Word Embedding

🎉 当前，我们已经完成了分词任务，即将一段文本划分成了若干个最小的单元组成的序列，也就是token，并且每个token我们都给了一个ID（通过维护了一个vocab完成token到ID之间的相互映射）

vocab

什么是 Word Embedding?

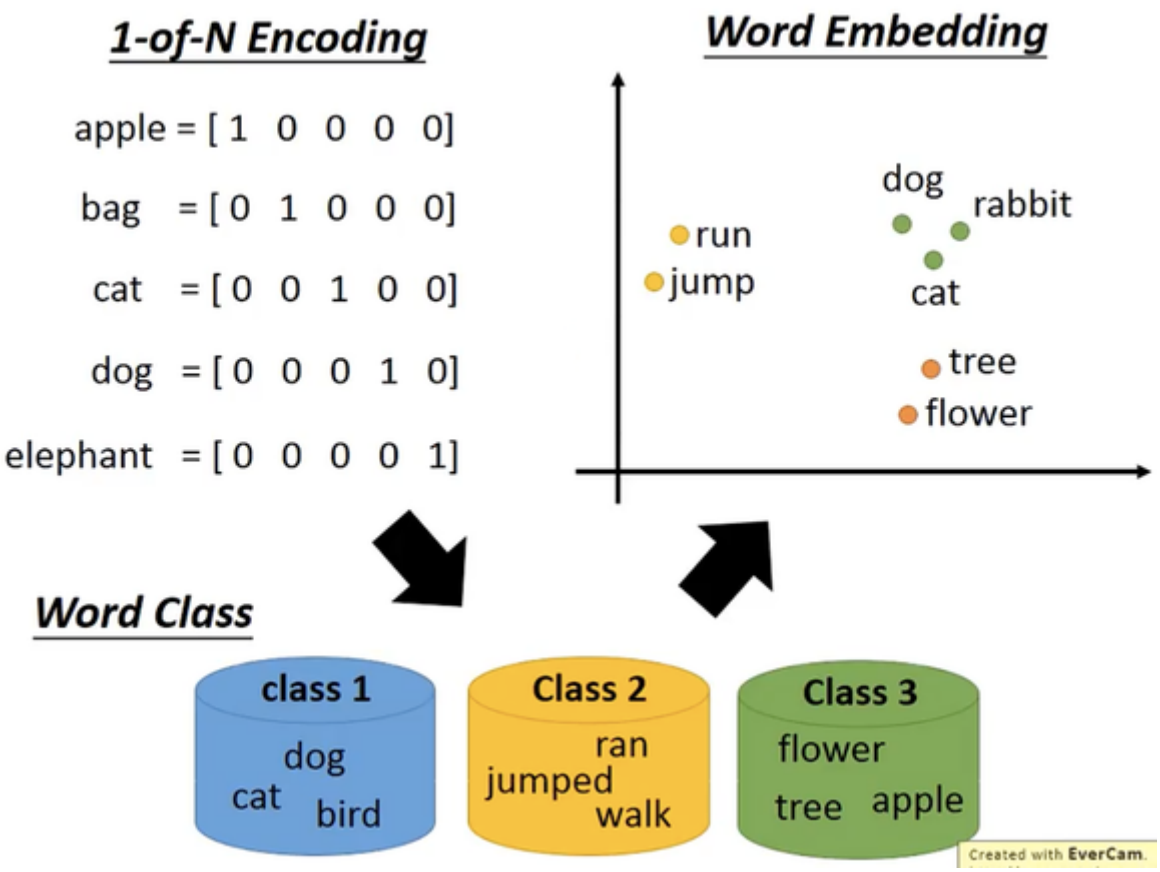
Word Embedding 就是把文字转换成数字的形式，这样计算机就能更容易地理解和处理这些文字。我们要在得到每个token ID的情况下，进一步得到能输入神经网络的数字表示。

