自制AI图像搜索引擎

明恒毅◎著

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

自制

AI图像搜索引擎

明恒毅◎著

人民邮电出版社
北京

图书在版编目（CIP）数据

自制AI图像搜索引擎 / 明恒毅著. -- 北京 : 人民邮电出版社, 2019.3
ISBN 978-7-115-50401-2

Ⅰ. ①自… Ⅱ. ①明… Ⅲ. ①图象识别－搜索引擎
Ⅳ. ①TP391.41

中国版本图书馆CIP数据核字(2018)第286066号

内 容 提 要

图像搜索引擎有两种实现方式—基于图像上下文文本特征的方式和基于图像视觉内容特征的方式。本书所指的图像搜索引擎是基于内容特征的图像检索，也就是通常所说的“以图搜图”来检索相似图片。本书主要讲解了搜索引擎技术的发展脉络、文本搜索引擎的基本原理和搜索引擎的一般结构，详细讲述了图像搜索引擎各主要组成部分的原理和实现，并最终构建了一个基于深度学习的 Web 图像搜索引擎。

本书适用于对图像搜索引擎感兴趣的广大开发者、程序员、算法工程师以及相关领域研究人员，也适合作为高等院校计算机及相关专业的本科生和图像检索、机器视觉等方向的研究生的参考用书。

◆ 著　　明恒毅
责任编辑　张　爽
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市君旺印务有限公司印刷

◆ 开本：800×1000　1/16
印张：13.5
字数：298 千字　　2019 年 3 月第 1 版
印数：1－2 400 册　　2019 年 3 月河北第 1 次印刷

定价：59.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

序

大约十年前的某一天，我正徜徉在互联网的世界里，忽然一个名叫“TinEye”的图像搜索引擎网站映入我的眼帘。我满怀憧憬地在那个网站中上传了一幅图片，它很快搜索并返回了许多这幅图片在互联网中不同 URL 上的结果。我接着尝试上传了另一幅图片，一会儿它又返回了许多近似这幅图片的结果，很显然，结果中的很多图片是在同一幅图像上修改的。面对如此准确和令人惊艳的结果，我不禁脑洞大开、浮想联翩，构思着一个个可以运用该技术实现的奇思妙想。猛然间，我觉得心中产生了一股强大的力量——我要弄懂它背后的技术原理。

为了彻底弄清楚这类图像搜索引擎的技术原理，我反复查找和阅读当时互联网上甚为稀缺的相关资料，但收效甚微。直到后来，我遇到了一个叫作 LIRE 的开源项目，它让我初步理解了图像搜索引擎的技术原理。但是在实际应用中，LIRE 的效果并不是太好。为了解决这个问题，我又找到“深度学习”这个强有力的助手。在探索原理的过程中，我发现国内几乎找不到一本介绍图像搜索引擎基本原理和实现的书，这也成了本书诞生的缘由。

基于内容的图像检索技术自 20 世纪 90 年代提出以来，得到了迅速的发展。研究人员提出了不同的理论和方法，其中具有代表性的是 SIFT、词袋模型、矢量量化、倒排索引、局部敏感散列、卷积神经网络，等等。与此同时，产业界也推出了许多实用的图像搜索引擎，比如 TinEye、谷歌图像搜索、百度图像搜索和以淘宝为代表的垂直领域图像搜索引擎。但是到目前为止，此项技术还远未完全成熟，还有许多问题需要解决，改进和提高的空间还很大。搜索的结果和用户的期望还有一些距离，存在一定的图像语义鸿沟。这也是从事这项技术研究与开发的人员不断进步的源动力。

