以将文本数据转化为计算机能够理解和处理的向量形式。以下是一些常用的词汇向量化方法：

One-Hot 编码： 这是最简单的向量化方法之一。对于每个词汇，创建一个由 0 和 1 组成的向量，其中向量的长度等于词汇表的大小，而对应词汇的位置置为 1，其他位置为 0。这种方法简单，但无法捕捉词汇之间的语义关系。

词嵌入（Word Embeddings）： 词嵌入是一种将词汇映射到连续向量空间的方法，能够捕捉词汇之间的语义关系。Word2Vec、GloVe 和 FastText 等模型可以生成这种词嵌入。这些向量表示可以通过预训练模型获取，也可以在特定任务上进行微调。

BERT（Bidirectional Encoder Representations from Transformers）： BERT 是一种预训练的语言模型，它能够生成上下文感知的词汇表示。你可以使用 BERT 模型或其变体来为词汇生成向量，这些向量会考虑词汇在上下文中的含义。

Transformer 编码器输出： 如果你有一个训练有素的 Transformer 编码器（如 BERT、GPT 等），你可以将文本输入编码器，然后使用编码器的输出作为文本的向量表示。

如果目标词汇是 "apple"，上下文是 ["juicy", "fruit"]，那么训练样本可以是 ("apple", "juicy") 和 ("apple", "fruit")。

## 利用神经网络训练词向量

在使用word2vec将词向量传入神经网络训练的时候，词向量会发生改变，神经网络的权重也会发生改变。

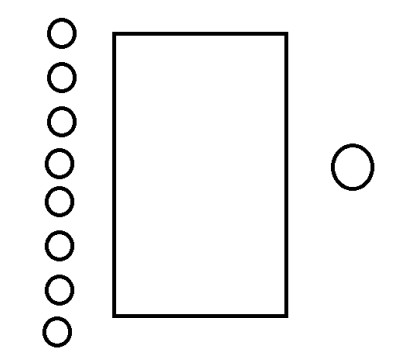
使用Word2Vec预训练的词向量作为初始输入，然后将它们传入神经网络进行进一步的训练，通常会涉及两个部分的权重更新：词向量权重和神经网络权重。

词向量权重更新： 当使用预训练的Word2Vec词向量作为初始输入时，这些词向量本身可以被看作是一种初始的表示。在训练过程中，这些词向量也会发生微小的调整，以更好地适应特定任务。这种微小的调整有时被称为“微调”。这意味着，预训练的词向量可能会在训练期间进行轻微的优化，以更好地捕捉语义信息和任务特定的上下文。

神经网络权重更新： 神经网络的权重包括各种层（例如，全连接层、卷积层等）的权重以及可能的偏置项。当将预训练的词向量传入神经网络进行训练时，神经网络的权重会在反向传播的过程中进行更新，以使网络能够适应特定的任务。这些权重的更新取决于损失函数、优化算法（如随机梯度下降或其变体）、学习率等。

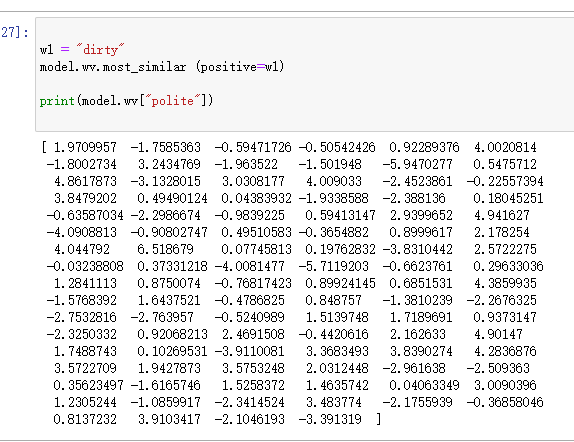
词向量分别是4个，假设a和b相似度为0.8，0.8是人为定义的，假设a=[0.2, 0.3, 0.4, 0.5],b=[0.1,0.9,0.6,0.5]