举个例子

假设我们有两个词：“猫”和“狗”。通过某种算法，我们得到了它们的词向量：

- “猫” -> `[0.1, 0.8, 0.5, ...]`
- “狗” -> `[0.2, 0.7, 0.4, ...]`

你会发现这两个词的数字序列很接近，这反映了“猫”和“狗”在很多情况下是类似的。

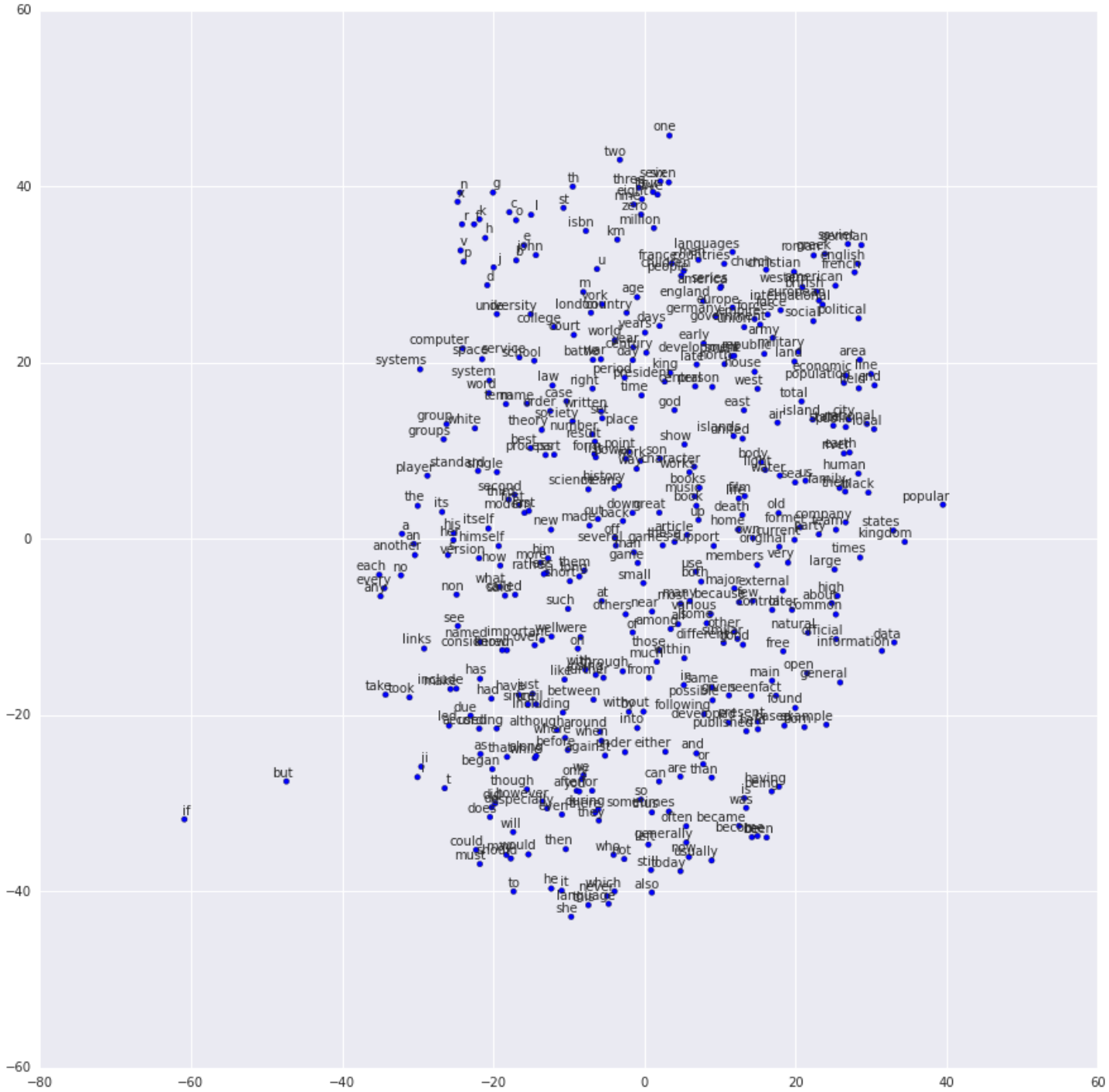


词向量

词向量就是某一个多维向量能唯一代表某一个词。

举个例子：假设中文只有4个词【你、我、他、它】，如果用0或1去表示他们，【你】可以表示为(1,0,0,0)，【我】可以表示为(0,1,0,0)，以此类推。
这些0或1组成的向量就可以称之为词向量，但是这种模式的词向量并不好，主要原因有两点：第一中文中有非常多的词，如果每一个词就占一位，那么这样的词向量就会非常长，
同时如果添加了新的词汇，向量的维度就需要增加。第二这些词向量永远都是有一个唯一的1，其余都是0，这样的词向量只能代表这个词，但是不能代表这个词的意思，每个向量之间都是正交。
所以以上述onehot方式编码的向量，并没有什么意义。

为了解决上述这个问题呢，Mikolov就提出了一种模型叫Word Embeddings，这种模型可以把词投射到一个固定的多维空间，
每个词的词向量都是同样的长度且相近语义的词会聚集在一起。



