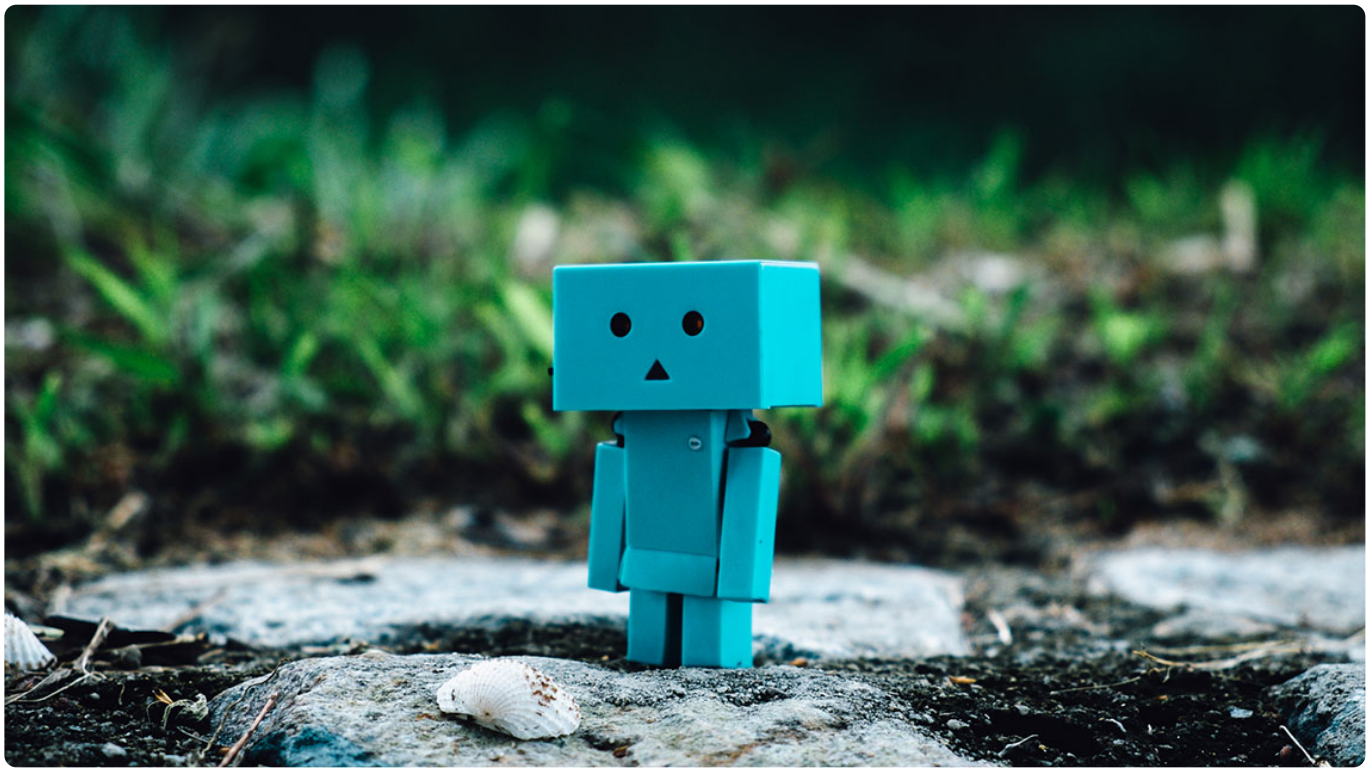
035 | 搜索索引及其相关技术概述

2017-12-22 洪亮劼

AI技术内参 进入课程 >



讲述：初明明
时长 07:53 大小 3.62M

本周我们分享的主题是从宏观上来剖析现代搜索架构。周一我介绍了搜索系统的一个大的分类，一类是从 20 世纪 50 年代开始研发并使用的传统文本匹配信息检索系统，一类是从 2000 年开始发展并逐渐成熟的机器学习信息检索系统。周三我们剖析了搜索系统的另一个框架体系，多轮打分系统，阐述了为什么需要多轮打分，以及每一轮打分又有什么特性。

