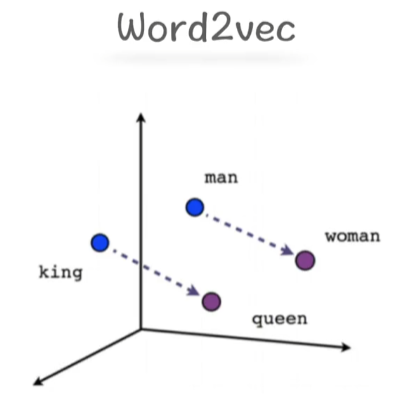
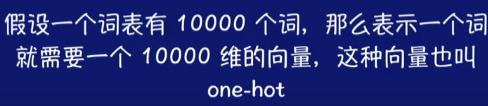
Word2Vec，是一种将单词转化成向量的方法

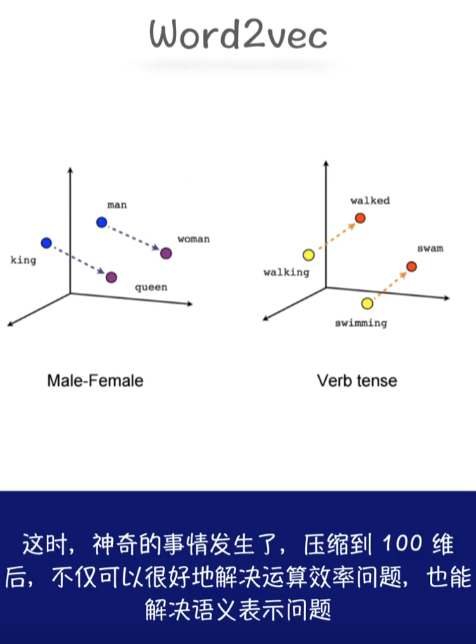
什么向量呢，是一种稠密的向量，也是一种语义向量

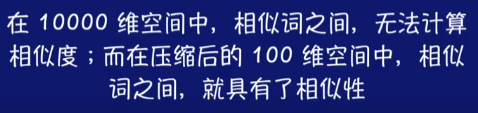


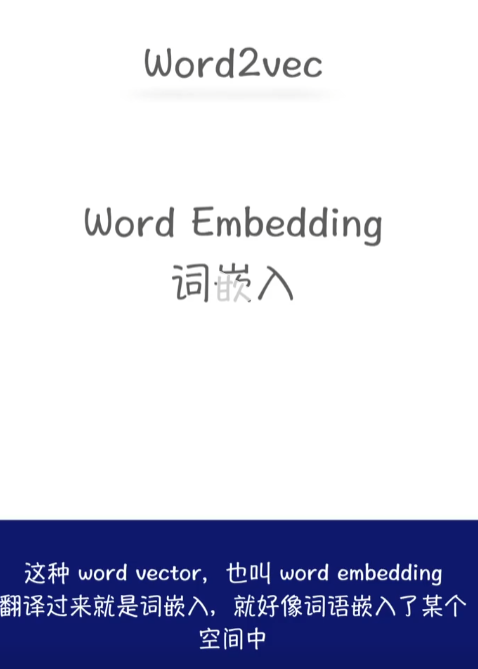
为什么稠密，如下：

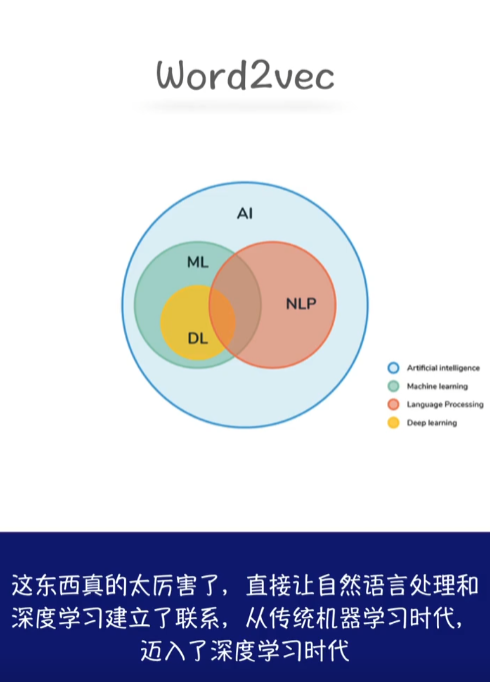




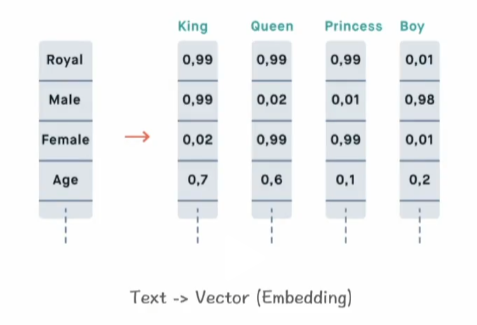




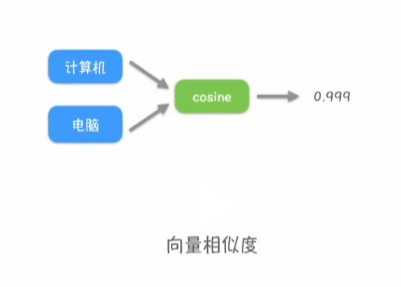




word embedding将文字这种不可计算的数据表示成了可计算的数据，并赋予数据以语义，在这种有语义的词向量中，但这种有语义的词向量中有什么用呢？



向量之中的cosine就代表相似度



也可以对很多次聚类得到向量中心，或者将句子中的词向量加起来，表示成句向量，就能得到句子的语义。



通汇贯通词向量这种技术，那么我们就可以做到万物皆可embedding。