希望本书的出版能够在一定程度上缓解图像搜索引擎资料稀少的现状，并能够吸引和帮助更多的技术人员关注并研究图像检索技术。

明恒毅

2018 年 11 月

前　言

得益于基于内容的图像检索技术的发展，近十年来互联网业界涌现出一些以 TinEye 图像搜索、淘宝图像搜索为代表的通用和垂直领域图像搜索引擎。这些图像搜索引擎改变了以往单一的关键字检索方式，极大地满足了人们日益多样的图像检索需求。作者在研究图像搜索引擎的过程中发现，目前国内尚无一本系统论述图像搜索引擎原理与实现的书籍，因此产生了撰写本书的想法。

本书内容共分为 5 章。

第 1 章由文本搜索引擎的原理讲起，逐步抽象出搜索引擎的一般结构，引领读者由文本搜索过渡到图像搜索。

第 2～3 章分别按照传统人工设计和深度学习两种方式对图像特征提取的相关理论和方法进行讲解。

第 4 章详述了图像特征索引和检索的相关理论和方法。

上述每一章都在阐述相关理论和方法的同时，使用基于 Java 语言的实现代码和详实的代码注释对理论和方法进行复述。力求使读者不但能够理解深奥的理论知识，而且能将理论转换为实际可运行的程序。

第 5 章会带领读者从零开始逐步构建一个基于深度学习的 Web 图像搜索引擎，使读者能够更透彻地理解图像检索的理论，并具有独立实现一个在线图像搜索引擎的能力。

图像搜索引擎技术涵盖知识面广，目前尚在不断发展中，由于作者水平所限，书中难免存在错误和不足之处，欢迎各位读者批评指正。反馈意见和建议可以通过加入本书 QQ 群（743328332）进行沟通交流，或致信邮箱 imgsearch@126.com，我将不胜感激。

在这里，我要感谢人民邮电出版社编辑张爽的邀请，通过写作此书，我感受到了技术写作的不易与乐趣，也得到了一次难得的提升能力的机会。还要感谢父母妻儿对我的支持和理解，以及生活上的照顾，正是有了他们的支持，才能让我能够心无旁骛、安心写作。

资源与支持

本书由异步社区出品，社区（https://www.epubit.com/）为您提供相关资源和后续服务。

配套资源

本书提供配套源代码，请在异步社区本书页面中单击 配套资源 ，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。

与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“**异步社区**”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 https://www.epubit.com。

“**异步图书**”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。

异步社区

微信服务号

目　录

第1章 从文本搜索到图像搜索

1.1 文本搜索引擎的发展[1]

1990年，加拿大麦吉尔大学的Alan Emtage等学生开发了一个名叫Archie的系统。该系统通过定期搜集分析散落在各个FTP服务器上的文件名列表，并将之索引，以供用户进行文件查询。虽然该系统诞生在万维网的出现之前，索引的内容也不是现代搜索引擎索引的网页信息，但它采用了与现代搜索引擎相同的技术原理，因此被公认为现代搜索引擎的鼻祖。

1991年，明尼苏达大学的学生Mark McCahill设计了一种客户端/服务器协议Gopher，用于在互联网上传输、分享文档。之后产生了Veronica、Jughead等类似于Archie，但运行于Gopher协议之上的搜索工具。

同一时期，英国计算机科学家Tim.Berners.Lee提出了将超文本和Internet相结合的设想，并将之称为万维网（World Wide Web）。随后，他创造了第一个万维网的网页，以及浏览器和服务器。1991年，他将该项目公之于众。自此，万维网成为了Internet的主流，全球进入了丰富多彩的WWW时代。搜索引擎也逐步从FTP、Gopher过渡到了万维网，并进一步演进。

1993年，麻省理工学院的学生Matthew Gray开发了第一个万维网spider程序WWW Wanderer，它可以沿着网页间的超链接关系对其进行逐个访问。起初，WWW Wanderer只是用来统计互联网上的服务器数量，后来加入了捕获URL的功能。虽然它功能比较简单，但它为后来搜索引擎的发展提供了宝贵的思想借鉴。这一构思激励了许多研究开发者在此基础上进行进一步改进和扩展，并将spider程序抓取的信息用于索引构建。我们今天在开发一个网站或做搜索引擎优化时所用到的robot.txt文件，正是告诉spider程序可以爬取网站的哪些部分，不可

