第2讲 网络爬虫

2.1 爬虫定义

一种自动获取网页内容的程序，是搜索引擎的重要组成部分。通俗的讲，也就是通过HTML源码解析来获得想要的内容。

2.2 爬取过程

从一个或若干初始网页的URL开始，直到满足系统的一定停止条件。

爬虫从一个或若干个网页的URL开始，抽取URL放入 _队列__ 中

2.3 爬取框架

2.3.3 URL判重

访问标记

- 由于搜索引擎在爬取时要访问大量的网页，因此在查找网址l是否访问过及标记网址l已经访问时为了提高查找和访问效率通常建立一个散列，其中存放访问过每一个网址
- 为了减少这个散列表所占用的空间, 通常在其中存放网址经过散列函数（如MD5、SHA-1等）计算出的对应的固定长度的散列值
- 这样便可以在平均情况下O(1)的时间内查找和更新占用O(n)空间的网址列表（n为已访问的网址数目）。
    - 直接寻址法

2.4 爬虫必须具有的功能

2.4.1 礼貌性

Web服务器有显式或隐式的策略控制爬虫的访问

- 隐式的礼貌:即使没有特别的说明，也不应该频繁的访问同一个网站
- 显式的礼貌: 根据网站站长的说明，选择允许爬取的部分进行爬取
- 只爬允许爬的内容、尊重 robots.txt
- **Robots.txt**对爬取过程进行限制

2.4.2 鲁棒性

能从采集器陷阱中跳出，能处理Web服务器的其他恶意行为

2.4.3 性能和效率

- 充分利用不同的系统资源，包括处理器、存储器和网络带宽
- 优先抓取“有用的网页”
- 搜索策略：深度优先, 广度优先

理论上，两者能够在大致的时间里完成所有的整个静态网页的爬取工作。

- BFS优于DFS
    - 工程上，网络爬虫更应该定义为“如何在有限的时间里最多的爬下那些重要的网页”，
    - 我们一般认为一个网页的首页是相当重要的。
- DFS优于BFS
    - 爬虫的分布式结构以及网络通信的握手成本有关，“握手”就是指下载服务器与网站的服务器建立通信的过程。
    - 时间网络爬虫是由成百上千万太服务器组成的分布式系统，对于某一个网页，一般由特定的一台或者几台服务器专门下载，这样可以避免握手次数太多。
- 实际应用的网络爬虫不是对网页次序的简单BFS或者BFS，而是一个相对复杂的下载优先级排序的方法，管理这个系统的叫做“调度系统”(Scheduler)，会有一个Priority Queue。BFS成分更加多一些。
- DFS 要限定爬取的深度
    - 在爬取时为了防止有些错误链接导致的无穷递归爬取，需要限定爬取的深度。
    - 此外层次越深的网页对用户来说可用的信息越少，理论上呈对数的倒数关系。

2.4.4 分布式

可以在多台机器上分布式运行

1. 分布式带来的问题
   哈希表判重，在一台下载服务器上建立和维护一张哈希表并不是难事，分布式，多台服务器一起下载网页，就会出现问题：

- 问题1、哈希表太大，一台下载服务器存不下。
- 问题2、每台下载服务器在开始下载前和完成下载后都要维护这表哈希表，这个存储哈希表的通信就成为爬虫系统的瓶颈。

2. 解决方法：
   A、明确每台下载服务器的分工，即一看到某个URL就知道交给哪台服务器去执行
   B、批量处理，减少通信的次数

2.4.6 新鲜度

对原来抓取的网页进行更新

2.5 爬虫分类

2.5.1基于整个Web的信息采集(Universal Web Crawling)

2.5.2 增量式Web信息采集 (Incremental Web Crawling)

2.5.3 基于主题的Web信息采集(Focused Web Crawling)

2.5.4 基于用户个性化的Web信息采集(Customized Web Crawling)

开源工具：

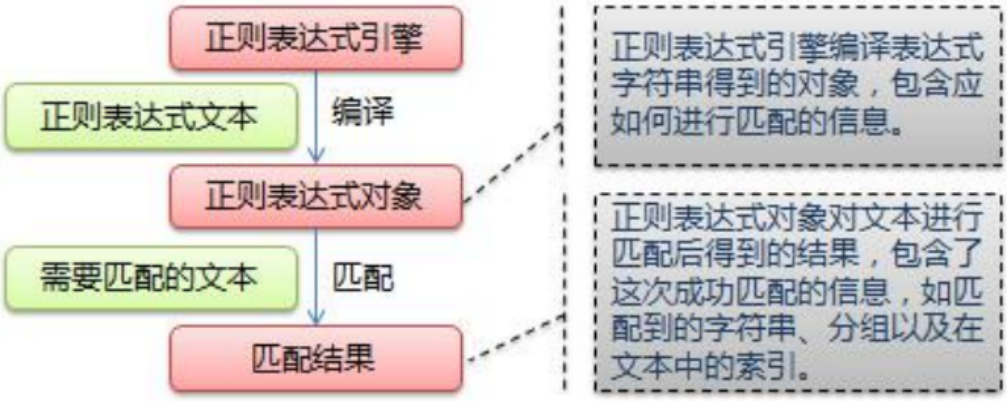
Nutch

第3讲 网页分析技术

- 对于HTML文档，有两种看待方式:
  - 一种是将文档看作字符流;
    - 正则表达式
  - 一种是将文档看作树结构。
    - 基于DOM

3.1 正则表达式

- 又称规则表达式
- 正则表达式是对**字符**串操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“**规则字符串**”，这个“规则字符串”用来表达对字符串的一种**过滤逻辑。**
- 正则表通常被用来检索、替换那些符合某个模式(规则)的文本。
- 是由一组普通字符和一组元字符组成的字符串，用来表示符合一定模式的一组字符串，常用于字符串处理，表单验证等场合，表示能力与正规文法相同。