1. 通过词向量技术，我们知道一个单词在每一个维度都有一个具体的值，训练之后的词向量不会过于稀疏。
2. 语言也是以数字来存储的，在计算机当中，本来就有一种叫unicode的编码来表示每个词汇，但是这种方式没办法表达词汇与其他词汇之间的关系，就好比说我们有一个地图。里面有香港中文大学，北京故宫广州塔等等各个地方，我们通过一张地图就能清楚不同地方之间的距离，通过这些距离找到他们的相似性，所以我们也有这么一张词表，类比于现实中的地图，当我们说两个词汇的时候，我们也能够知道这两个词汇的远近，也就是他们的相似性，这样我们就能更方便的去查看，每个词汇之间的关系。
3. 例如，如果一个单词在一些上下文中经常与另一个单词一起出现，那么这两个单词的嵌入向量在向量空间中就会有相似的位置。这意味着它们有相似的含义和语义。
4. embedding重要的原因在于它可以表示单词或者语句的语义。

One-hot的维度由词库的大小决定

我们用余弦相似度来计算计算任意两个词语之间的相关性，结果都会是0，经过以上问题我们学会用词向量来表示词语，具体表示为一串数字，他可以取任意实数，这样一来有限的维度就是表示无数的词语，大家可以用排列组合的思维来思考一下，12维的One-hot只可以表示12个词语，12维的词向量可以表示无数个词语，纳表示词向量这个数字是怎么得来的呢？