[1] Michael Busby. Learn Google: Wordware Publishing, Inc., 2003

以爬取哪些部分的一份协议。同年，英国 Nexor 公司的 Martin Koster 开发了 Aliweb。它采用用户主动提交网页简介信息，而非程序抓取的方式建立链接索引。是否使用 robot、spider 采集信息也形成了搜索引擎发展过程中的两大分支，前者发展为今天真正意义上的搜索引擎，后者发展为曾经风靡一时，能够提供分类目录浏览和查询的门户网站。

1994 年可以说是搜索引擎发展史上里程碑的一年。华盛顿大学的学生 Brain Pinkerton 开发了第一个能够提供全文检索的搜索引擎 WebCrawler。而在此之前，搜索引擎只能够提供 URL 或人工摘要的检索。自此，全文检索技术成为搜索引擎的标配。这一年，斯坦福大学的杨致远和 David Filo 创建了大家熟知的 Yahoo，使信息搜索的概念深入人心，但其索引数据都是人工录入的，虽能提供搜索服务，但并不能称之为真正的搜索引擎；卡耐基梅隆大学的 Michael Maldin 推出了 Lycos，它提供了搜索结果的相关性排序和网页自动摘要，以及前缀匹配和字符近似，是搜索引擎的又一历史性进步；搜索引擎公司 Infoseek 成立，在其随后的发展中，它首次允许站长提交网址给搜索引擎，并将“千人成本”（Cost Per Thousand Impressions，CPM）广告模式引入搜索引擎。

1995 年，一种全新类型的搜索引擎——元搜索引擎诞生了，它是由华盛顿大学的学生 Eric Selburg 和 Oren Etizioni 开发的 MetaCrawler。元搜索引擎采用将用户的查询请求分发给多个预设的独立搜索引擎的方式，并统一返回查询结果。但是由于各独立搜索引擎搜索结果的打分机制并不相同，常常返回一些不相干的结果，精准性往往并不如独立搜索引擎好，因此元搜索引擎始终没有发展起来。

同一年，DEC 公司开发了第一个支持自然语言搜索及布尔表达式（如 AND、OR、NOT 等）高级搜索功能的 AltaVista。它还提供了新闻组搜索、图片搜索等具有划时代意义的功能。

1998 年，斯坦福大学的学生 Larry Page 和 Sergey Brin 创立了 Google（谷歌）——一个日后影响世界的搜索引擎。Google 采用了 PageRank（网页排名）的算法，根据网页间的超链接关系来计算网页的重要性。该算法极大地提高了搜索结果的相关性，使其后来居上，几乎垄断了全球搜索引擎市场。

1.2 文本搜索引擎的结构与实现

目前，基于文本信息的搜索引擎虽然还有一定的提升空间，但其工作原理已经相对稳定，基本结构也已趋于成熟。文本搜索引擎基本可以分为抓取部分、预处理部分、索引部分、搜索部分以及用户接口，如图 1-1 所示。