那数据集应该是[[0.2, 0.3, 0.4, 0.5],[0.1,0.9,0.6,0.5],0.8]，所以我们的神经网络应该如下：



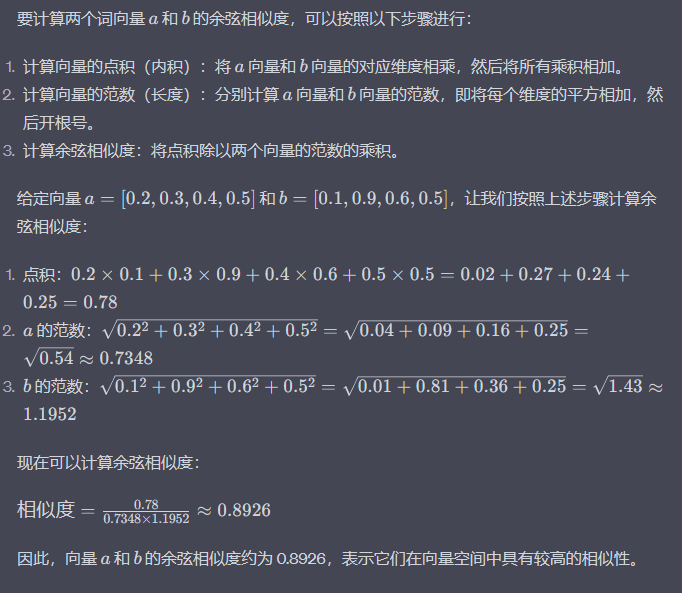
训练好的词向量

打印词汇表中的dirty对应的向量



## 怎么求两个词向量的相似度

假如有两个词向量，a=[0.2, 0.3, 0.4, 0.5],b=[0.1,0.9,0.6,0.5]



## 词向量通过什么方法组成段落向量

词向量 通过一定算法组成 句向量

句向量 通过一定算法组成 段落向量

段落向量 通过一定算法组成 文章向量

每一步的算法都可以使用如下算法：

1、求平均值或加权平均值： 可以对段落中的所有句子向量进行平均或加权平均，得到段落向量。这是一种简单的方法，但可能无法很好地处理句子之间的关系。

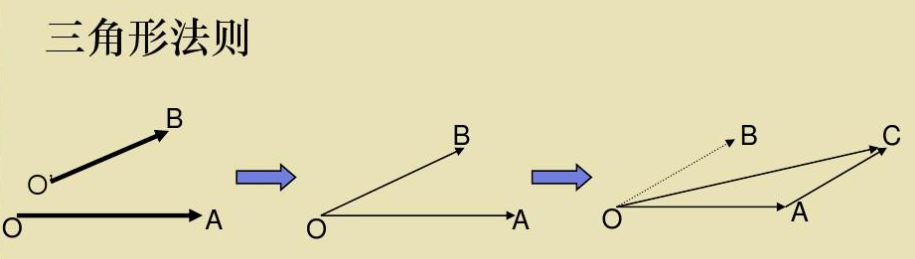
2、循环神经网络 (RNN) 或 Transformer 编码器： 类似于生成句子向量的方式，可以使用RNN或Transformer编码器来处理句子向量序列，生成一个表示整个段落的向量。

3、层次化注意力： 可以使用注意力机制来捕捉段落内句子之间的关系。例如，可以使用自注意力机制来为句子赋予不同的权重，然后加权平均句子向量以生成段落向量。

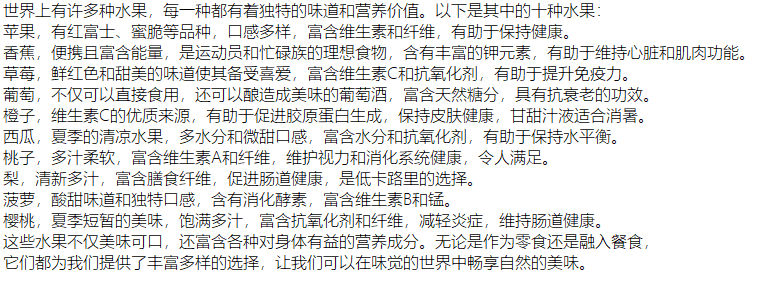
4、递归神经网络 (Recursive Neural Network)： 这种神经网络结构允许在树状结构中捕捉句子之间的层次关系，从而生成段落向量。