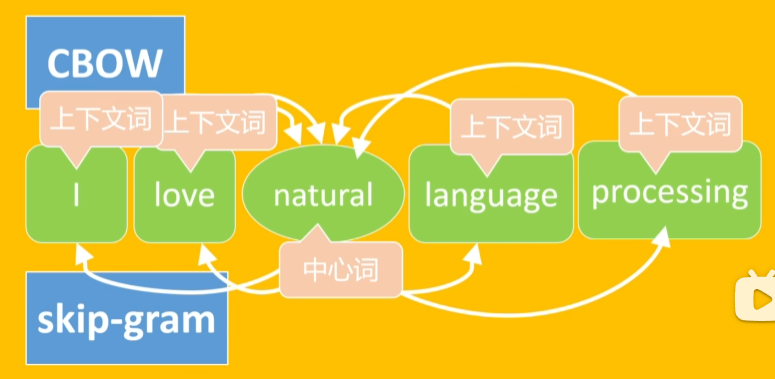


Word2vec有一个重要的假设，他的算法也是基于这个假设来设计算法的



Skip-gram依据中心词来预测上下文词（一般用这个）

CBOW依据上下文词来推测中心词汇，



关键词：中心词，上下文词。针对某个中心词语，它的哪些词语算是它的上下文词呢，

所以我们需要指定窗口大小，中心词前后都是上下文词，最大化中心词语与上下文词语共同出现的概率，这样我们就能得到词向量，例如当窗口大小为2的时候，

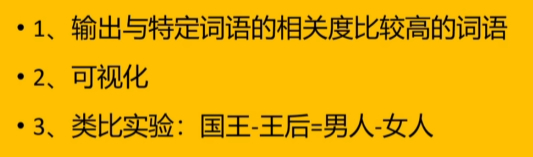






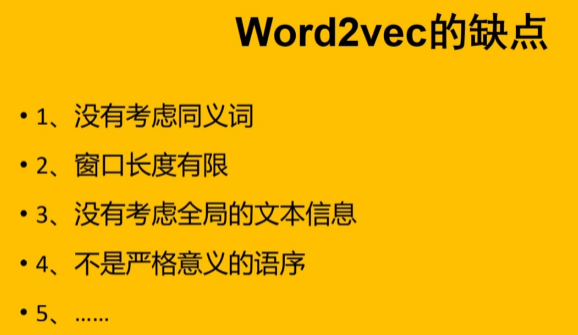
当语料库足够大的时候，非上下文字太大了，会导致计算灾难，所以我们只采用部分样本，