由于抓取部分不是本书所讨论的内容，故不做详细介绍。下面来着重介绍一下文本数据预处理、索引及搜索。

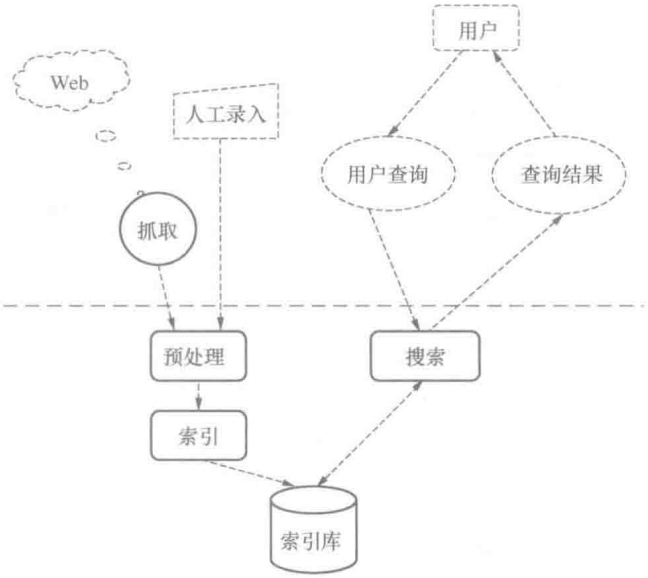

图 1-1 文本搜索引擎结构

1.2.1 文本预处理

蜘蛛程序（Spider）抓取的数据在进行一定程度的预处理之后才能用于索引的建立。文本数据预处理主要是为了提取词语而进行的文本分析，而文本分析又可分为分词、语言处理等过程。

1. 分词

文本分词过程通常分为三步：第一步，将文本分为一个个单独的单词；第二步，去除标点符号；第三步，去除停止词（Stop words）。停止词是语言中最普通的一些单词，它们的使用频率很高，但又没有特殊意义，一般情况下不会作为搜索关键词。为了减小索引的大小，一般将此类单词直接去除。为方便读者理解，下面举例说明，如图 1-2 所示。

2. 语言处理

语言处理主要对分词产生的词元进行相应语言的处理。以英文为例：首先将词元变为小写，然后对单词进行缩减。缩减过程主要有两种，一种被称为词干提取（Stemming），另一种被称为词形还原（Lemmatization）。词干提取是抽取词的词干或词根，词形还原是把某种语言的词汇还原为一般形式。两者依次进行相关语言处理，比如将 books 缩减为 book（去除复数形式），将 tional 缩减为 tion（去除形容词后缀）。词干提取采用某种固定的算法进行缩减。词形还原通常使用字典的方式进行缩减，缩减时直接查询字典，比如将 reading 缩减为 read（字典中存在 reading 到 read 的对应关系）。词干提取和词形还原有时会有交集，同一个词，使用两种方式都会得到同样的缩减。接上面的举例，继续说明，如图 1-3 所示。