- 

- Python re模块

3.2 基于HTML DOM提取内容

3.2.1 文档对象模型（document object model，DOM）

DOM将一个XML文档转换成一个对象集合，然后可以任意处理该对象模型。这一机制也称为“随机访问”协议，可以在任何时间访问数据的任何一部分，然后修改、删除或插入新数据。
DOM**将HTML视为树状结构**的元素，所有元素以及他们的文字和属性可通过DOM树来操作与访问。

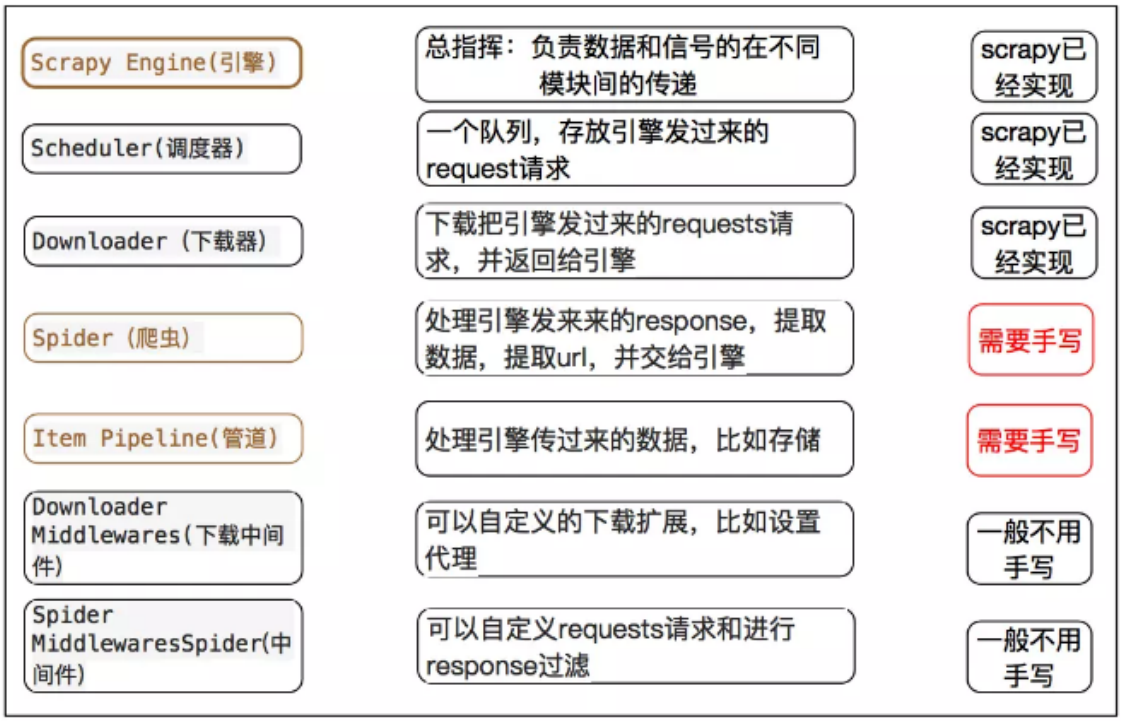
3.3 Beautiful Soup模块

- python的一个模块
- Beautiful Soup提供一些简单的、python式的函数用来处理导航、搜索、修改分析树等功能。
- 它是一个工具箱，通过解析文档为用户提供需要抓取的数据。
- 简单，不需要多少代码就可以写出一个完整的应用程序。
- bs4库将任何读入的html文件或字符串都转换为**utf-8编码**

3.4 python爬虫框架 Scrapy

快速、高层次的屏幕抓取和web抓取框架，用于抓取web站点并从页面中提取**结构化**的数据。
Scrapy吸引人的地方在于它是一个框架

- 所谓“框架”，便是整个或部分系统的可重用设计。
- 在python中也可以说，一个框架就是一个可复用的“巨大模块”。
- 任何人都可以根据需求方便的修改。
- 借助Scrapy框架这个爬虫利器，只需根据自己的需要，编写几个专属的模块就可以轻松地实现一个爬虫项目



第4讲 爬虫与网站的博弈

网站反爬后端策略

- 网页在后端拦截
  - User-Agent + Referer检测
  - 账号及Cookie验证
  - 验证码
  - IP限制频次
    网站反爬前端策略
- 网页在前端显示不一致
  - 懒加载
  - FONT-FACE拼凑式： 猫眼电影里，对于票房数据，展示的并不是纯粹的数字。必须同时爬取字符集，才能识别出数字

第5讲 数据抽取与包装器（总结性）

5.1 WEB数据抽取

5.1.3 Web数据抽取

Web 数据抽取是指从页面中将用户感兴趣的数据利用程序自动抽取到本地的过程。

基于包含数据的页面是由通过页面模板生成的这个基本前提，Web数据抽取被定义为其逆过程，即反向获取页面中数据的过程。解决该问题的关键在于逆向推导页面模板，即找到大量页面中的共同部分、格式约定和页面数据模式。