📌 一句话总结Embedding：

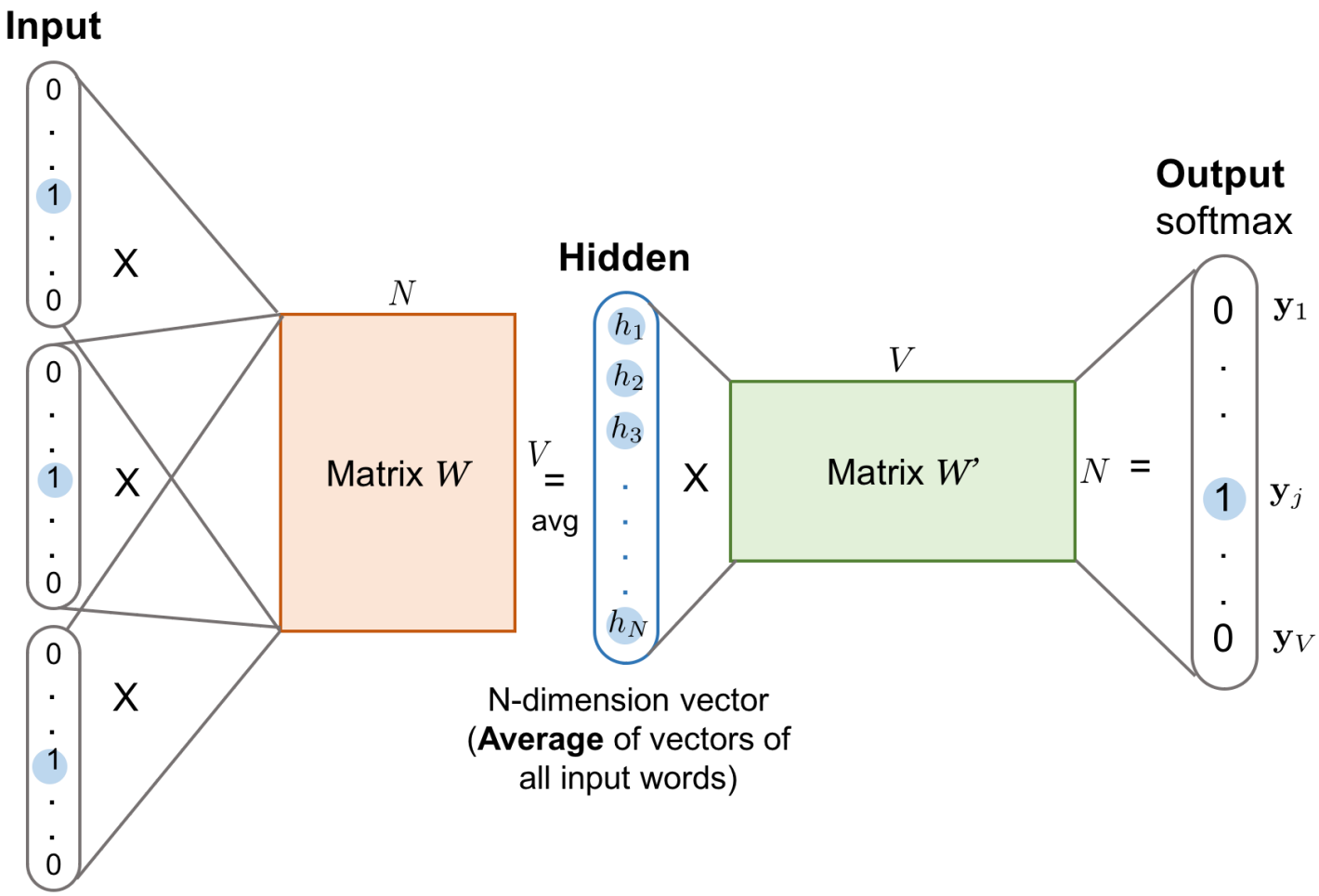
Embedding就是把token转换为向量的过程

如何实现

实现这种模型，又分两种主流的方法，一个叫Continuous Bag-of-Words model(CBOW)，一个叫skip-gram。

CBOW

CBOW认为一个词的语义可以通过其周边的词来表示，所以通过周边的词来预测被选词。



这张图其实完整的描述了cbow的全部过程，但是并不好理解，所以我用一个例子一步一步解释这张图