5、文档级预训练模型： 一些预训练模型（如DocBERT、Longformer等）专门设计用于生成文档级别的嵌入。这些模型可以直接用于生成段落向量。

上面提到的算法当中，我举例一个最简单的算法，就是直接把向量相加

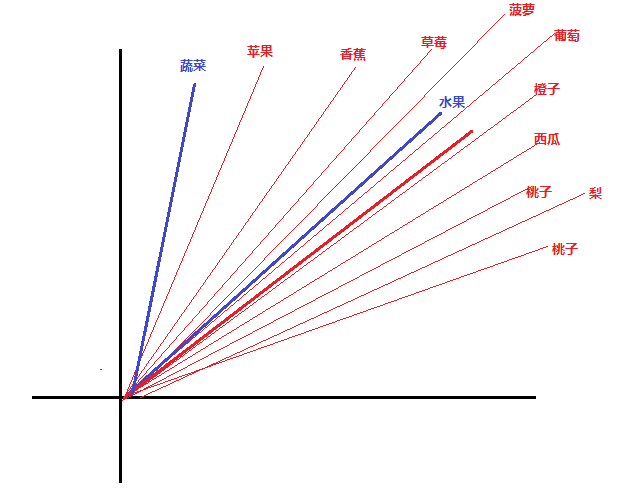


例如一篇文章如下



假设我们已经拥有了所有水果词汇的向量，我们可以对文章进行分类，分类的简单方式如下

假设将所有红色向量加起来，得到了一个粗红色的向量，我们将红色的向量在我们的分类库里面查找，找到最相似分类红色粗线，水果分类是蓝色粗线，根据余弦相似度计算红色粗线和蓝色粗线，证明我们这篇文章是关于水果的，而不是蔬菜的。



上面的通过关键词的向量相加，能够提取的信息只是一篇文章的相关关键词的数量信息，但是无法提取一篇文章的关键词的顺序信息。假设有向量a，b，c，他们之间相加的先后顺序不影响最终结果，如a + b + c = b + a + c = c + a + b。所以假设有两篇文章，一篇讲的是男生喜欢女生的情感心理活动，一篇讲的是女生喜欢男生的情感心理活动。关键词“男生”“喜欢”“女生”，男生喜欢女生 和 女生喜欢男生 是不一样的，所以这种相加方式无法区分出这种顺序特征，所以可以利用一些更加复杂的算法是生成，复杂的算法代表了更加耗费训练时间。

## Word Embedding 是什么？

数据的embedding，可以通过很多种方式去实现。

下面是一些可以本地embedding且开源的方法：

**免费：**

1、word2vec

2、DeepWalk 算法

3、Node2vec

4、EGES

**收费：**

1、text-embedding-ada-002：OpenAI 出品的 text-embedding-ada-002 模型被广泛使用，虽然该模型的效果较好，但此模型不开源、也不免费，所以它也是一个http的形式开放出来。

Word2vec相对简单，通过了解Word2vec，进而了解Embedding是什么，演示

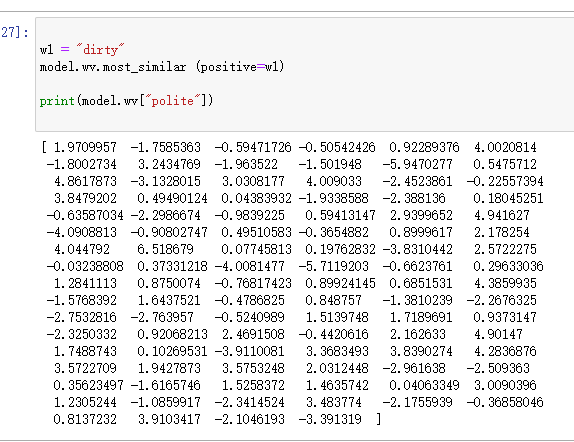
## Anaconda工具

允许片段执行python代码，有利于我们调整我们的模型参数。

## 什么是向量数据库？与普通数据库的区别？

有很多种向量类型，嵌入向量（Embedding Vectors）特征向量（Feature Vectors）词向量（Word Vectors）像特征向量（Image Feature Vectors）音频特征向量（Audio Feature Vectors）

但是所有的向量类型都是数字数组，如下图



也就是说向量数据库存储着大量这样的数字，每一词对应一个数组，一万个词就有一万个这样的数组，这就构成了一个数据字典，当我们输入一个词向量的时候，向量数据库可以通过各种方式找到最相似的向量，传统数据库没有提供这样子的搜索能力，需要我们认为去实现。

传统的数据库不适合用来存储和检索向量数据，传统数据是依据sql语句和关键词精准搜索不同，搜索过程就是查询出和查询向量最为相似的一些向量，具有一定的模糊性。

向量数据库提供了很多自带的api，用来应对不同数据量级的搜索场景。如穷举所有向量，提供不同的算法聚类向量形成码本或者进一步形成子码本等等功能，还有一个更加先进的HNSW算法。但是无论使用哪种算法，都是在三个方面做出权衡，**时间，空间，搜索质量**。

如某一种算法时间快，但是存在占用内存大和搜索质量下降的问题。搜索质量说的是依据这种算法查出来的向量可能不是最优结果。