TF-IDF原理与实践

# TF-IDF算法介绍

TF-IDF（Term Frequency–InverseDocument Frequency）是一种用于资讯检索与文本挖掘的常用加权技术。TF-IDF的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

上述引用总结就是：一个词语在一篇文章中出现次数越多, 同时在所有文档中出现次数越少, 越能够代表该文章。

TF(Term Frequency, 词频)表示词条在文本中出现的频率，这个数字通常会被归一化(一般是词频除以文章总词数)，以防止它偏向长的文件（同一个词语在长文件里可能会比短文件有更高的词频，而不管该词语重要与否）。TF用公式表示如下



其中，nij表示词条ti在文档dj中出现的次数，TFij就是表示词条ti在文档dj中出现的频率。

但是，一些通用的词语对于主题并没有太大的作用，反倒是一些出现频率较少的词才能够表达文章的主题，所以单纯使用是TF不合适的。权重的设计必须满足：一个词预测主题的能力越强，权重越大，反之，权重越小。所有统计的文章中，一些词只是在其中很少几篇文章中出现，那么这样的词对文章的主题的作用很大，这些词的权重应该设计的较大。IDF就是在完成这样的工作。

IDF(Inverse Document Frequency, 逆文件频率)表示关键词的普遍程度。如果包含词条i的文档越少， IDF越大，则说明该词条具有很好的类别区分能力。某一特定词语的IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到



其中，|D|表示所有文档的数量，表示包含词条ti的文档数量，为什么这里要加 1 呢？主要是防止包含词条ti的数量为0从而导致运算出错的现象发生。

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语，表达为



# 1、原理

简单来说，TF-IDF是一种数值统计方法，它通常用以评估，在一个文件集或一个语料库中，其中某一份文件的某一字词的重要程度。

它由两部分组成，TF和IDF。TF（Term Frequency）词频，是指一个词在一篇文章中出现的频率，其公式化如下：



上式中，分子是指这个词在文章中出现的次数，而分母是指文章中所有字数之和。最后得出词频（TF）。

IDF（inverse document frequency）逆文档频率，指一个词在多少篇文章中出现过，是词频的“权重”度量，在词频的基础上，如果一个词在多篇文档中出现的频率较低，也就表示这是一个比较少见的词，但在某一篇文章中却出现了很多次，则代表这个词IDF值越大，在这篇文章中的“权重”也越大。所以当一个词越常见，它的IDF值反而越低。IDF的计算公式为：



概括来讲， IDF反应了一个词在所有文本中出现的频率，如果一个词在很多的文本中出现，那么它的IDF值应该低，比如上文中的“to”。而反过来说，如果一个词在比较少的文本中出现的次数多，那么它的IDF值应该高。比如一些专业的名词如“Machine Learning”等等。这样的词IDF值应该高。因此，还有一个极端的情况，那就是如果一个词在所有的文本中都出现，那么它的IDF值应该为0。

当计算出TF和IDF的值后，两个一乘就得到TF-IDF，这个词的TF-IDF指数越高就表示，就表示这个词在这篇文章中的重要性越大，越有可能就是文章的关键词。

假设，共有N篇文档，分别用 d1,d2,d3,…,dn来表示。

TF = 某个词在di篇文章中出现的次数/di篇文章的总词数 = count（W in di）/ count(di）。因此，TF计算的是单个词在单个文档中出现的词频。

IDF = 总的文档数 / 出现词W的文档数。IDF其实反映了词W在文档之间的区别度。如果W在仅在一篇文档中出现，则说明可以使用W将该文档与其他文档区别开来。即IDF可以反映W的独特性。

TF\*IDF，可以得到词的重要性。比如：北京和西安在同一篇文档中的词频均为20%，那如何估计北京是该文的关键字，还是西安呢？如果同时有10篇文章均提到了北京，恰好只有这篇文章提到了西安，则西安作为这篇文章的关键字更为合理。

于是TF-IDF = TF \* IDF, 它表明字词的重要性与它在某篇文档中出现的次数成正比，与它在 所有文档中出现的次数成反比

使用TF-IDF的一个假设前提是：已经去掉了停用词。

TF-IDF的优点是计算速度快，结果稳健。

需要输入多篇文档，可以输出每篇文档的关键词。

# 2、TF-IDF与TextRank算法的比较

1. 从算法原理上来看，基础都是词频统计，只是TD-IDF通过IDF来调整词频的权值，而TextRank通过上下文的连接数来调整词频的权值。TextRank通过滑动窗口的方式，来实现词的位置对词的权值的影响。
2. TD-IDF计算简单，运行性能更好。

参考：

<https://mp.weixin.qq.com/s/7sbMdNxbIvp_mST7PB4KUw>

<https://mp.weixin.qq.com/s/1mmoHIWHIy_CD5n6WHHJTg>

<https://mp.weixin.qq.com/s/usTdmF6Fs_9fkHHK40LzfA>

<https://mp.weixin.qq.com/s/xOAfDclb49RaqIWN6UOtPA>