词向量表示为词语具体的表示为一串数字，这些数字到底是什么意思，它的效果好不好，我们人类是难以观察出来的，我们可以通过某些方法去验证一下。

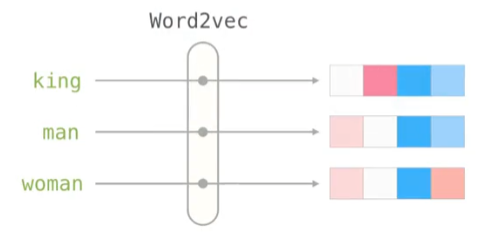


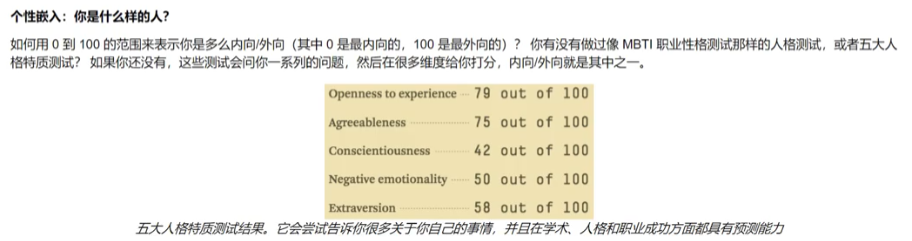
那么训练好词向量之后我们可以做什么？



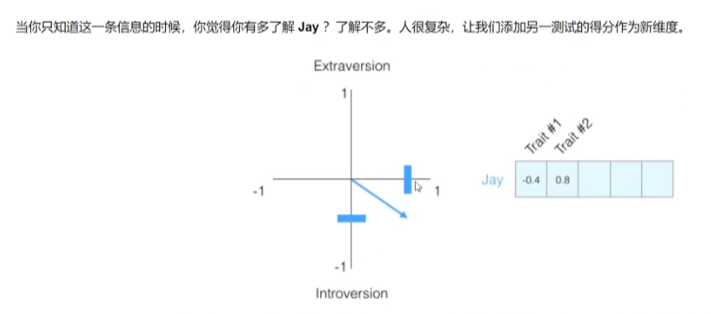


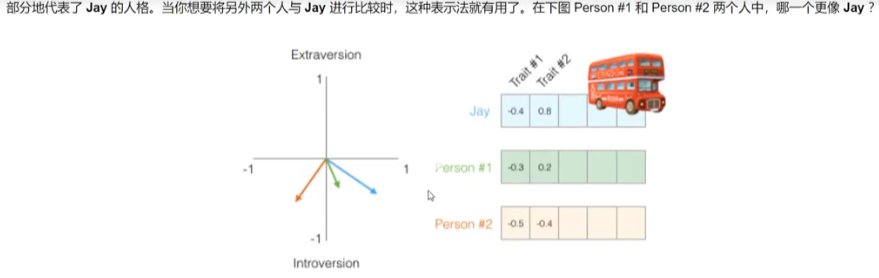




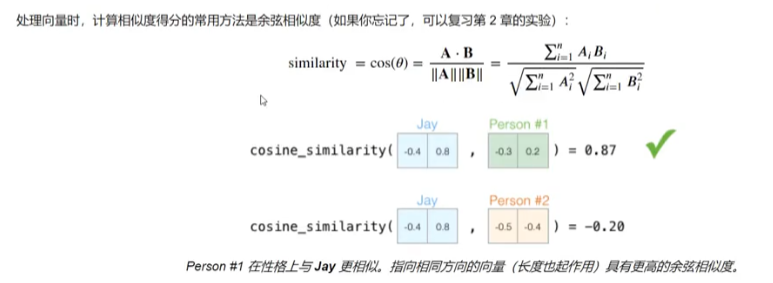


了解一个人，如下图知道爱好和身高，就得出了下图蓝色线的向量





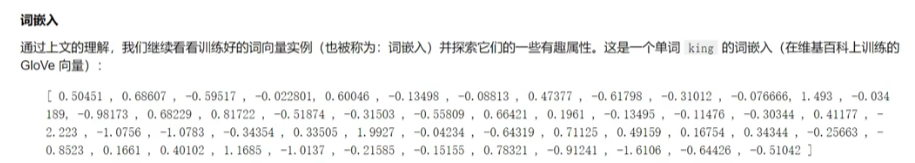
通过余弦函数，看下哪个人格更加适合jay，计算夹角余弦

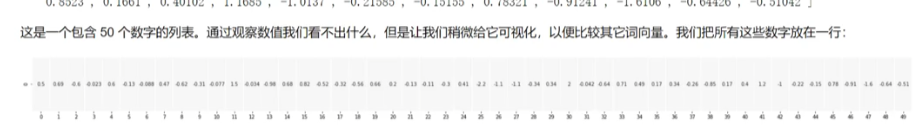