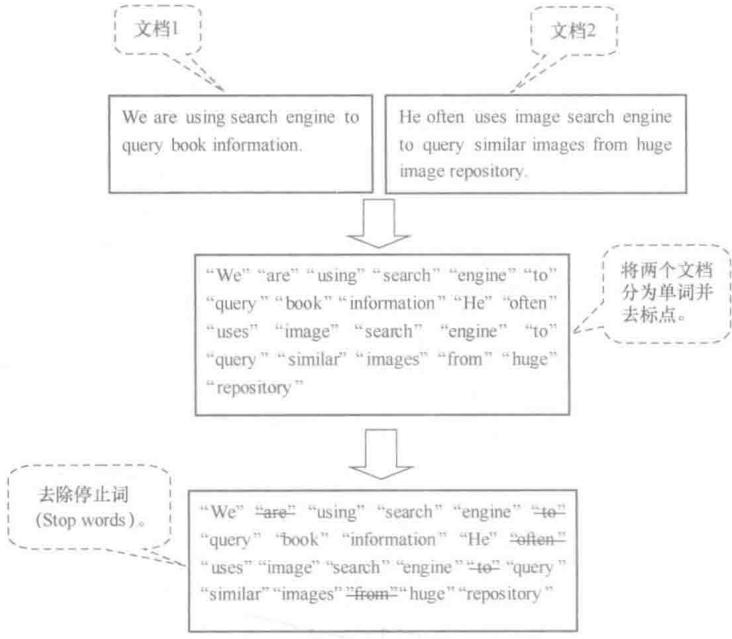

图 1-2　文本预处理

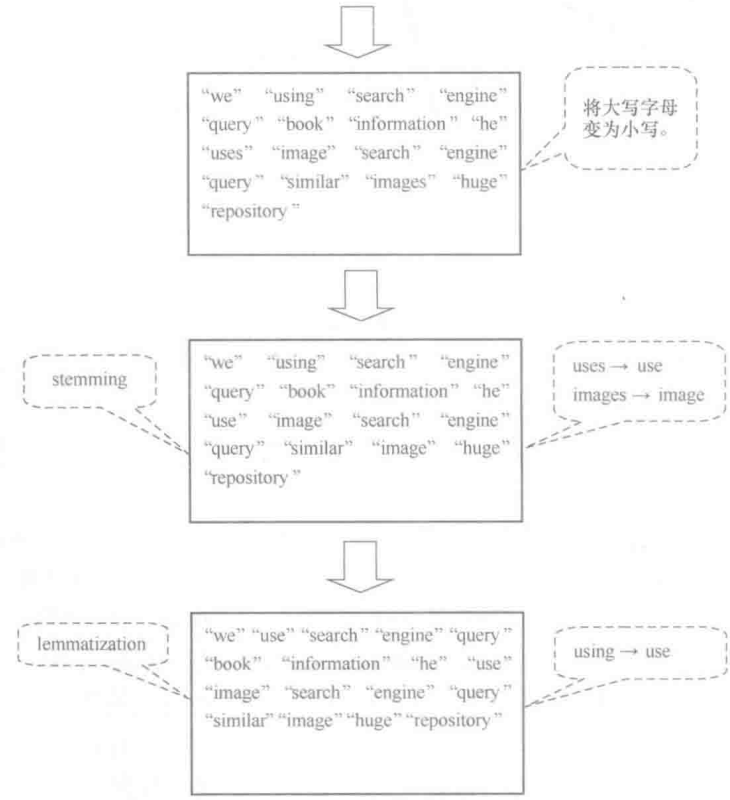

图 1-3　语言处理

1.2.2 建立索引

经过文本分析后，得到的结果称为词（Term），我们利用它建立索引。首先使用得到的词创建一个字典，然后对字典按字母顺序进行排序，最后合并相同的词，形成文档倒排表（Posting List），具体过程如下。

1. 使用词生成字典，如表 1-1 所示

表 1-1 使用词生成字典

词	文档 ID
we	1
use	1
search	1
engine	1
query	1
book	1
information	1
he	2
use	2
image	2
search	2
engine	2
query	2
similar	2
image	2
huge	2
repository	2

2. 对字典按字母顺序排序，如表 1-2 所示

表 1-2 对字典按字母顺序排序

词	文档 ID
book	1
engine	1
engine	2
he	2
huge	2
image	2
image	2

续表

词	文档 ID
information	1
query	1
query	2
repository	2
search	1
search	2
similar	2
use	1
use	2
we	1

3. 合并相同的词，形成文档倒排链表

在文档倒排表中，有几个概念需要解释一下。文档频率（Document Frequency）表示共有多少个文档包含这个词。词频率（Term Frequency），表示这个文档中包含此词的个数。在图 1-4 中，左边是按字母顺序排序的字典合并相同词，并统计出该词在文档中出现次数的结果。中间和右边是文档 1 和文档 2 中包含某个词的次数——词频率。它们之间是用链表的形式串起来的，又因为是根据词的值来查找相关文档的，而非在文档中查找相关的值，和正常顺序是相反的，故称其为文档倒排链表或倒排索引。