为了能够保证抽取的准确性，必须要能够识别页面模板。

定义(页面模板) 页面模板T= < C，L，S>

- 是不完全的页面，
- 它是相似页面之间不变的部分，

- 用来按照结构化数据生成供用户浏览的页面。
- 包含了一些共同的页面内容C、严格定义的格式L和既定的页面数据模式S.
    - C包含了导航、版权声明、固定页面修饰等这些不变的内容;
    - L包含了页面数据的格式规范;
    - S则是能够从页面数据中观察到的模式

定义 (Web数据抽取) 给定页面集合W={wi}，它通过页面模板T生成，包含数据D={di}，即W={wi| wi =T(di))，Web数据抽取问题则可以定义为通过一定的技术手段，从W中逆向推导出T，还原数据D。

抽取方式

- 手工爬虫
- 机器学习爬虫　训练集 T　测试集

5.2 Web数据抽取方法

Web数据抽取的**目的**是获得页面中的**数据**，需要**借助一个或多个页面逆向推导出页面模板T。**
包装器(Wrapper) 是一种**软件过程**，这个过程使用已经定义好的信息抽取规则，将网络中Web页面的信息数据抽取出来，转换为用特定的格式描述的信息。一个包装器一般针对某一种数据源中的一类页面。包装器运用规则执行程序对实际要抽取的数据源进行抽取。

在人们研究Web 数据抽取的过程中，从自动化程度来区分，形成了

- 人工抽取。
    - **人工分析出页面模板**
- 半自动抽取。
    - **由计算机应用页面模板抽取数据生成具体包装器，而页面模板的分析仍然需要人工参与。**
- 自动抽取。
    - 仅仅需要很少的人工参与(例如检查结果进行校准) 或者完全不需要人工参与，

5.3 Web 数据抽取评价标准

对某个测试参考集，**对应的相关数据集合为R。**假设用某个检索策略进行处理后，**得到一个结果集合A。**
令Ra表示R与A的交集。
召回率(Recall)：也称为查全率，指**抽取到的正确结果与要抽取页面的全部结果的比。**即R=|Ra| / |R|
召回率是用来衡量一项抽取技术能否将所有要抽取出的数据都抽取出来的指标。

准确率(Precision)：也称为查准率，指**抽取到的正确结果与抽取到的全部结果的比。**即P=|Ra| / |A|
准确率用来衡量一项抽取技术抽取出的数据是否符合预期指标。

两个指标分别衡量了系统的某个方面，但是也为比较带来了难度
解决方法：将两个指标融成一个指标——F值(F-measure)

- 准确程度
- 抽取自动化程度：这项标准用来衡量用户在抽取过程中的参与程度，分为手工、半自动和全自动三类。
- 适应性：指在页面的内容和结构发生较小**变化**的情况下，该抽取方法或工具具有自适应能力，仍然能够继续正常工作。
- 修正率：其含义是需要手工调整使得准确率和召回率达到100%的Web数据库数量。该评价标准对于开发实际的Web数据抽取应用是非常重要的。

5.4 包装器

包装器是针对某一类特定的网页、计算机可以理解并执行的**程序或抽取规则**，其任务就是负责**将HTML格式的数据抽取并转化为结构化的格式。**

包装器是Web数据抽取系统的重要组成部分之一，特别是在半自动化抽取系统中，需要通过和用户的交互生成。

包装器的**核心**是抽取规则。

- **抽取规则**是基于HTML文档格式的，用于从每个HTML文档中抽取相关信息。
- 相应地，抽取规则也可以分为：
    - **基于分界符(或界标符)的规则**
    - **基于树路径的规则**

5.4.1 基于分界符的规则

基于分界符的规则将HTML文档看作字符流，给出数据项的**起始**和**结束分界符**，将其中的数据抽取出来。

5.4.2 基于树路径的规则

基于树路径的规则是将文档看作一个树结构。所抽取的数据存储在树节点中，因而可**根据其路径来定位。**

- 首先根据HTML标签将文档分析成**树结构**，如DOM树，
- 然后通过规则中的**路径**在树中搜索相应的节点，
- 最终得到所需数据。

5.4.3 包装器与爬虫软件的不同

包装器是一个能够将数据从HTML网页中抽取出来，并且将他们还原为结构化的数据的软件程序。

包装器归纳 wrapper induction

- 使用机器学习的方法产生抽取规则
- 基于有监督学习的，
    - 从标注好的训练样例集合中学习数据抽取规则
    - 用于从**其他**相同标记或相同网页模板抽取目标数据

步骤

- 网页清洗
- 网页标注
- 包装器空间的生成
- 包装器评估：准确率和召回率

自动抽取：

- 通过挖掘多个数据记录中的重复模式来寻找这些模板
- 无监督学习
- 步骤
    - 1.包装器训练：
        - 将一组网页通过聚类将相似的网页分成若干个组，
        - 每组相似的网页将获得不同的包装器。
    - 2.包装器应用：
        - 将需要抽取的网页与之前生成包装器的网页进行比较，
        - 在某个分类下则使用该分类下的包装器来获取网页中的信息。