假设在世界上只有一个句子 `I love you`，也就是说只有三个单词（用大写的V表示，V=3），通过onehot编码的方式，`I` 可以表示为 `(1,0,0)`，`love` 可以表示为 `(0,1,0)`，`you` 可以表示为 `(0,0,1)`。

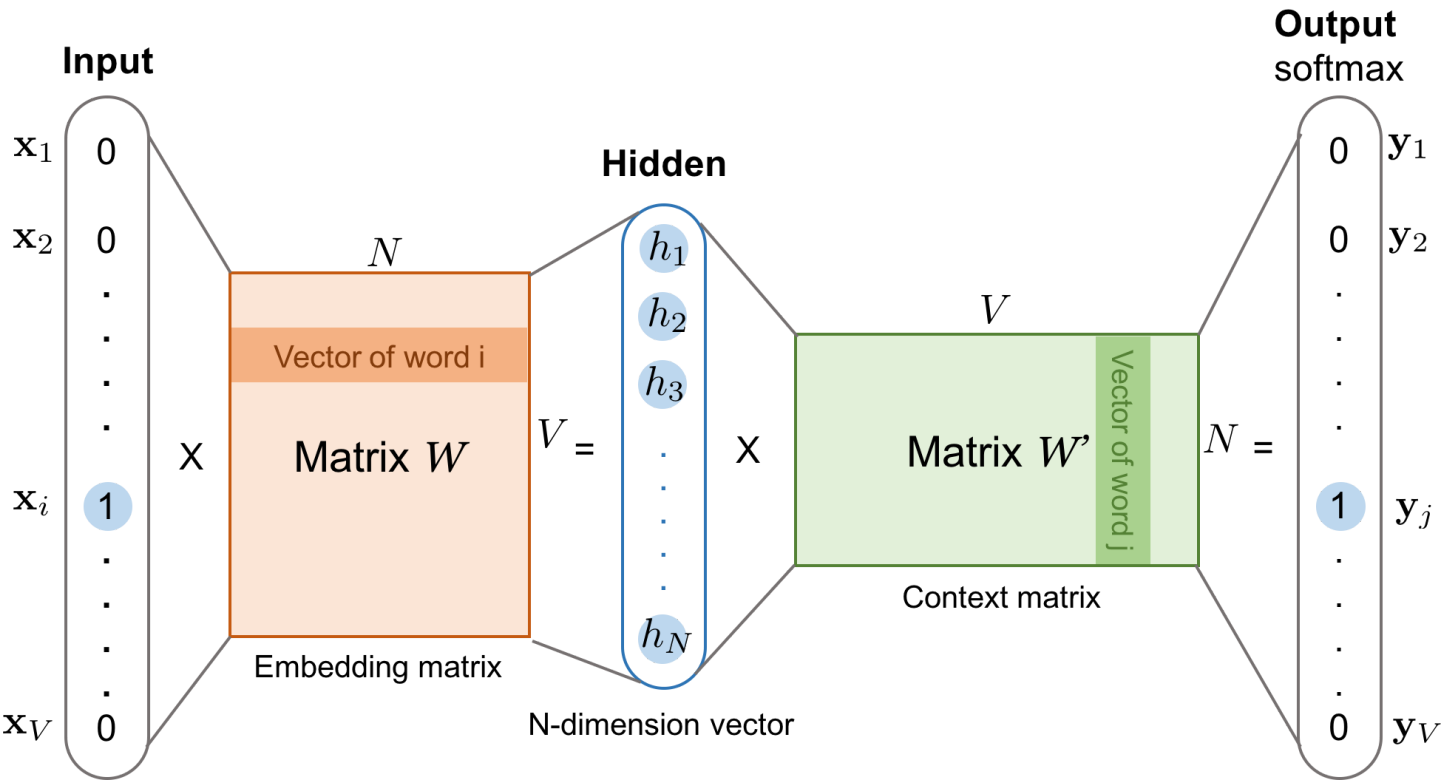
cbow用周边n个词预测中间的词，假设n=1，我们的模型要通过 `(I,you)` 预测出 `love`。