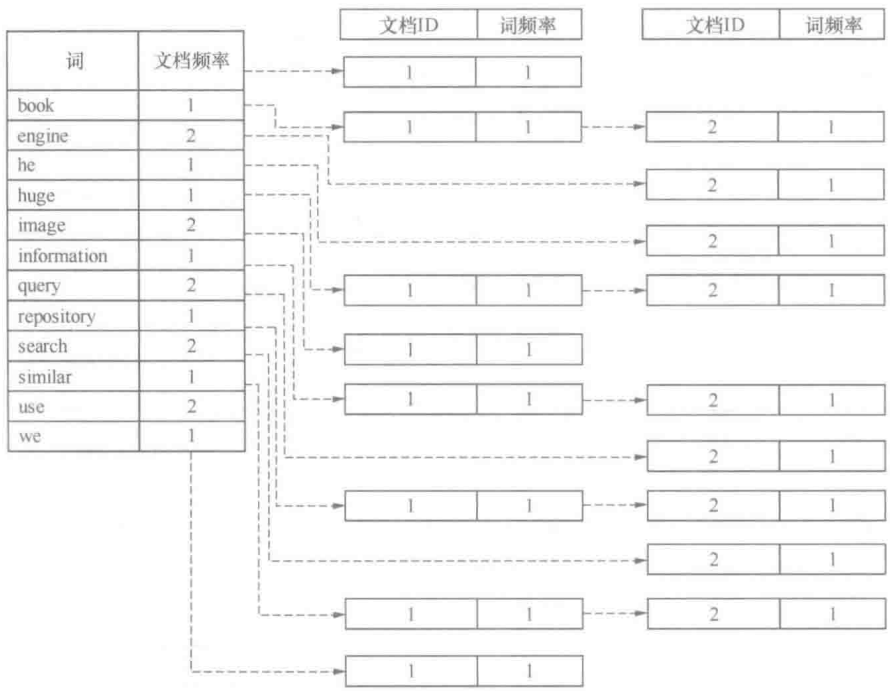

图 1-4　文档倒排链表

至此，索引已经构建好了。根据以上的文档倒排链表，我们就能使用关键词来查到相应的文档了。

1.2.3 对索引进行搜索

上面我们已经可以查找到包含关键词的相关文档了，但它还不能满足实际搜索的要求。如果结果只有几个，当然没有问题，全部显示就是了。但在实际应用中，搜索引擎需要返回几十万，甚至百万、千万级的结果。我们怎样才能将最相关的文档显示在最前面呢？这也是下面需要探讨的问题。

1. 用户输入查询语句

目前，搜索引擎均提供自然语言搜索以及布尔表达式高级搜索，所以查询语句也是遵循一定的语法结构。比如我们可以输入查询语句“search AND using NOT image”，它搜索包含 search 和 using 但不包含 image 的文档。

2. 对查询语句进行词法分析、语法分析、语言处理

词法分析用来提取查询词以及布尔关键字，上面的查询语句提取出的查询词为 search using image，布尔关键字是 AND 和 NOT。语法分析会将词法分析的提取结果生成一棵语法树。上例形成的语法树如图 1-5 所示。

语言处理与创建索引时的语言处理过程几乎相同。如图 1-6 所示，上例中的 using 将转换为 use。

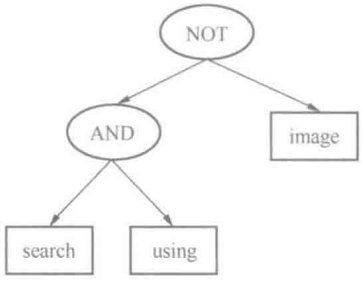

图 1-5 语法树

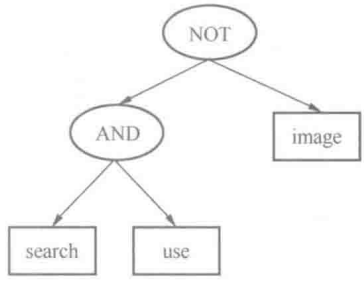

图 1-6 语言处理后的语法树

3. 搜索索引，返回符合上述语法树的结果

首先，在反向索引中分别找出包含 search、use 和 image 的文档链表。然后，将包含 search 和 use 的文档链表合并，得到既包含 search，又包含 use 的文档链表。接着，在上一步的结果中去除包含 image 的文档链表，最终的文档链表就是符合上述语法树的结果。

4. 对结果进行相关性排序

虽然在上一步中我们得到了想要查找的文档，但这些文档并未按照与查询语句的相关性进