今天，我们来看一个在本周已经反复涉及到的话题：**倒排索引**（Inverted Index）。一起来聊聊它的核心技术。值得注意的是，关于索引的很多话题其实都会牵涉到搜索中的“查询关键字处理”（Query Processing），我们今天的分享就主要来谈谈索引及相关技术在“查询关键字处理”这个场景下的应用。

经典的索引结构

经典的索引结构由“字段”（Field）和对应的列表组成。一般来说，“字段”就是某一个查询关键字。在英文里，这就是一个单独的单词；在中文里，这也许就是一个词或者短语。每个字段所对应的列表就是包含这个查询关键字的文档列表。

有两点值得注意。

第一，在文档列表里的文档，大多按照某种重要的顺序排列，这方便我们首先提取重要性高的文档。比如，某个文档整体和查询关键字的相关度大，那么就会排列到这个列表的前面。

第二，对于每个字段，也就是查询关键字而言，所有包含这个查询关键字的文档并不一定都会包含到这个列表中，这个列表可以是一个节选。

另外，我们前面已经讲过了，之所以叫做“索引”，也是因为这个列表中并不实际存储整个文档，往往是只存储文档的编号。

如果用户输入的查询关键字包含多个词组，根据这个最基础的结构，我们可以很容易地获取包含所有关键字的文档集合。这个操作仅仅相当于在多个列表中做“**归并排序**”（Merge Sort）。

除了在索引中仅仅保存最基本的文档标号信息以外，另外一些文档的基础信息也可以一并存放在索引中。比如，**经常存放的信息还有文档包含某个查询关键字的次数。**保存次数信息本质上是在保存“词频”（Term Frequency）这个文档特性。

我们前面分享经典的信息检索模型的时候，介绍过很多模型，例如 TF-IDF、BM25 或者语言模型，都对词频的计算有很强的依赖。**在索引中存放词频信息有助于近似计算这些基础的检索模型。**

另外一个经常存放的信息就是查询关键字在文档中出现的位置（Position）。位置信息对于有多个查询关键字的时候尤为重要。比如，我们要搜索的词组是“五道口电影院”。在这样的情况下，我们非常希望“五道口”在某个文档中出现的位置和“电影院”在文档中出现的位置相邻。这样，我们可以确认这个文档的确是关于“五道口电影院”的，而不是恰好含有“五道口”和“电影院”这两个词。

同时，位置信息还可以帮助搜索引擎生成搜索结果界面上的“结果摘要”信息。我们经常看到搜索结果页面上有几句话的摘要信息，这个信息就需要查询关键字的位置来生成。

索引技术

除了最基础的索引技术以外，研发人员开发了多种技术让索引更加高效。

第一个技术当然就是希望对索引进行压缩。索引信息很快就会随着可能的关键字数目的膨胀而扩展。索引中每一个关键字所对应的文档列表也会越来越庞大。因此，能否快速处理索引信息并为后续的计算节约时间就变得非常关键。本周三我们分享了多轮打分系统。多轮打分系统的的一个重要思想就是整个流程必须在几百毫秒的响应时间内完成。因此，每一个步骤，包括从索引中提取“顶部 K 个文档”的过程都需要很快捷。

压缩技术博大精深，我们在今天的分享中就不展开讨论这部分的内容了。在这里，我们只需要从高维度上把握这个问题的一个基本思路。索引的一个基本信息就是相对于某个查询关键字的文档列表。而存储在文档列表里的并不是文档本身的数据，而是文档的某种信息，比如文档本身的编号。而编号就是数字，文档列表最终就是一个数字序列。压缩技术中有很多算法就是对一个数字序列进行压缩。