1. 将 `I(1,0,0)` 和 `you(0,0,1)` 输入输入层，由于n=1，所以输入层由两个神经元组成。
2. 每个输入向量乘以一个V*N的矩阵，V是上面说的词库的大小，在这里就是3，N则是隐藏层神经元的个数。
3. 因为输入向量是一个1*V的矩阵，所以相乘后的结果是一个1*N的矩阵。
4. 把所有的输入向量产生的乘积累加之后取平均值得到一个新的1*N的矩阵就是我们在隐藏层的输出。
5. 把这个输出的向量再乘以一个N*V的矩阵(与上述1*N的矩阵没有任何关系)，又会得到一个1*V的矩阵，可以把这一层叫做投影层，这层的作用就是把隐藏层的结果再映射到词库里。
6. 我们把最终得到的1*V的矩阵通过softmax得到矩阵中最大概率的那个列，而这一列就应该对应love这个单词，如果不是，说明我们的隐藏层和投影层的矩阵并不符合要求。

7. 我们通过loss function（损失函数）以及反向传播算法去不断调整这两个矩阵的参数，直到其到达一个我们相对满意的程度。这就是训练模型的过程。
8. 当这个模型达到一定的准确率后，我们通过每个词的onehot编码乘以隐藏层的矩阵得到的1*N矩阵，就是我们要得到的这个词的词向量了，所以隐藏层神经元的个数就决定了词向量的维度。

