参考链接：<http://www.ruanyifeng.com/blog/2013/03/cosine_similarity.html>

为了找出相似的文章，需要用到["余弦相似性"](http://en.wikipedia.org/wiki/Cosine_similarity" \t "http://www.ruanyifeng.com/blog/2013/03/_blank)（cosine similiarity）。下面，我举一个例子来说明，什么是"余弦相似性"。

为了简单起见，我们先从句子着手。

　　句子A：我喜欢看电视，不喜欢看电影。

句子B：我不喜欢看电视，也不喜欢看电影。

请问怎样才能计算上面两句话的相似程度？

基本思路是：如果这两句话的用词越相似，它们的内容就应该越相似。因此，可以从词频入手，计算它们的相似程度。

****第一步，分词。****

句子A：我/喜欢/看/电视，不/喜欢/看/电影。

句子B：我/不/喜欢/看/电视，也/不/喜欢/看/电影。

****第二步，列出所有的词。****

　　我，喜欢，看，电视，电影，不，也。

****第三步，计算词频。****

　　句子A：我 1，喜欢 2，看 2，电视 1，电影 1，不 1，也 0。

　　句子B：我 1，喜欢 2，看 2，电视 1，电影 1，不 2，也 1。

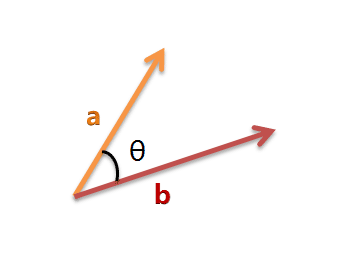
****第四步，写出词频向量。****

　　句子A：[1, 2, 2, 1, 1, 1, 0]

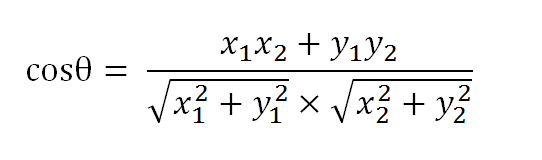
　　句子B：[1, 2, 2, 1, 1, 2, 1]

到这里，问题就变成了如何计算这两个向量的相似程度。

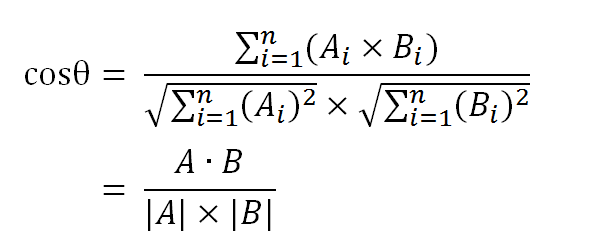
我们可以把它们想象成空间中的两条线段，都是从原点（[0, 0, ...]）出发，指向不同的方向。两条线段之间形成一个夹角，如果夹角为0度，意味着方向相同、线段重合；如果夹角为90度，意味着形成直角，方向完全不相似；如果夹角为180度，意味着方向正好相反。****因此，我们可以通过夹角的大小，来判断向量的相似程度。夹角越小，就代表越相似。****



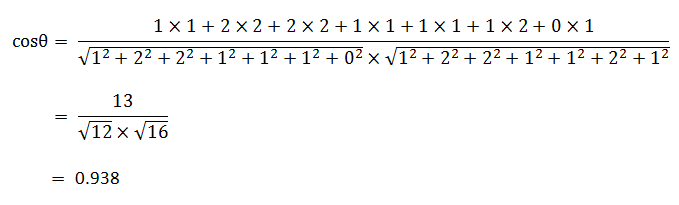
假定a向量是[x1, y1]，b向量是[x2, y2]，那么可以将余弦定理改写成下面的形式：



数学家已经证明，余弦的这种计算方法对n维向量也成立。假定A和B是两个n维向量，A是 [A1, A2, ..., An] ，B是 [B1, B2, ..., Bn] ，则A与B的夹角θ的余弦等于：



使用这个公式，我们就可以得到，句子A与句子B的夹角的余弦。



****余弦值越接近1，就表明夹角越接近0度，也就是两个向量越相似，这就叫"余弦相似性"。****所以，上面的句子A和句子B是很相似的，事实上它们的夹角大约为20.3度。

由此，我们就得到了"找出相似文章"的一种算法：

**（1）使用TF-IDF算法，找出两篇文章的关键词；**

**（2）每篇文章各取出若干个关键词（比如20个），合并成一个集合，计算每篇文章对于这个集合中的词的词频（为了避免文章长度的差异，可以使用相对词频）；**

**（3）生成两篇文章各自的词频向量；**

**（4）计算两个向量的余弦相似度，值越大就表示越相似。**