

32 | 和搜索引擎的对话：SEO的原理和基础

2019-11-22 四火

全栈工程师修炼指南

[进入课程 >](#)



讲述：四火

时长 18:58 大小 13.04M



你好，我是四火。

今天，我们来聊一聊搜索引擎和 SEO（Search Engine Optimization）。当网站发布上线以后，我们希望通过适当的优化调整，让它可以被搜索引擎更好地“理解”，在用户使用搜索引擎搜索的时候，网站的内容可以更恰当地暴露给用户。

作为程序员，和更擅长于与内容打交道的运营相比，**我们的角度是不一样的，我们更关注工程实现而非网页内容，也更需要从原理的角度去理解 SEO。**这一讲，就让我们从理解互联网搜索引擎的工作原理开始。

互联网搜索引擎

要说 SEO，我觉得我们需要先来简单了解一下互联网上的搜索引擎。

组成部分

对于 Google 和百度这样的巨型 Web 搜索引擎来说，这里面的机制很复杂，而它们之间又有很多区别。比如被搜索的数据是怎样产生的，权重是怎样分配的，用户的输入又是怎样被理解的等等，但是大体上，它总是包含这样三部分。

1. 爬取 (Crawling)

搜索引擎会有若干个“爬虫”客户端定期地访问你的网站，如果数据有了变更，它们会将可访问的网页下载下来。搜索引擎发现网页的方式，和人是一样的，就是通过超链接。因此理论上，如果你建立了一个网站，但是你没有主动“告知”搜索引擎，也没有任何网站页面有超链接指向它，那么它是无法被搜索引擎的爬虫发现的。

2. 建立索引 (Indexing)

这一步其实就是将前面爬取的结果，经过解析和处理以后，以一种有利于搜索的方式归类存放起来。在 [🔗\[第 26 讲\]](#) 我们介绍了搜索引擎的倒排索引机制。

3. 返回结果 (Serving Results)

拆解用户的搜索条件，根据多种因素来决定返回哪些网页或其它资源给用户，也包括确定它们的展示顺序 (Ranking)。

这三个大致的步骤协同合作并提供了如今互联网搜索引擎的服务，当然，实际过程会复杂得多。比方说，上述的第 2 步就要包含解析、分词、去重、去噪等等许多步的操作。

另外值得一提的是，搜索的数据现在确实都是增量更新的，可早些年其实并不是如此。Google 在 2003 年以前，爬虫完成不同的数据爬行需要不同的时间，其中最慢的需要几个月，之后的索引需要一周，数据分发也需要一周，在这以后才能被搜索到，因此人们往往也只能搜索到很早以前的数据（当然，那时候互联网的数据没那么大，变更也没那么频繁）。在一次重要更新 [🔗Fritz](#) 以后，爬虫才每天都爬网页的数据，搜索数据也才做到了日更新。

PageRank

纵观上面所述的三个步骤，从功能实现的流程和工程上说，它们各自看起来并没有太大的技术门槛，但是搜索质量却天差地别。其中重要的一项就是怎样对返回给用户的网页进行排名，对于 Google 搜索，这一系列算法中最核心的那个，就叫做 PageRank。

在 PageRank 以前，排序大多依靠对搜索关键字和目标页的匹配度来进行，这种排序方式弊端非常明显，尤其对于善于堆砌关键字“舞弊”的页面，很容易就跳到了搜索结果的首页。但是这样的页面对于用户来说，价值非常小。

PageRank 算法的本质，就是利用网页之间的关联关系来确定网页的影响力权重。而这个关联关系，就是网页之间的超链接，换言之，如果一个页面被各种其它页面引用，特别是被“重要”的网站和页面引用，这就说明这个页面的权重更高。

在实际搜索的时候，**需要做到两个因素的平衡：一个是 Reputation，也就是上面说的这个影响力，它并不会因为用户单次搜索的关键字不同而改变；还有一个是 Proximity，也就是接近程度，这是根据用户搜索的关键字的匹配程度来确定返回网页的。**

如果只考虑 Reputation，那么所有用户搜到的东西都是一样的，这就不是一个搜索引擎了，而是一个网页的有序列表；如果只考虑 Proximity，那么用户搜到的东西就是杂乱无章的匹配页面，而不是像现在这样，将“重要”的页面显示在前面。无论是百度还是 Bing，**不同搜索服务的算法不同，但都是立足于做到这两个基本因素的制衡。**