	手工方法	包装器归纳	自动抽取
优点	1. 对于任何一个网页都是通用的，简单快捷； 2. 能抽取到用户感兴趣的数据。	1. 需要人工标注数据集； 2. 能够抽取到用户感兴趣的数据； 3. 可以运用到大规模网站的信息抽取	1. 无监督的方法，无需人工进行数据的标准； 2. 可以运用到大规模网站的信息抽取。
缺点	1. 需要对网页数据进行标注，耗费大量的人力； 2. 维护成本高； 3. 无法处理大量站点的情况。	1. 可维护性比较差； 2. 需要投入大量的人力去做标注。	1. 需要相似的网页作为输入； 2. 抽取的内容可能达不到预测，会抽取一些无关信息。

第6讲 页面抽取方法

6.1 数据抽取对象——网页的分类

1. 按照页面内数据组织形式的不同，分为

- 单记录页面
- 多记录页面

2. 按照页面承载内容的不同，分为

- 数据型页面
- 文档型页面

两者组合起来共有4 种页面类型

- 多记录数据型页面
- 单记录数据型页面
- 单记录文档型页面
- 多记录文档型页面

6.2 多记录数据型页面的抽取方法

1. 针对多记录的数据型网页

- 方法
  - 观察得到规则
  - 数据记录抽取： 确定数据区域（数据记录的定位），计算数据记录的边界，去除噪声数据记录
  - 数据项抽取： 数据项识别，数据项匹配

6.3 单记录数据型页抽取方法

增量式抽取

- 即**从多个连续页面中抽取同结构的记录**

6.4 单文档型页面Authority抽取方法

6.4.3 改进的自适应数据抽取方法

一种基于**贝叶斯最优决策**的方法
如果最优决策结果为vision，则采用基于视觉的数据抽取方法，并在数据库中相应地更新抽取规则
若最优决策结果为rule，则采用基于抽取规则的抽取方法。

6.5 多记录文档型Hub页面抽取方法

基于**序列匹配与树匹配**的混合抽取方法

第7讲 web数据存储与应用

7.1 爬虫数据存储

爬虫数据存储

- 结构化数据
  - 结构化文件
    - Excel
    - CSV文件
    - JSON 文件
    - Xml 文件
  - 数据库
- 非结构化数据（文本图像等）
  - 非结构化文件，raw data储存直接存诚txt文件
  - 后续可以加载到HDFS(Hadoop分布式文件系统)
  - 建立新的索引结构---倒排表

7.1.1 结构化文件

1. CSV(comma-separated values)
   CSV是以逗号间隔的文本文件。

- 被Excel和很多的应用程序支持。用来做数据存储容量小，很多数据集采用格式

2. JSON文件
   JSON 是轻量级的文本数据交换格式，是存储和交换文本信息的语法。类似 XML。 比 XML 更小、更快，更易解析。
   在JSON中，有两种结构：对象和数组
3. XML文件
   可扩展标记语言
   数据重用——用来存储，携带，交换数据的，不是用来显示数据的
   半结构化集成数据
4. EXCEL 文件
   二进制文件

7.1.2 数据库

- 数据库提供了更强有力的**数据存储和分析能力**，

第8讲 词项词典

8.1 文档解析(Parsing a document)

文档格式、语言（可以看成是机器学习中的**分类**问题，但在实际中往往采用启发式方法来实现）、编码方式

8.2 词条化 (Tokenization)

Web 数据中的文本->字符序列　分词
“词条”Token
词条化：将给定的字符序列拆分成一系列子序列的过程，其中每一个子序列称之为一个“词条”Token。

- 利用空格，标点符号进行分割

8.3 词项归一化 (Normalization)

- 归一化：将文档和查询中的词条“归一化”成一致的形式
- 归一化的结果： 在IR系统的词项词典中，形成多个近似词项的一个等价类
- 词项归一化的策略：建立同义词扩展表。
    - 手工建立同义词词表
        - 为每个查询维护一张包含多个词的查询扩展词表
        - 在建立索引建构时就对词进行扩展

8.4 词干还原 (Stemming)

通常指**去除单词两端词缀的启发式过程。**
词干还原能够提高召回率，但是会降低准确率
Porter算法（英文处理中最常用的**词干还原**算法）

8.5 词形归并 (Lemmatization)

利用词汇表和词形分析来减少屈折变化的形式，将其转变为基本形式。
词形归并可以减少词项词典中的词项数量

8.6 停用词 (Stop Words)

应用太广泛，对这样的词搜索引擎无法保证能够给出真正相关的搜索结果，难以帮助缩小搜索范围，同时还会降低搜索的效率，所以通常会把这些词从问题中移去，从而提高搜索性能。

- 消除方法：
    - 查表法
    - 基于文档频率

构造停用词表

- 语法剔除
- 利用词频
- 搜集网络上一些公开的停用词表

8.7 开源 NLP库/工具

NLTK

Gensim

第9讲 中文分词

9.1 分词

针对不同的语言，采取不同策略的词条化方法
中文分词(Chinese Word Segmentation)

- 指的是将一个汉字序列切分成一个一个单独的词。
- 分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。