Skip-gram

不同于CBOW，Skip-Gram模型通过选中词预测周围n个词。



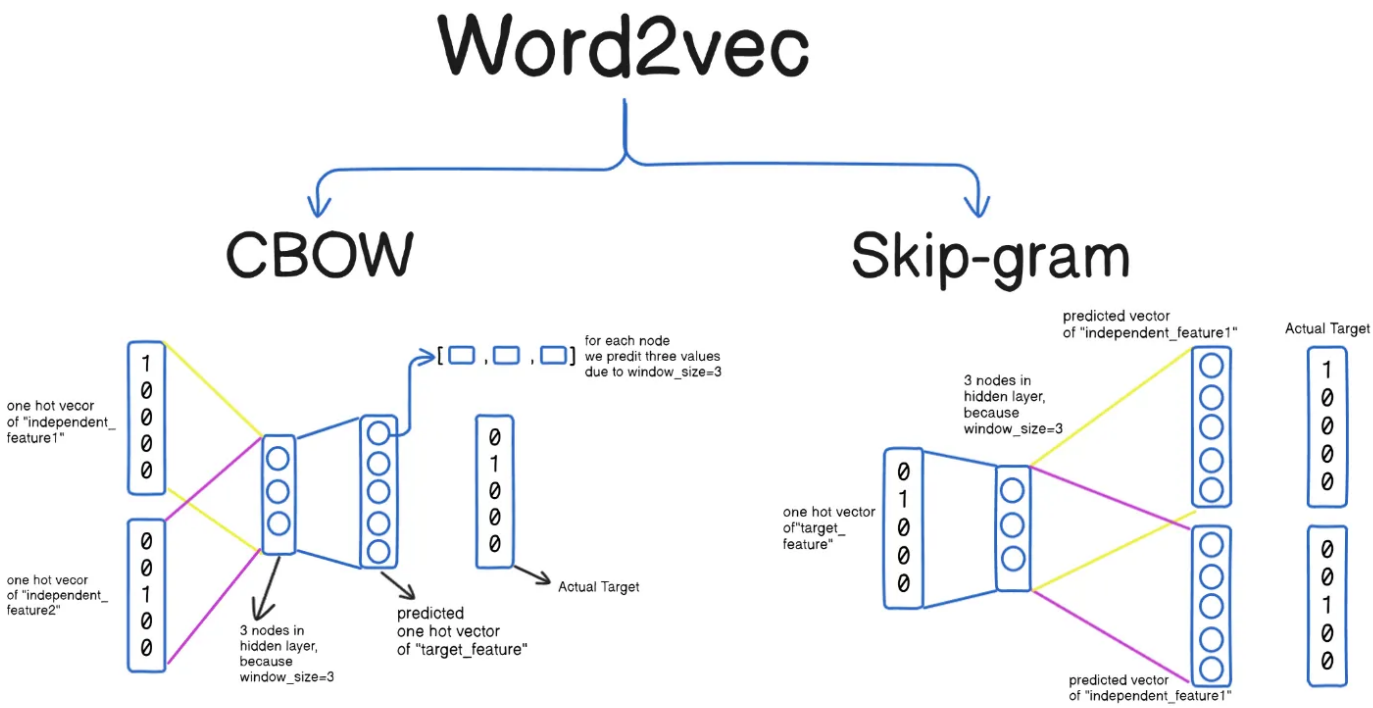
还是按照CBOW的例子，在Skip-Gram模型下，当n=1时，我们的模型需要通过love预测出I和you。

Skip-Gram的训练过程基本与CBOW相同，唯一的区别就是CBOW需要输入向量是多个，需要把隐藏层的结果累加取平均值，
而Skip-Gram则不需要，直接计算就好。

