

TF-IDF与余弦相似性的应用（二）：找出相似文章

作者： 阮一峰

日期： 2013年3月21日

上一次，我用[TF-IDF算法](#)自动提取关键词。

今天，我们再来研究另一个相关的问题。有些时候，除了找到关键词，我们还希望找到与原文章相似的其他文章。比如，"Google新闻"在主新闻下方，还提供多条相似的新闻。

北京气象专家解释“泥雪”：长期无降水空气脏

金羊网 - 4小时前

两人合撑一把伞在雨中打车。昨天，京城迎来一场雨夹雪。记者陶冉摄。今天是春分节气，时中到大雪，而平原地区由于气温原因以雨夹雪为主。截至昨晚8点，城区 ...



凤凰网



搜狐



每日甘肃



搜狐



腾讯网



北国网

北京暴雪清污染京城三月飘雪好预兆【组图】

www.591hx.com - 3小时前

飞雪迎春袭北京京城今晨或现“堵城”

大洋网 - 3小时前

北京普降瑞雪银装素裹树挂景观成春日美景

艾拉家居网 - 7小时前

延庆迎春雪城区下泥雪专家称系内蒙古沙尘被卷来

凤凰网 - 9小时前

昨夜北京普降大雪道路结冰早高峰注意出行安全

张家界在线 - 11小时前

北京春分降雪空气净化专家称三月下雪很正常

腾讯网 - 11小时前

为了找出相似的文章，需要用到"[余弦相似性](#)"（cosine similarity）。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。

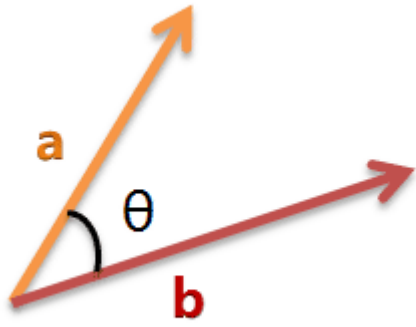
第四步，写出词频向量。

句子A：[1, 2, 2, 1, 1, 1, 0]

句子B：[1, 2, 2, 1, 1, 2, 1]

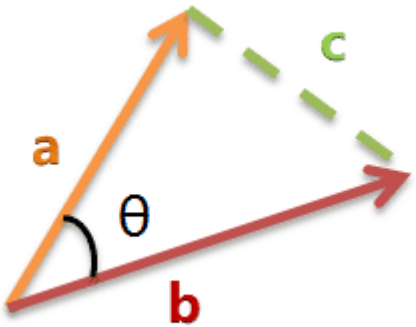
到这里，问题就变成了如何计算这两个向量的相似程度。

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。



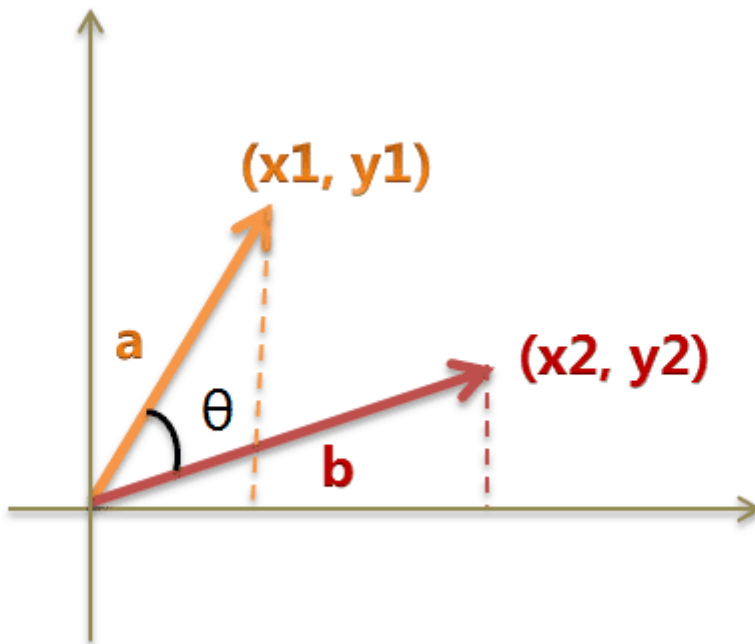
以二维空间为例，上图的a和b是两个向量，我们要计算它们的夹角 θ 。[余弦定理](#)告诉我们，可以用下面的公式求得：

$$\cos\theta = \frac{a^2 + b^2 - c^2}{2ab}$$



假定a向量是 $[x_1, y_1]$ ，b向量是 $[x_2, y_2]$ ，那么可以将余弦定理改写成下面的形式：

$$\cos\theta = \frac{x_1x_2 + y_1y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$



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 $[A_1, A_2, \dots, A_n]$ ，B是 $[B_1, B_2, \dots, B_n]$ ，则A与B的夹角 θ 的余弦等于：

$$\begin{aligned}\cos\theta &= \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \\ &= \frac{A \cdot B}{|A| \times |B|}\end{aligned}$$

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。

$$\begin{aligned}\cos\theta &= \frac{1 \times 1 + 2 \times 2 + 2 \times 2 + 1 \times 1 + 1 \times 1 + 1 \times 2 + 0 \times 1}{\sqrt{1^2 + 2^2 + 2^2 + 1^2 + 1^2 + 1^2 + 0^2} \times \sqrt{1^2 + 2^2 + 2^2 + 1^2 + 1^2 + 2^2 + 1^2}} \\ &= \frac{13}{\sqrt{12} \times \sqrt{16}} \\ &= 0.938\end{aligned}$$

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。

"余弦相似度"是一种非常实用的算法，只要是计算两个向量的相似程度，都可以采用它。

下一次，我想谈谈如何在词频统计的基础上，自动生成一篇文章的摘要。

(完)

文档信息

- 版权声明：自由转载-非商用-非衍生-保持署名（创意共享3.0许可证）
- 发表日期：2013年3月21日

算法训练营体验课

7 天掌握大厂算法面试 必考知识点

知识点精讲

- 时间复杂度
- 数组、链表、跳表
- 树、二叉树、二叉搜索树
- 递归

大厂面试题解析

- 美团
- 滴滴出行
- 京东商城

社群活动多多

- 2020 最新面经分享
- 每周一次讲师直播
- 班主任督学
- 助教作业精讲 + 答疑

大厂面试题解析

- 美团
- 滴滴出行
- 树
- 跳表
- 链表
- 数组
- 递归

相关文章

- 2019.01.28: [Prolog 语言入门教程](#)