中文分词技术属于自然语言处理（NLP)技术范畴
机器翻译（MT）、语音合成、自动分类、自动摘要、自动校对等等，都需要用到分词。

9.2 分词算法介绍

9.2.1 基于理解的分词方法

- NLP，语义分析，句法分析。
- 通过让计算机模拟人对句子的理解，达到识别词的效果

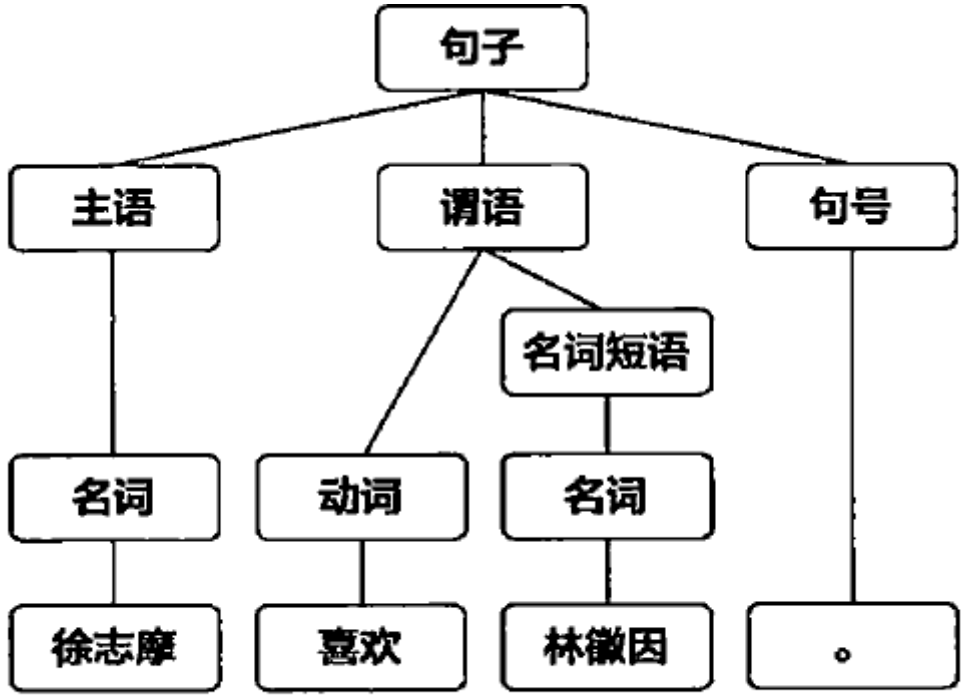


图 2.3　句子的语法分析树

9.2.2 基于字符串匹配的分词方法

基于词典(规则)的方法

- “查字典” 法
  - 按照一定策略将待分析的汉字串与一个“词典”中的词条进行匹配，如果匹配成功，那么该汉字串就是一个词。遇到不认识的字串就分割成单字词
- 策略
  - 按照**扫描方向**：正向匹配和逆向匹配
  - 按照**扫描长度**：最大匹配和最小匹配
  - 最少切分（使每一句中切出的词数最小）
- 正向最大匹配
- 逆向最大匹配

- 双向最大匹配

9.2.3 基于统计的分词方法。

- 假设：如果相连的字在不同的文本中出现的次数越多，就证明这相连的字很可能就是一个词。
- 主要统计模型
  - N元文法模型（N-gram）
    - 语言统计模型
  - 隐马尔可夫模型（Hidden Markov Model，HMM）
    - 最大熵马尔可夫模型（MEMM）

9.3 统计语言模型 (Statistical Language Models)

统计语言模型

n-gram语言模型
N-1阶马尔可夫假设:假定文本中的每个词ωi和前面的N-1个词有关，而与更前面的词无关
对应的语言模型称为N元模型(N-Gram Model)。

9.4 基于HMM的中文分词方法

9.4.1 HMM（定义、参数、应用环境、解决什么问题、算法）

隐马尔可夫模型（Hidden Markov Model，HMM），是比较经典的机器学习模型，在语言识别，自然语言处理，模式识别等领域得到广泛的应用。

HMM 统计模型，用来描述一个含有隐含未知参数的马尔可夫过程。

隐马尔可夫模型是关于时序的概率模型;

- 描述由一个**隐藏**的马尔可夫链随机生成**不可观测的状态随机序列**，再由各个状态生成一个观测而产生**观测随机序列**的过程,
- 序列的每一个位置又可以看作是一个时刻。

HMM模型是一个五元组:

- StatusSet: 状态值集合 Q
- ObservedSet: 观察值集合 V
- TransProbMatrix: 转移概率矩阵 A
- EmitProbMatrix: 发射概率矩阵 B
- InitStatus: 初始状态分布

HMM模型应用

- 股票预测
- 模式识别
- 语音识别
- 分词

  解决概率计算问题、学习问题、预测问题

  viterbi

9.4.2 HMM 分词

预测问题，也称为解码问题
ObservedSet 输入的句子
StatusSet 输出的分词结果

三个已知参数是经过训练获得:

