Lucene为什么这么快？

倒排索引，倒排索引就是Lucene的核心！根据属性的值来查找记录。这种索引表中的每一项都包括一个属性值和具有该属性值的各记录的地址。由于不是由记录来确定属性值，而是由属性值来确定记录的位置，因为称为倒排索引（inverted index）。不同于传统的顺排索引（知道那个在文章的哪个位置），倒排索引是根据一个词，知道有哪几篇文章有这个词。

单词——文档矩阵

Lucene之所以那么快，是因为在搜索前，Lucene已经帮我们生成倒排索引，相比此前的数据库like的模糊搜索效率更高！

两个概念，document 和 field

document

用户提供的源是一条条记录，它们可以是文本文件、字符串或者数据库表的一条记录等等。一条记录经过索引之后，就是以一个document的形式存储在索引文件中的。用户进行搜索，也就是以Document列表的形式返回。

field

一个Document可以包含多个信息域，例如一篇文章可以包含“标题”、“正文”、“最后修改时间”等信息域。这些信息域就是通过Field在Document中存储的。Field有两个属性可选：存储和索引。通过存储属性可以控制是否对该Field进行索引。

lucene的工作方式

lucene提供的服务实际包含两部分：一入一出。所谓入是写入，即将提供的源（本质是字符串）写入索引或者将其从索引中删除；所谓出是读出，即向用户提供全文搜索服务，让用户可以通过关键词定位源。

写入流程：

源字符串首先经过analyzer处理，包括：分词，分成一个个单词；去除stopword（可选）。将源中需要的信息加入Document的各个Field中，并把需要索引的Field索引起来，把需要存储的Field存储起来。将索引写入存储器，存储器可以是内存或磁盘。

读出流程

用户提供搜索关键词，讲过analyzer处理。对处理后的关键词搜索索引找出对应的Document。用户根据需要从找到的Docuement中提取需要的Field。

Lucene打分公式

Lucene的打分公式决定搜索出来的文件排序，然而，Lucene的打分公式非常复杂：

$$score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_{t \text{ in } q} (tf(t \text{ in } d) \times idf(t)^2 \times t.getBoost() \times norm(t,d))$$

TF：单个文章的词频，词在文档中出现的词频

IDF：逆词频，词在这篇文档中出现过次数/词在所有文章出现的次数。

https://www.jianshu.com/p/1f3ba892fc64