SEO 相关技术

下面我们再来从工程实现的角度一窥 SEO 技术。技术是要为业务目的服务的，我们最大的目的，是为了让网站的内容更真实、更合理地暴露在搜索引擎的搜索结果中。

1. 白帽和黑帽

当我们明确了上述的目的，遵循搜索引擎规则，通过正当和高效的技术途径来实现 SEO 的效果，这样的方法叫做**白帽 (White Hat) 法**。相应的，如果是通过作弊、欺骗这样的手段，就叫做**黑帽 (Black Hat) 法**。如果你了解过网络安全中的白帽和黑帽，那么这里的含义其实是一致的。

搜索引擎在评估网站前文所述的影响力的时候，有许许多多不同的“**Ranking Signal**”，它指的就是会影响返回的网页排序的“信号”，它们共同决定了一个页面的影响力，对于

Google 搜索来说，**前面我们提到的 PageRank，只是其中之一**。这里面大多数的信号，都可以应用相应的 SEO 规则来进行优化，我随便举几个例子：

网站的正常运行时间。比方说，如果一个站点，在爬虫爬取的时候总是遭遇 4xx、5xx 这样的错误，显然对影响力是一个负面的加权。

网站的年龄，网页内容的新鲜程度，好的原创内容总是最好的优化方式。

网站采用 HTTPS 还是 HTTP，显然 HTTPS 要更优。


HTML 代码的质量，是否存在错误。

网页在站点访问的深度。

当然，黑帽法我们也来简单了解几个。

关键字堆砌：说白了就是放置大量的甚至和网页内容无关的关键字，比方说在页面上放置一些无关的关键字，并将它们的样式设置为透明，这样用户看不见，但是搜索引擎就以为这个页面和这些额外的关键字有关，在一些本该无关的搜索中会增加曝光度。**这其实就是给搜索引擎和用户看不同的页面**，搜索引擎看的页面堆砌了大量的无关关键字，而用户看到的才是正常的网页，这种方法被称为 Cloaking。

还有一种方法叫做 Doorway Pages，这种技术则是创建一个堆砌关键字的临时页面，用户访问的时候，则自动转向正常的网页，或是主页。你可以看到这些黑帽技术，都是为了糊弄搜索引擎而添加了某些本不该出现在页面里的关键字。

链接农场 (Link Farm)：将网站链接放到很多本不该进行外链的其它网页面上，比如花钱买一些不相关的内容，强行建立外链。不知道你有没有听说过 “ Google 轰炸”，它本质上就属于这种方法。当年人们搜索 “more evil than Satan”（比撒旦还邪恶）的时候，结果的第一条居然出现了微软的主页。

Article Spinning：这种技术将一些其它网站已有的内容拷贝过来，做一些用来欺骗搜索引擎的修改，让搜索引擎以为是一份新的内容。比如，替换一些特定的词语、句子，添加一些毫无意义的用户不可见的内容，等等。

2. 站内优化和站外优化

SEO 的优化方式，可以大致分为站内的和站外的。站内优化，其实指的就是在自己管理的网站内部做优化工作来实现 SEO。比如我们之前反复提到的关键字，现在，我们不妨动手来体会一下。

在浏览器地址栏中输入 [🔗https://time.geekbang.org](https://time.geekbang.org)，打开极客时间的页面，右键点击页面空白处并查看网页源代码，你会看到这样的 meta 标签：

📄 复制代码

```
1 <meta name=keywords content= 极客时间,IT, 职业教育, 知识付费, 二叉树, 极客 Live, 极客
```

这就是极客时间网站的关键词，这些关键词会让搜索引擎用户在搜索的时候准确地找到这个网站。除了 keywords 的 meta 标签，还有一些其它起到帮助搜索引擎更准确地认识网站的 HTML 标签，比如 description 的 meta 标签，title 标签等等。对于 HTML 的正文，你也许还记得我们在 [🔗\[第 17 讲\]](#) 介绍的 HTML 语义化标签，它们都可以帮助搜索引擎更好地理解内容。

正如其名，站外优化则和站内优化相反，优化工作是在目标站之外开展的，比如众所周知的“友情链接”，就是一种提供外链的站外优化方式。

3. robots.txt

“robots.txt” 是网站根目录下直接能够访问到的文本文件，它是一个对于网络爬虫的规约，告诉它这个网站下哪些内容你是可以爬取的，哪些内容你是不能爬的。值得注意的是，**robots.txt 不是标准，也不是规范，而是一种“约定俗成”**，几乎所有的搜索引擎都会遵守它。