参数 InitStatus 初始状态分布

– 句子的第一个字属于{B,E,M,S}这四种状态的概率

参数TransProbMatrix转移概率矩阵，

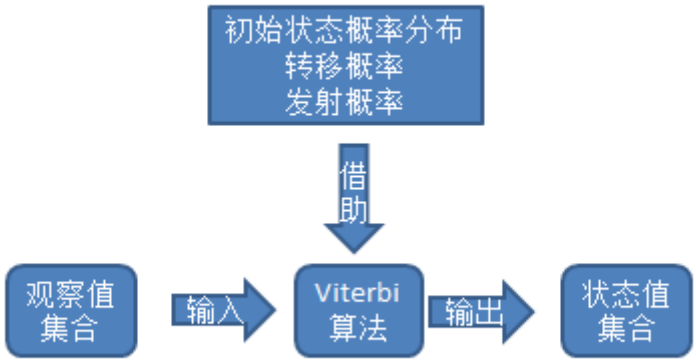
– 一个4x4(4就是状态值集合的大小)的二维矩阵

– **B****后面只可能接(M or E)，不可能接(B or S)。而****

M**后面也只可能接(M or E)，不可能接(B, S)****。

参数 发射概率(EmitProb):

– 在某一状态下对应到某字的概率



9.5 中文分词软件介绍

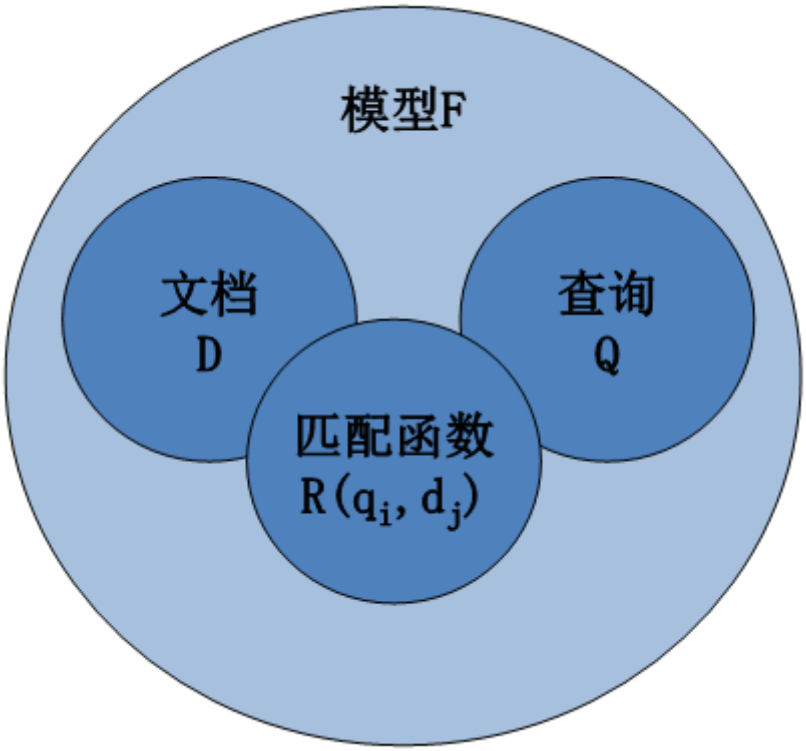
- python中文分词：结巴分词

第10讲 信息检索与倒排索引

10.2 信息检索模型（IR model）

- 信息检索模型（IR model），依照用户查询，对文档集合进行相关排序的一组前提假设和算法。IR模型可形式地表示为一个四元组< D, Q, F, R(qi,dj) >
- D是一个文档集合
- Q是一个查询集合，用户任务的表达，由查询需求的逻辑视图来表示。
- R(qi,dj) 是一个排序函数，它给查询qi和文档 dj 之间的相关度赋予一个排序值
- F是一个框架,用以构建文档,查询以及它们之间关系的模型

描述信息检索中的文档、查询和它们之间关系(匹配函数)的数学模型。



基于内容的信息检索模型：

- 集合论模型：**布尔模型**、模糊集合模型、扩展布尔模型
- 代数模型： **向量空间模型**、广义向量空间模型、潜在语义标引模型、神经网络模型
- 概率模型： **经典概率论模型**、推理网络模型、置信（信念）网络模型
- 深度学习模型

10.3 布尔检索模型

- 一种简单的检索模型。建立在经典的集合论和布尔代数的基础上
- 布尔代数
  - 布尔变量
  - 只有“真”、“假”取值的变量
- 遵循两条基本规则:
  - **每个索引词在一篇文档中只有两种状态**：出现或不出现，对应权值为 0或1。
  - 每篇文档：索引词（0或1）的集合

bag of words 模型

- “词袋”模型
- 在信息检索中，Bag of words model假定
- 对于一个文本，忽略其词序和语法，句法，将其仅仅看做是一个词集合，或者说是词的一个组合，
- 文本中每个词的出现都是独立的，不依赖于其他词是否出现，在任意一个位置选择一个词汇都不受前面句子的影响而独立选择的。

布尔代数

- 布尔表达式
  - 多个布尔变量之间通过布尔操作组成的表达式
  - 蕴含：两个布尔表达式P、Q，如果P为true，那么Q为true，则称P蕴含Q，记为P->Q
- 布尔操作(关系)

- 与(AND)：(A AND B) = true iff A=true and B=true
- 或(OR)：(A OR B) = true iff A=true OR B=true
- 非(NOT)：(NOT A) = true iff A=false