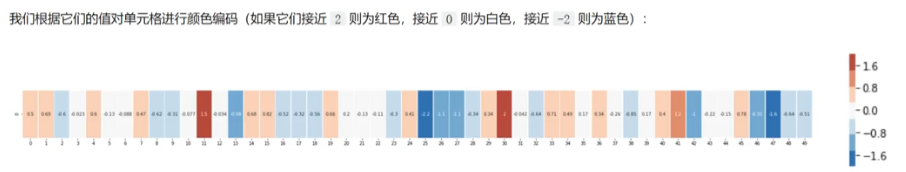
增加维度

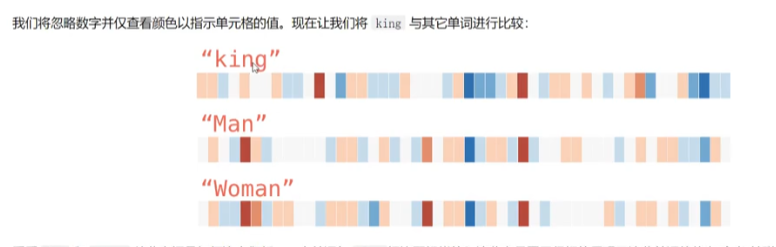


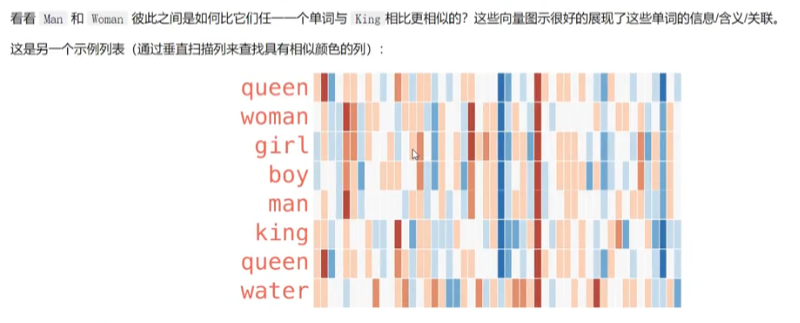


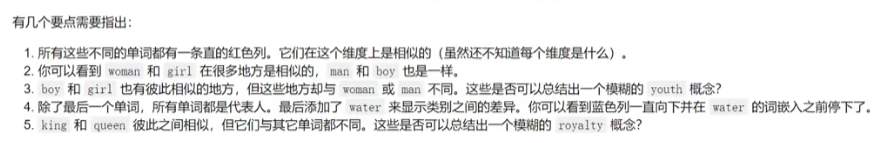












今天是周二学生都需要去上学

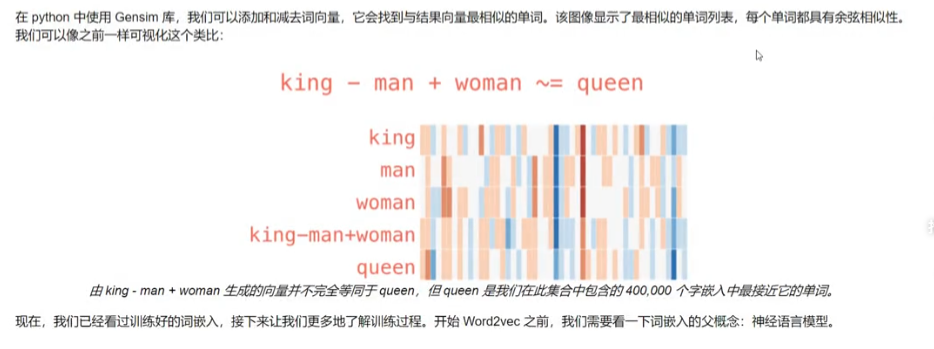
今天是周二孩子都需要去上学

今天是周二男孩都需要去上学

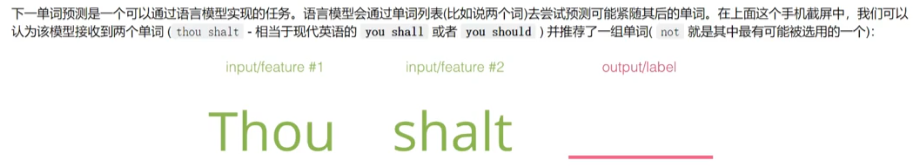
今天是周二少年都需要去上学

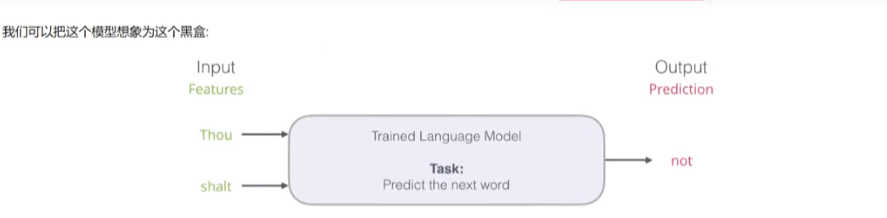
通过指定窗口大小和神经网路的训练，我们可以得出学生，孩子，男孩，少年在一个维度上面他们是相似的。他们之所以在某一维度是相似的，是因为在大量的维基百科训练文本当中，凡是出现学生，孩子，男孩，少年，较大概率会出现“上学”字眼，表示他们在上学方面的相关性