这就好像你在家门口贴了张条，哪些过路人可以敲你家的门，而哪些人不可以，那么路过的人大多会按这张纸条上的要求去做，但如果你不受欢迎而硬要去敲门（访问），那么也没有任何人可以阻止你，但至于主人开不开门（是否响应请求），或者给不给好脸色（是否返回正常结果），就是另一回事了。

现在，你可以打开浏览器，在浏览器中输入 [🔗https://www.google.com/robots.txt](https://www.google.com/robots.txt) 来访问 Google 的 robots.txt 文件。你将看到如下信息：

```
1 User-agent: *
2 Disallow: /search
3 Allow: /search/about
4 Allow: /search/static
5 ...
6 Disallow: /imgres
7 ...
8 (省略大量 Disallow 和 Allow 的配置)
9
10 User-agent: Twitterbot
11 Allow: /imgres
12
13 User-agent: facebookexternalhit
14 Allow: /imgres
15
16 Sitemap: https://www.google.com/sitemap.xml
```

这是说，对于默认的爬虫（User-agent 为 *），/search 和 /imgres 是不允许爬取的，但是 /search/about 和 /search/static 是可以爬取的，请注意 Allow 指令比 Disallow 有更高的优先级；对于 Twitter 和 Facebook 的爬虫，却是允许访问 /imgres 的。

你可以看到，这样的配置是运行配置默认值，然后通过特殊值来覆写的（不知这能否让你回想起 [🔗\[第 28 讲\]](#) 中介绍的类似的“默认值 + 特殊值覆写”的配置方式）。最后一行是网站地图 sitemap.xml 的位置，我们下面会讲。

另外，如果你想让搜索引擎友好一点，就不要那么频繁地访问你的网站，你可以使用 Crawl-delay 参数，用来告知连续的请求之间至少间隔多少秒，比如：

```
1 Crawl-delay: 5
```

同样的，你可以看看百度的 robots.txt，访问 [🔗https://www.baidu.com/robots.txt](https://www.baidu.com/robots.txt)，你会看到百度比较“特立独行”，它不允许 Google、有道、搜狗等多家搜索引擎的数据爬取。

除了全站的搜索引擎爬取设定以外，能够按页来设置吗？可以，这时候你需要使用一个名为 robots 的 meta 标签，这个标签在 HTML 的 head 内，用来告知该页的爬取策略。

```
1 <meta name="robots" content="noindex,nofollow" />
```

除页面以外，HTML 的 a 标签（链接）也能够告诉搜索引擎不要进一步追踪爬取，方法就是使用 nofollow，如下：

```
1 <a href="http://www.another-website.com/" rel="nofollow"> 另一个站点 </a>
```

因此，是否允许爬取的建议，是可以在网站、页面和链接这三个级别分别设置的。

4. 网站地图

网站地图就像前面提到的 robots.txt 一样，是另一个和搜索引擎对话的途径。网站可能非常大，爬取一遍耗时长，但**网站地图则可以清晰直接地告诉搜索引擎网站内“重要”的页面都有哪些（无论是否被链接指向），它们的更新习惯，包括最近一次是什么时候更新的，更新频率是多少，以及对于整个网站来说，不同页面的重要性比重是多少。**

对于使用 SPA（我们曾在 [\[第 17 讲\]](#) 介绍过 SPA，你可以回看）的网站应用来说，由于缺乏页面跳转，搜索引擎无法正确理解页面的关系、更新、指向等等，网站地图就显得更为重要了。

这次我来拿 B 站举个例子，访问 <https://www.bilibili.com/sitemap.xml>，你会看到如下的内容：

```
1 <sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
2   <sitemap>
3     <loc>http://www.bilibili.com/sitemap/v.xml</loc>
4     <lastmod>2019-10-26T18:25:05.328Z</lastmod>
5   </sitemap>
6   ...
7 </sitemapindex>
```

它是由多个子 sitemap 配置文件组成的，随便打开一个，比如

🔗 <http://www.bilibili.com/sitemap/v.xml>:

📄 复制代码

```
1 <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
2   <url>
3     <loc>http://www.bilibili.com/v/douga/</loc>
4     <lastmod>2019-10-26T18:25:17.629Z</lastmod>
5     <changefreq>daily</changefreq>
6   </url>
7   ...
8 </urlset>
```