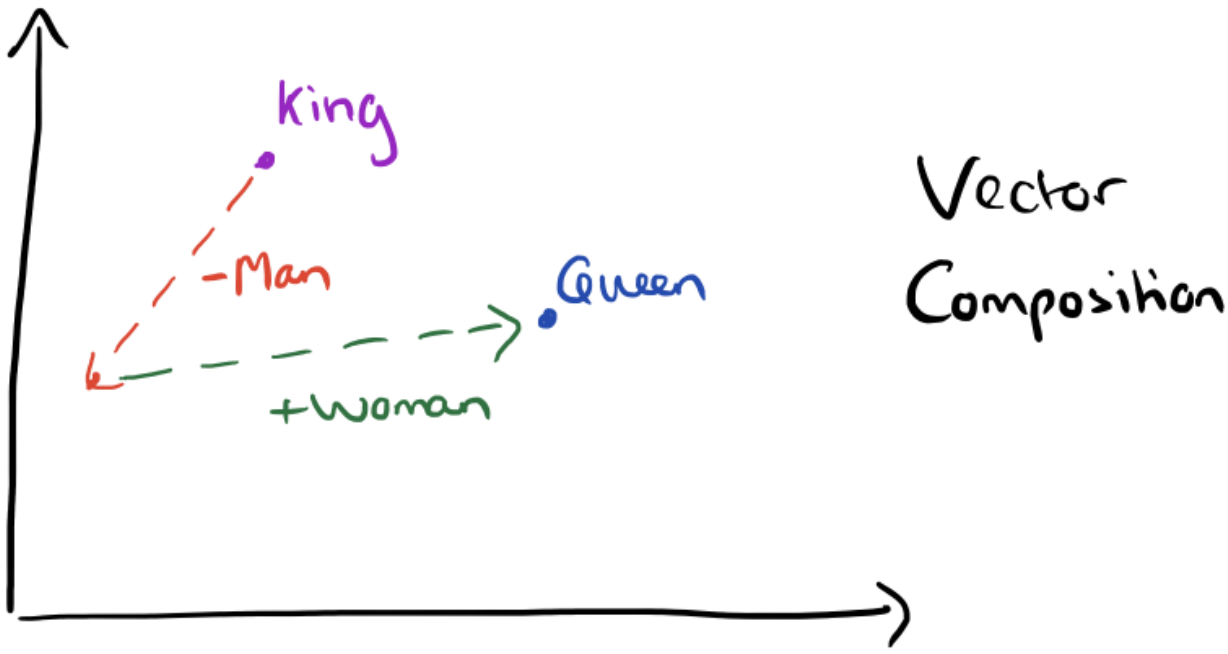
Word2Vec

Word2Vec是一个由 Google 在 2013 年推出。Word2Vec 的核心思想是通过上下文来预测单词，或者通过单词来预测上下文，从而生成词向量。这些词向量能够捕捉词与词之间的语义和语法关系。

