BM25算法 原理简介

小白的进阶 于 2021-09-06 20:10:27 发布

bm25 是什么?

bm25 是一种用来评价搜索词和文档之间相关性的算法，它是一种基于**概率检索模型**提出的算法，再用简单的话来描述下bm25算法：我们有一个query和一批文档Ds，现在要计算query和每篇文档D之间的相关性分数，我们的做法是，先对query进行切分，得到单词q_i，然后单词的分数由3部分组成：

- query中每个单词和文档d之间的相关性
- 单词和query之间的相关性
- 每个单词的权重

最后对于每个单词的分数我们做一个求和，就得到了query和文档之间的分数。

下面将从3个部分来介绍bm25算法。

单词权重

单词的权重最简单的就是用idf值

$$IDF(q_i) = log\frac{N - df_i + 0.5}{df_i + 0.5}$$

其中N表示索引中全部文档数，df_i为包含了q_i的文档的个数。依据IDF的作用，对于某个q_i，包含q_i的文档数越多，说明q_i重要性越小，或者区分度越低，IDF越小，因此IDF可以用来刻画q_i与文档的相似性。

单词和文档的相关性

tf-idf中，这个信息直接就用“词频”，如果出现的次数比较多，一般就认为更相关。但是BM25洞察到：词频和相关性之间的关系是非线性的，具体来说，每一个词对于文档相关性的分数不会超过一个特定的阈值，当词出现的次数达到一个阈值后，其影响不再线性增长，而这个阈值会跟文档本身有关。

在具体操作上，我们对于词频做了”标准化处理“，具体公式如下：

$$S(q_i, d) = \frac{(k_1 + 1)tf_{td}}{K + tf_{td}}, \quad K = k_1(1 - b + b * \frac{L_d}{L_{ave}})$$

其中，tf_{td}是单词t在文档d中的词频，L_d是文档d的长度，L_{ave}是所有文档的平均长度，变量k_1是一个正的参数，用来标准化文章词频的范围，当k_1=0，就是一个二元模型（binary model）（没有词频），一个更大的值对应使用更原始的词频信息。b是另一个可调参数（$0 < b < 1$），他是用决定使用文档长度来表示信息量的范围：当b为1，是完全使用文档长度来权衡词的权重，当b为0表示不使用文档长度。

单词和查询的相关性

当query很长时，我们还需要刻画单词与query的之间的权重。对于短的query，这一项不是必须的。

$$S(q_i, Q) = \frac{(k_3 + 1)tf_{tq}}{k_3 + tf_{tq}}$$

这里tf_{tq}表示单词t在query中的词频，k_3是一个可调正参数，来矫正query中的词频范围。

完整公式

于是最后的公式为：

$$RSV_d = \sum_{t \in q} log\left[\frac{N - df_i + 0.5}{df_i + 0.5}\right] \cdot \frac{(k_1 + 1)tf_{td}}{k_1(1 - b + b * \frac{L_d}{L_{ave}}) + tf_{td}} \cdot \frac{(k_3 + 1)tf_{tq}}{k_3 + tf_{tq}}$$

————————————————————