可以说一目了然，页面位置、上次修改时间，以及修改频率。这可以让搜索引擎有目的和有条件地扫描和爬取页面数据。

对于网站地图，除了被动等待爬虫的抓取，搜索引擎服务往往还提供另一种方式来报告网站地图的变更，那就是**允许网站管理员主动去提交变更信息，这种方式 and 爬虫来爬取比较起来，类似于我们从第一章就开始讲的 pull 和 push 的区别**，这种方式对于网站管理员来说更麻烦，但是显然可以更为及时地让搜索引擎获知并收录最新数据。

这种方式从实现上说，就是由搜索引擎服务的提供商开放了一个 Web API，网站在内容变更的时候调用，去通知搜索引擎（关于 Web API 的设计，你可以回看 🔗[第 05 讲]）。

5. 统计分析

在进行 SEO 的改动调整之后，我们需要一些方式来跟踪和评估效果。像 Google Analytics 和百度统计，就提供了这样的功能。

原理上很简单，以 🔗Google Analytics 为例，它会为你的网站生成一段 JavaScript 代码，你就可以把它嵌入每一个你希望得到跟踪的网页。这样，在页面访问时，这段代码会收集相关信息，并向页面嵌入一个大小为 1 像素的 gif 图片，而这个图片的 URL 带有当前浏览器、操作系统等等客户端的不同类型的信息。这样，Google Analytics 就可以捕获这些信息来完成数据统计了。

下面给出了我在 Mac 上访问极客时间的页面时，网页向 Google Analytics 服务器发送的统计信息 URL（别看这个 URL 没有 gif 字样，但这个请求返回的就是一个 gif 图片，这一点可以从响应的 Content-Type 中看出来）：

 复制代码

```
1 https://www.google-analytics.com/collect?v=1&_v=j79&a=775923213&t=pageview&_s=.
```

通过收集这样的信息，可以获得很多网站用户的情况统计，比如访问量、页面停留时间、地区分布、电脑访问或手机访问的比例等等，并能观察这样的统计信息基于时间的走势。

总结思考

今天我们学习了一些互联网搜索引擎的工作机制，并结合例子从工程的角度了解了几个常见的 SEO 相关技术。今天我们就放具体的思考题了，但 SEO 本身是一个可以挖掘很深的领域，我在扩展阅读中放置了一些资料，供你延伸。

好，到今天为止，“寻找最佳实践”这一章就接近尾声了，你是否有所收获、有所体会，欢迎你在留言区分享。

扩展阅读

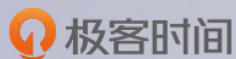
文中介绍了几个典型的 SEO 黑帽法，作为视野的拓展，你可以阅读 [这篇文章](#) 了解更多的黑帽法。特别地，你也可以参阅这一 [词条](#) 了解更多历史上的“Google 轰炸”事件。

如果对 Google Analytics 感兴趣的话，那么官方有一些很好的 [学习材料](#)；如果用的是百度统计，那么你也可以浏览一下官方的 [文档](#)。

对于 PageRank 算法，互联网上其实有很多学习材料，比如维基百科的 [词条](#)，再比如科普作家卢昌海的文章——[谷歌背后的数学](#)。这个算法的来源，是 [The Anatomy of a Large-Scale Hypertextual Web Search Engine](#) 这篇 Sergey Brin 和 Lawrence Page 最早写的关于 Google 搜索引擎原理的论文，当然，它并非这一讲的学习周期内要求的阅读材料，而仅供感兴趣且有余力的你阅读。

[单页应用 \(Single Page Application\)](#) 的搜索引擎优化，专栏第三章已经介绍了 SPA 的优势，但是 SPA 网站并不是一个擅长将喜怒哀乐表现出来的孩子，他对擅长察言观色

的搜索引擎颇不友好，因此要对 SPA 网站进行有效的 SEO，是需要一些特殊技巧的，推荐阅读。



全栈工程师修炼指南

从全栈入门到技能实战

熊燚

Oracle 首席软件工程师



新版升级：点击「👤 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 31 | 防人之心不可无：网站安全问题窥视

下一篇 33 | 特别放送：聊一聊程序员学英语

精选留言 (1)

💬 写留言



許敲敲

2019-11-22

老师你好，我想了解下，我自己在github上搭建一些静态博客，使用google analytic就可以分析我网站的一些被浏览信息是嘛？不知道配置这个麻烦吗，今天下班去翻个墙研究下。

作者回复: 对你说的这几个组合起来，我没有尝试过，但是技术上看，配置 Google Analytics 应该是非常简单的，一小段脚本就可以了。

💬 2



