

1. 相关度评分算法揭秘

(1) boolean model: 所谓bool 就是 true false

1. match operator: or and not
2. bool => must/must not/should
3. term->doc分数

(2) boost: 权重

(3) TF-IDF

1. TF(词频term frequency): 关键词在每个doc中出现的次数
2. IDF(反文档词频inversed document frequency):关键词在整个索引中出现的次数
3. norm:字段长度越长, 值越小。
4. 空间向量模型 或者向量空间模型
 - 1) 比如 query: hello word
5. 计算公式:

$$\begin{aligned} \text{score}(q,d) = & \\ & \text{queryNorm}(q) \\ & \cdot \text{coord}(q,d) \\ & \cdot \sum (\\ & \quad \text{tf}(t \text{ in } d) \\ & \quad \cdot \text{idf}(t)^2 \\ & \quad \cdot t.\text{getBoost}() \\ & \quad \cdot \text{norm}(t,d) \\ &) (t \text{ in } q) \end{aligned}$$

- 1) $\text{score}(q,d)$: query对一个doc最终的评分结果。
- 2) $\text{queryNorm}(q)$: 想想normalization, 在不影响相互关系的前提下, 把看似离散的数据, 转换到一个相近的区间=>人性化
 - a. $\text{queryNorm} = 1 / \sqrt{\text{sumOfSquaredWeights}}$
 - b. $\text{sumOfSquaredWeights}$ 是通过将查询中每个项的IDF平方相加来计算的。
- 3) $\text{coord}(q,d)$: 对匹配的结果加分, 越匹配的doc加分越多
$$\begin{aligned} \text{doc with hello} &\rightarrow \text{score: } 1.5 * 1 / 3 = 0.5 \\ \text{doc with hello word} &\rightarrow \text{score: } 3.0 * 2 / 3 = 2.0 \\ \text{doc with hello word elastic} &\rightarrow \text{score: } 4.5 * 3 / 3 = 4.5 \end{aligned}$$
总分数 * 匹配的term数 / 总term数
- 4) \sum : doc对query中每个term的权重的总和
- 5) $\text{tf}(t \text{ in } d)$:
 - a. $\text{tf}(t \text{ in } d) = \sqrt{\text{frequency}}$
 - b. 该term在doc中出现的次数的平方根

- 6) $\text{idf}(t)$:
 - a. $\text{idf}(t) = 1 + \log (\text{numDocs} / (\text{docFreq} + 1))$
 - b. trem 的反文档频率(idf)是索引中文档数量的对数，除以包含该 trem 的doc数量
- 7) $\text{t.getBoost}()$: 设置的权重值.
- 8) $\text{norm}(t,d)$: 字段长度越长，结果月越小
 - a. $\text{norm}(d) = 1 / \sqrt{\text{numTerms}}$
 - b. 字段长度范数(范数)是字段中项数的平方根的倒数

2. Java API

(1) Transport Client: TransportClient不推荐使用，而推荐使用Java High Level REST Client，并将在Elasticsearch 8.0中删除。

(2) JAVA REST Client

1. **Java Low Level REST Client:** 低级别的REST客户端，通过http与集群交互，用户需自己编组请求JSON串，及解析响应JSON串。**兼容所有ES版本**
2. **Java High Level REST Client:** 高级别的REST客户端，基于低级别的REST客户端，增加了编组请求JSON串、解析响应JSON串等相关api。使用的版本需要保持和ES服务端的版本一致，否则会有版本问题。

1) 要求最低JDK 1.8

2) Maven配置

```
<dependency>
<groupId>org.elasticsearch.client</groupId>
<artifactId>elasticsearch-rest-high-level-client</artifactId>
<version>7.6.2</version>
</dependency>
```