文档表示

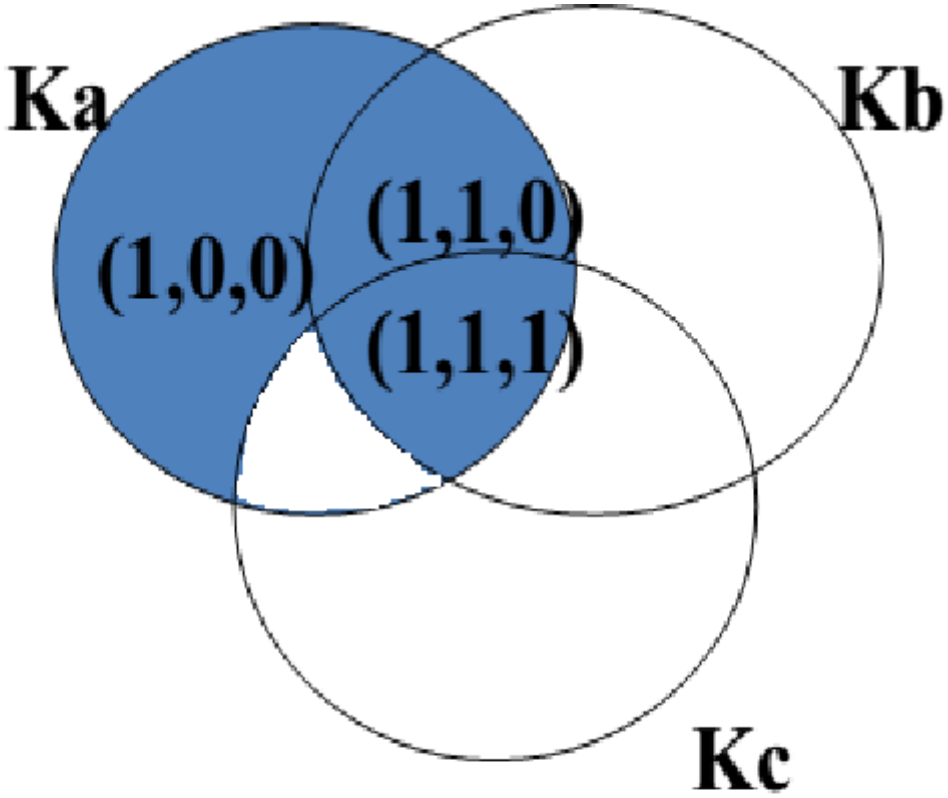
- 一个文档被表示为关键词(bag of words)的集合

查询表示

- 查询式(Queries)被表示为关键词的布尔组合，用“与、或、非”连接起来

相关度计算

- 一个文档当且仅当它能够满足布尔查询式时，才将其检索出来
- 检索策略是二值匹配{0,1}



$$q = k_a \wedge (k_b \vee \neg k_c)$$

形式化表示

- sim(dj, q) 为该模型的匹配函数
  - 如果 sim(dj,q)=1，则表示文献dj与q相关，否则为不相关。

10.4 倒排索引

搜索引擎的核心数据结构为倒排文件（Inverted Files），也称倒排索引(Inverted index)

- 词项 + 倒排记录
- 词项词典：对于每一个词项，存储所有包含这个词项的文档的一个列表。
- 倒排记录表：一个文档用一个序列号docID来表示。
- 应当使用可变长度的记录列表