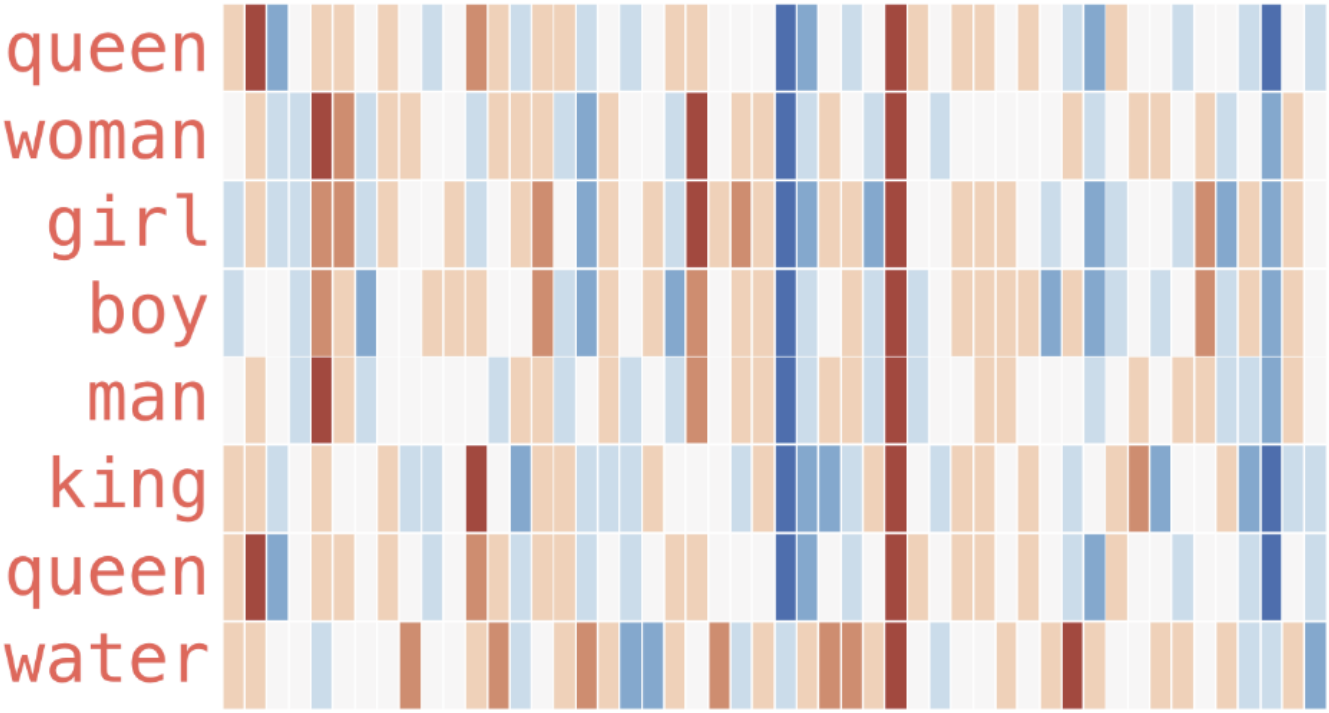
其技术原理就是在上述两种模型的模式下，进行大量的文本得到的预训练好的模型矩阵

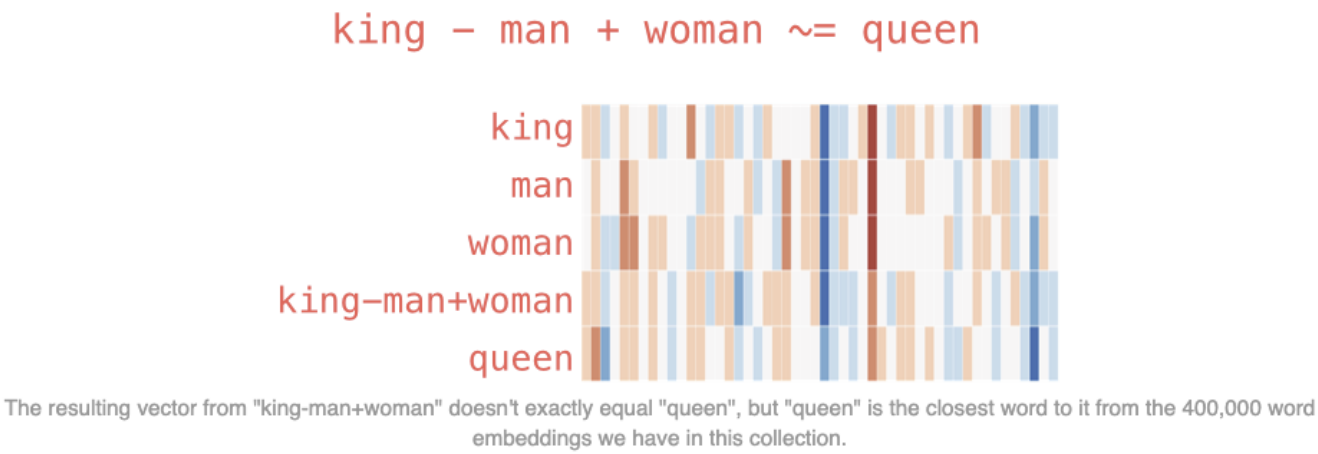


word2vec技术架构



embedding表示的语义相似性





The resulting vector from "king-man+woman" doesn't exactly equal "queen", but "queen" is the closest word to it from the 400,000 word embeddings we have in this collection.

思考一个问题

word2vec这种预训练的词向量表示有什么缺点呢?

参考资料

https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

https://medium.com/@fraidoonomarzai99/word2vec-cbow-skip-gram-in-depth-88d9cc340a50

https://jalammar.github.io/illustrated-word2vec/🌟🌟🌟推荐阅读