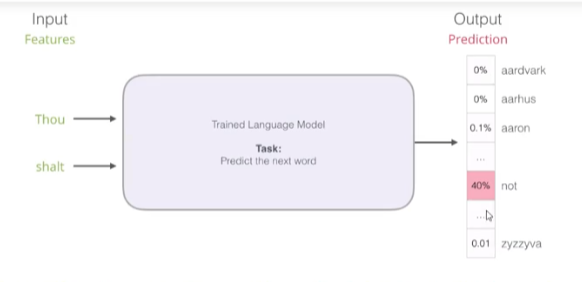




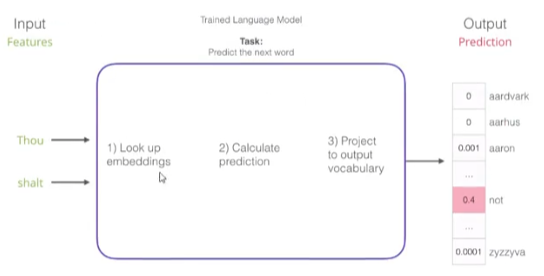






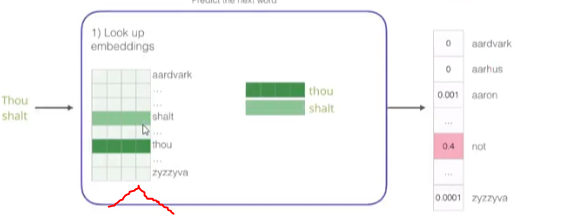


训练好之后，我们输入词汇，会根据下面模型预测，第一步就是embedding，





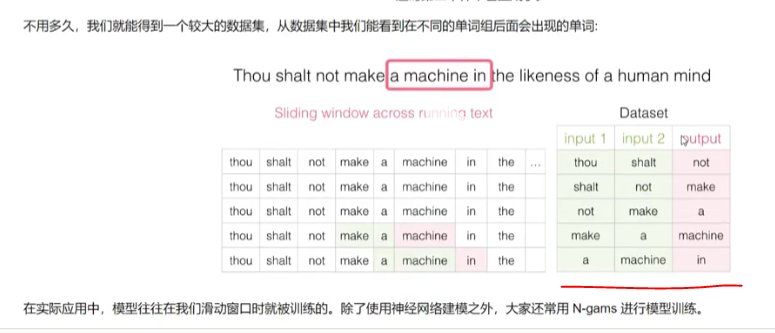
我们在训练目的就是得到一个embedding的词向量的矩阵





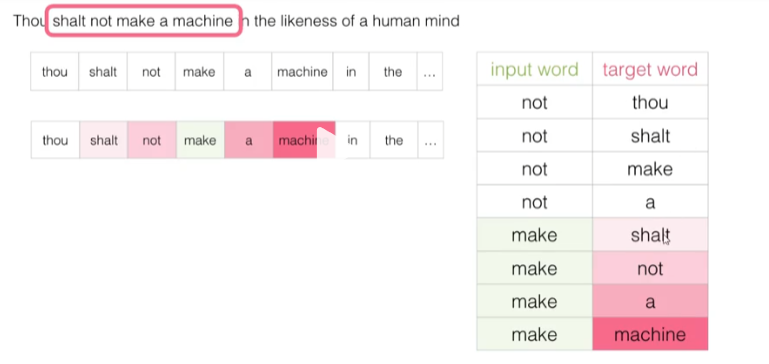
如下图三个单词，前两个单词就是特征，后面的not就是预期值，这就是一个训练集。

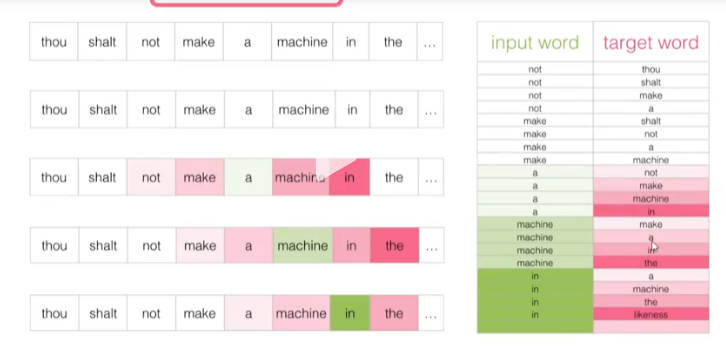






另一种方式







一开始得到的结果都输不正确，最后使用损失函数进行调整。





https://wangguisen.blog.csdn.net/article/details/113697460?spm=1001.2101.3001.6650.2&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-113697460-blog-77836142.235%5Ev38%5Epc\_relevant\_anti\_vip\_base&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-113697460-blog-77836142.235%5Ev38%5Epc\_relevant\_anti\_vip\_base&utm\_relevant\_index=3

**根据 Skip-gram 模型的理念，中心词决定了它的相邻 词**

**基于 Skip-gram 框架的 Word2vec 模型解 决的是一个多分类问题。**

基于 Skip-gram 框架的 Word2vec 模型可以结局解决one-hot的词向量，也可以解决word2vec的词向量。

# Embeddinge

就是嵌入，是从数据中提取的特征，就像嵌入到了特征空间中。深度学习广泛用embedding来指数据的特征，而现在基于大语言模型做知识库的常见方法，也是去提取和匹配embeddin

除目标单词外全是0，目标单词是1

纽约|是|金融中心。

纽约[1, 0, 0, 0]，是[1,0,0,0]