那么，到底怎样才能起到压缩的作用呢？我们这里举一个例子。比方说，有一种压缩算法是基于一种叫“**差值编码**”（Delta Encoding）的技术。简单来说，就是不直接记录文档编号本身，而是按照文档编号的顺序，记录文档编号之间的差值。

对于某些非常频繁的查询关键字而言，这些词汇有可能会出现在非常多、甚至是绝大多数的文档中。而采用这种“差值编码”来对文档列表进行重新编排，我们就可以用一组很小的数（这些数表达两个相邻文档编号的差值）来代表文档列表。当然，这种方法对于文档很少的查询关键字效果肯定不明显。同时，这种技术也要求文档列表不按照相关度排序，而要按照文档的编号排序。

在索引的发展过程中也开发出了一些很细小的技术，比如“略过”（Skipping）。简单来说，这个技术就是，当我们有多个查询关键字的时候，而且这些关键字之间的频率有非常大的差距，我们可以略过一些文档。

例如在“北京，地铁出行”这个组合中，“北京”有可能在整个数据集中出现的频率是“地铁出行”的几倍甚至十几倍、上百倍，因此我们其实并不需要搜索所有包含“北京”的文档，因为最终需要的仅仅是同时包含两个关键字的这样一个交集。因此，在处理“北京”的文档序列的时候，我们可以“略过”K 个文档，然后看有没有到达下一个包含“地铁出行”的文档。这里的 K 当然是一个参数，需要尝试。有了这样的思路，处理多个查询关键字时就可以很显著地提升效果。

查询关键字处理

最后我们来谈一谈查询关键字处理。说得通俗易懂一点，就是如何从索引中提取出相关的文档并计算分数。这里有两种基本思路。

第一种思路叫作“文档优先”（Document-at-a-Time）计算策略。简单来说，就是我们首先从索引中找到所有查询关键字所对应的文档集合。比如我们处理“北京，地铁出行”这一查询关键字组合，我们先取出所有包含这些关键字的文档；然后保持一个“优先队列”（Priority Queue）来保存分数最高的 K 个文档；再针对取出来的文档分别计算分数，这里的分数有可能就是词频的某种简化检索模型；计算完分数之后，我们把分数压入优先队列中。

第二种思路和“文档优先”思路相对应，叫作“词优先”（Term-at-a-Time）计算策略。在这种思路下，我们对所有查询关键字词组中的每一个字一一进行处理。请注意，这里的第一个步骤其实是一样的，我们依然要先取出所有的文档集合。但是这一步之后，我们先处理包含“北京”的文档，得到所有文档分数的一个部分值，然后再处理“地铁出行”，在刚才计算的部分值上进行更新，取得最后的分数。

在实际应用中，这两种策略是更加复杂的优化查询关键字处理的基础，在这两种思路的基础上演化出了很多高级算法，不仅能快速地处理文字特性，还包括我们讲过的类似 WAND 操作符这样能够模拟线性模型的算法。

小结

今天我为你讲了现代搜索技术的一个核心组成部分，那就是倒排索引系统。 一起来回顾下要点：第一，我们讲了索引系统的基本组成和原理。第二，我们讲了索引相关技术的一个概况，重点介绍了压缩以及“略过”的含义。第三，简要讲解了查询关键字处理的两种最基础的策略。

最后，给你留一个思考题，如果我们既有图像信息又有文字信息，那该如何构建我们的索引呢？

欢迎你给我留言，和我一起讨论。

上一篇　034 | 多轮打分系统概述

下一篇　036 | PageRank算法的核心思想是什么?

精选留言 (2)

写留言

风的轨迹
2018-07-06

不好意思老师，我不太明白“查询关键字处理”的目的是什么。在倒排索引中，不是已经把文档按照相关度进行排序了吗？直接从倒排索引中取出来展示不就行了吗，为什么还要进行查询关键字处理

展开

嘉彦
2018-04-14

文档优先策略中先根据分数选top K个文档，这个分数是仅仅关于文档的分数吧，类似于文档的重要程度。应该和查询关键字的相关度无关?