TextRank算法

TextRank算法是用来提取关键词和关键句，以及自动摘要的算法，因为TextRank算法是基于PageRank算法的，因此下面我们先介绍PageRank算法。(TD-IDF算法也可以用于提取关键字)

参考链接

1. PageRank算法

1.1原理

PageRank算法也叫做网页排名，根据网页之间的相互超链接来计算网页之间的排名，它体现了网页的相关性和重要性。

假设一个由四个网页组成的群体:A,B,C,D，那么对于一个页面A,它的PR(PageRank)值为：

其中：

--为页面A的PR值。

--是页面的PR值，在这里，页面是指向A的所有页面中的某个页面。

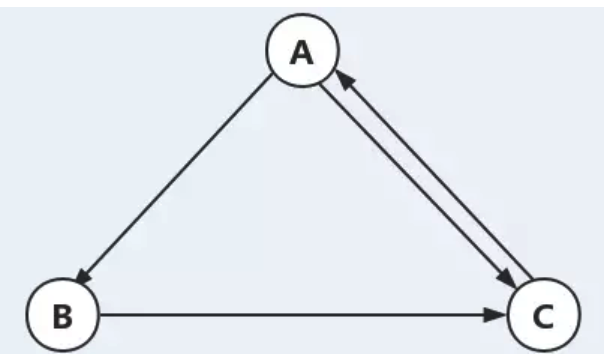
--是页面的出度，也就是页面指向其他页面的边的个数。

--d为阻尼系数，其意义是，在任何时刻，用户到达某页面后继续向后浏览的概率。d通常为0.85。它是根据上网者使用浏览器书签的平均频率得到的。

此外，还有另一个版本的公式：

其中N为页面的总数。

1.2具体实例



图一：三个页面A,B,C

初始时，为了方便计算，我们假设每个页面的PR值均为1，d轻为0.5，那么计算过程如下：

* PR(A) = 0.5 + 0.5\*PR(C) = 0.5+0.5\*1 = 1
* PR(B) = 0.5 + 0.5\*(PR(A)/2) = 0.5 + 0.5\*(0.5/2) = 0.75
* PR(C) = 0.5 + 0.5(PR(A)/2 + PR(B)) = 1.125

在经过12轮迭代后，各个页面的PR值如下：

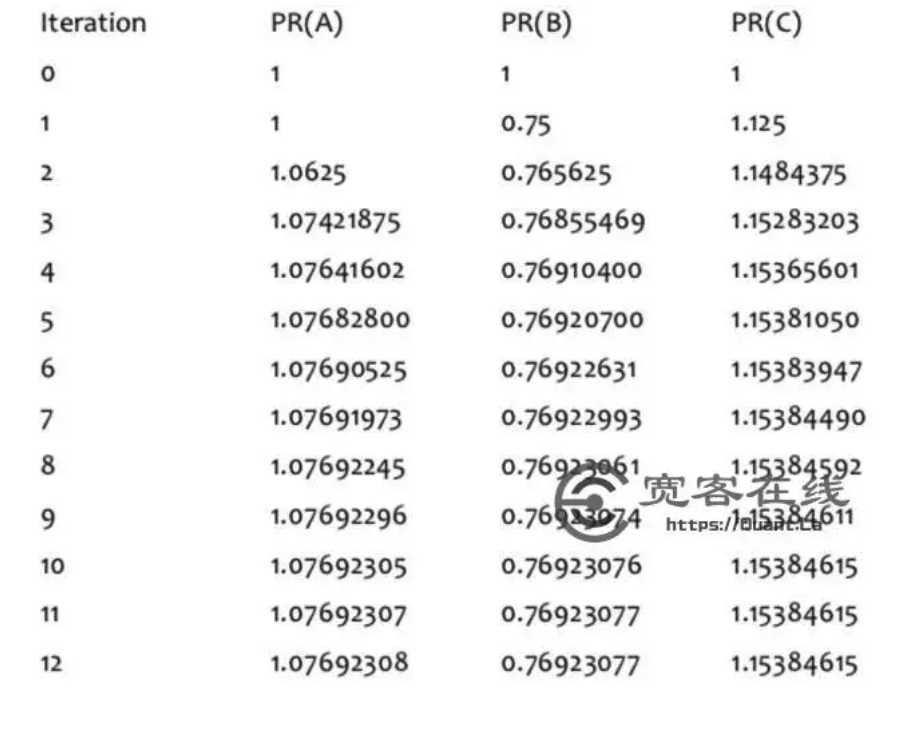


图2：PR值迭代过程

一般要设置收敛条件，比如上次迭代结果与本次迭代结果小于某个误差，那么我们结束程序运行。

1. TextRank算法

2.1算法原理

TextRank算法的基本思想来源于谷歌的PageRank算法，是一种用于文本的基于图的排序算法。通过将文本分割成若干组成单元(单词、句子)并建立图模型，利用投票机制对文本中的重要成分进行排序，仅仅利用单篇文档本身的信息就可以实现关键词的提取。

PageRank算法根据网页之间的连接关系构造网络，而TextRank算法根据词之间的共现关系构造网络；PageRank算法构造的网络中的边是有向无权边，而TextRank算法构造的网络中的边是无向有权边，TextRank算法的核心公式如下：

TextRank公式的计算与PageRank的公式计算非常相似，区别在于加入了一个参数,这个就是在句中第j个单词和第i个单词在窗口中一起出现的次数。 其实前面的In(Vi)其实就是可以理解为和单词i一起在窗口里面出现的所有单词【也就是pagerank里面的所有指向当前页面的页面】，然后对于和i一起出现的单词j，对他们的WS(Vj)值进行求和，但是这个求和是要有权重的，也就是这里的，这里的Out(Vj)其实就是和单词j在同一个窗口里面出现的所有词，也就是用Wij除以j的所有邻接词的总词频来表示概率【对应到PageRank中的1/C(T),1/C(T)其实也可以看做是个概率值】。

2.2实现过程

提取关键词和关键词组的步骤如下：

--1）将给定的文本按照整句进行分割，即

--2）对于每个句子,对其进行分词和词性标注，然后剔除停用词，只保留指定词性的词，比如形容词、动词、名词等，即,其中为句子中保留下的词。

--3）构建词图,其中为节点集合，由以上步骤生成的词组成，然后采用共现关系构造任意两个节点之间的边：两个节点之间存在边当且仅当他们对应的词在长度为K的窗口中出现，K表示窗口大小，即最多出现K个单词，一般K取2.

--4）根据上面的公式，迭代计算各节点的权重，直至收敛；

--5）对节点的权重进行倒序排序，从中得到最重要的t个单词，作为top-t关键词；

--6）对于得到的top-t关键词，在原始文本进行标记，若他们之间形成了相邻词组，则作为关键词组提取出来；

从给定的文本中提取关键句时，将文本中的每个句子看做是一个节点，如果两个句子有相似性，则认为这两个句子对应的节点之间存在一条无向有权边，衡量句子之间相似性的公式如下：

、是两个句子，是句子中的词。

分子部分表示的含义是同时出现在两个句子中的同一个词的数量，分母是对句子中的词的个数求对数然后求和，这样设计可以遏制较长的句子在相似度计算上的优势。

根据以上相似度计算公式循环计算任意两个节点之间的相似度，设置阈值去掉两个节点之间相似度较低的边链接，构建出节点链接图，然后迭代计算每个节点的TextRank值，排序后选出TextRank值最高的几个几点对应的句子作为关键句。