// 标签，IT

每一篇文章都经过一轮以下算法的洗礼

动态分类

**发出**

1. 获取20字总结：每篇文章都会有一个标题，我们直接将标题以及前200字送给GPT帮助我们总结出文章主要介绍那些内容。要求GPT提炼出关键词汇，比如一篇文章介绍了react里面的react-router的原理以及用法，一般来说，文章都会有前言介绍，前言200字会概括文章的内容。送给GPT命令约为“我发了一个标题和短文给你，请你帮助我总结内容，用50字总结标题以及内容，总结的内容前后通过加上@符号。我给出几个分类给你，分别是科学，自然，人文，IT，水果，美术”，请你根据短文内容归属出一个分类，然后用分类前后加上%符号。我有很多标签 IT 。。。。

**回复**

@React里面的react-router组件库迎来的较大的革新，本文主要讲解react-router的主要用法@

%IT%

**操作**

将回复内容描述存到数据库，那么我们就有了文章id，前200文章内容，文章总结，归属分类。

**重复以下操作：**

**发出**

**Java,python**

我刚刚只是发了文章的前200字内容给你，由于文章太长，我会将文章分开发送给你，请你说一下这篇短文中说的是关于谁的内容，哪些内容，总结的内容不超过20个字，文章内容为：xxx，总结内容前后加上@符号给我，并给出文章中心的讨论对象是谁，专业名词，用%符号包裹。

**回复**

@讲述了新老版本react-router在使用上有什么区别@

% react-router%

**生成目录**

重复以上操作，我们得到了一个类似于目录的结构

如：

1. 讲述了新老版本react-router在使用上有什么区别 react-router

2、讲述了路由状态和页面跳转 react-router

3、react-router新版本路由配置 react-router

4、讲述了路由状态和页面跳转 react-router

....

1. 讲述了BrowserRouter 的使用 BrowserRouter
2. 讲述了React-router的新版本用法 react-router

数据库存储为：

第一层：文章id， 文章标题 文章总结 归属分类

第二层：文章id index 目录 中心词

权重排序

未知分类

**使用：**

用户问问题：请问react-router的新版本用法是怎么样子的。

**发送：**

我写了一句话，请你提炼出句子中的关键词，中心词，以及句子的主要内容词语给我。可以选择多个关键词，中心词，请用@符号包裹。

**返回：**

@React-router@ @新版本@

**操作：**

将关键词语放到数据库进行搜索，搜索对象为文章总结，目录，中心词，例如我们找到了5篇文章的文章总结或者里面的目录出现了react-router字眼，出现react-router最多的文章排第一，按照关键字存在数量进行排序。

**发送1**

上面我们已经构造好了一篇文章的数据，我们将文章的标题，文章总结，目录内容，中心词发送给GPT，如：某用户提问了一个问题，问题为：请问react-router的新版本用法是怎么样子的。我们写了一篇文章，标题为：xxx，总结为:xxx,目录为1、讲述了新老版本react-router在使用上有什么区别。2讲述了路由状态和页面跳转。3、react-router新版本路由配置 ....6、讲述了React-router的新版本用法。请你给出哪个目录比较适合回答用户的问题，请给出目录的下标，下标前后加上@符号。如果目录没有涉及到回答用户的问题，请返回符号@-1@。

**返回**

@6@

**操作并发送**

将6对应的文章内容发送给GPT，并叫GPT依据内容进行回答。如：用户提问了一个问题：请问react-router的新版本用法是怎么样子的。我找到了一篇短文可能有讲解了用户的问题，文章内容为：xxxx，如果文章内容没有涉及到用户的问题，请返回@-1@。

**返回**

@-1@

**模仿发送1**

例如上面我们已经排序好了一篇文章的数据，我们将文章的标题，文章总结，目录内容，中心词发送给GPT，如：某用户提问了一个问题，问题为：请问react-router的新版本用法是怎么样子的。我们写了一篇文章，标题为：xxx，总结为:xxx,目录为1、讲述了新老版本react-router在使用上有什么区别。2讲述了路由状态和页面跳转。3、react-router新版本路由配置 ....6、讲述了React-router的新版本用法。请你给出哪个目录比较适合回答用户的问题，请给出目录的下标，下标前后加上@符号。如果目录没有涉及到回答用户的问题，请返回符号@-1@。