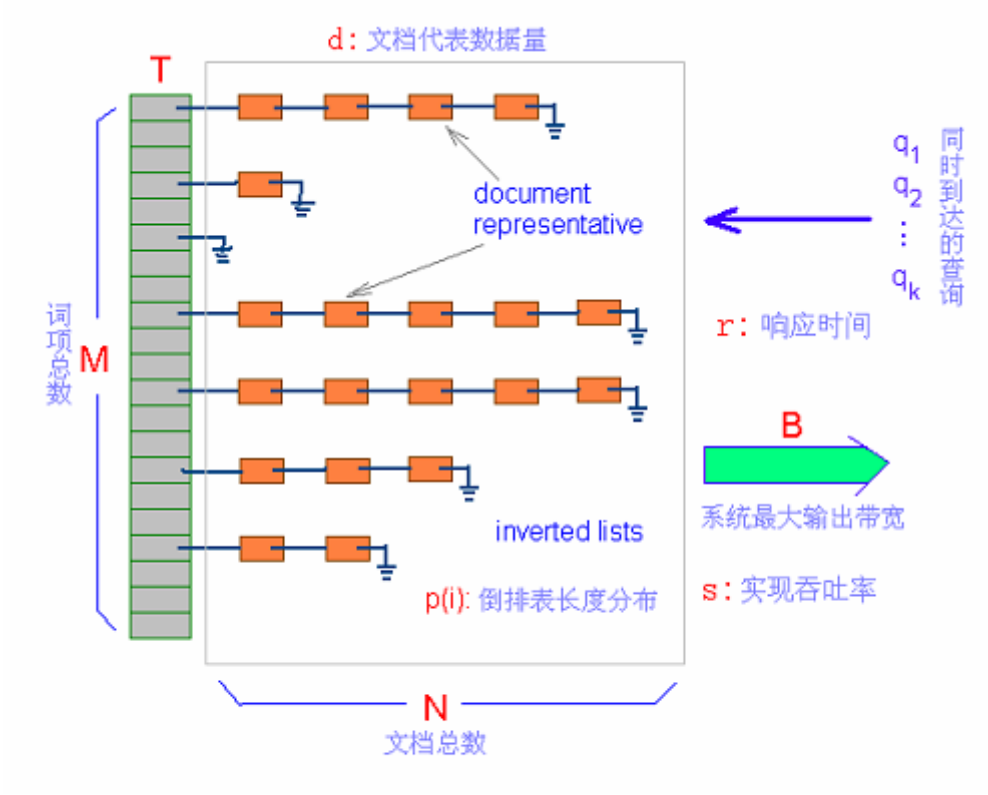


图 8-3 倒排文件结构示意图

10.4.1 倒排索引的操作

查询处理

- 单个词查询
- 多词查询
    - 逻辑与 and
    - 逻辑或 or

查询的处理：AND

10.5 开源搜索引擎

Lucene
Solr
ElasticSearch
Nutch

第11讲 向量空间模型

11.1 布尔检索模型的特点

- 优点
    - 查询简单，因此容易理解

    - 通过使用复杂的布尔表达式，可方便地控制查询结果
- 缺点
    - 准确匹配，信息需求的能力表达不足。不能输出部分匹配的情况
    - 无权重设计 无法排序，
    - 用户必须会用布尔表达式提问，一般而言，检出的文档或者太多或者太少。
    - 很难进行自动的相关反馈

排序检索

- 在排序检索模型中，系统根据**文档与query的相关性排序**返回文档集合中的文档，而不是简单地返回所有满足query描述的文档集合
- 希望根据文档对查询者的有用性大小顺序排序
- **如何根据一个query对文档进行排序?**
    - 给每个"查询-文档对"进行评分，在[0,1]之间
    - 这个评分值衡量**文档与query的匹配程度**

排序检索的基本——评分

- 布尔检索模型
    - 单个词组成的query
        - 如果该词项不出现在文档中，该文档评分为0
        - 该词项在文档中出现，则评分为1

排序检索模型中有布尔查询和自由文本查询两种方式

- 布尔查询
- 自由文本查询：用户query是自然语言的一个或多个词语而不是由查询语言构造的表达式

11.2 文档之间的相似度计算

11.2.1 Character Based Methods

- 以字符为单位比较的方法被统称为 Character Based
- 普遍用来进行较短文本、小规模的比较，
- 会注意文本中各字符的顺序和位置；

1. 最长公共子序列
2. 编辑距离(Leveinshtein莱文斯坦 Distance)
3. 扩展的编辑距离
4. Needleman-Wunsch Similarity
5. Smith-Waterman Similarity
6. Jaro Similarity 和 Jaro-Winkler Similarity
7. Hamming Distance

11.2.2 Term Based Methods

- 以词为单位的则被称为 Term Based。
- 更适合用来进行较长文本或大规模的比较

Jaccard相似度：比较文本相似度，用于文本查重与去重；
Jaccard距离：计算对象间距离，用于数据聚类等。

$$J_\delta(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

用Jaccard 系数评分的问题

没有考虑文档长短；词项频率(词项在文档中出现的次数)；罕见词比高频词的信息量更大，更加具有区分度

11.3 词项频率

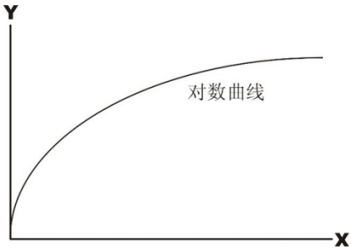
词项频率：词项t在文档d中出现的次数，记为tft,d
如何利用tf计算query-document评分？

- 采用原始tf值(raw tf)，不太合适：
  - 某个词项在A文档中出现十次，即tf = 10，在B文档中tf = 1，那么A比B更相关
  - 但是相关度不会相差10倍
  - 相关性不会正比于词项频率
- 对数词频

一种替代原始tf的方法: 对数词频

- 词项t在文档d中的对数频率权重

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d} & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$



- $tf_{t,d} \rightarrow w_{t,d}$: 0→0, 1→1, 2→1.3, 10→2, 1000 → 4
- 文档-词项的匹配得分是所有同时出现在q和文档d中的词项的词频的对数之和

$$Score(q,d) = \sum_{t \in q \cap d} (1 + \log tf_{t,d})$$

- 评分为0，表示文档和query中没有公共词项

11.4 tf-idf权重计算

除词项频率tf之外，我们还想利用词项在整个文档集中的频率进行权重和评分计算

- **罕见词项比常见词所蕴含的信息更多**
  - 对**罕见词项**赋予**高权重**
  - 对**常见词项**赋予**低权重**
  - 对**停用词**赋予**零权重**
- 对只有一个查询词的query，idf对排序结果没有影响
- 对于含有**两个以上查询词的query**，idf才会影响排序结果

文档频率 (Document frequency，df)

- **文档频率**：出现词项的**文档数目**
- dft 文档集合中包含t的文档数目
  - 与词项t包含的**信息量**成**反比**
  - dft <= N (N是文档的总数)
- idf (inverse document frequency)逆文档频

- idft = log10(N/dft)
  - idft 是反映**词项t的信息量**的一个指标
  - 用log (N/dft) 代替N/dft 来抑制idf的作用

tf-idf 是信息检索中最著名的权重计算方法

- 词项t的tf-idf 由它的tf和idf组合而成 **wt,d=(1+log tft,d) × log10(N/dft)**
- tf-idf值随着词项在单个文档中出现次数(tf)**增加而增大**
- tf-idf值随着词项在文档集中数目(df)**增加而减小**

TF-IDF是一种**统计方法**，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在**语料库**中出现的频率成反比下降

11.5 向量空间模型

词项-文档二值关联矩阵

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	1	1	0	0	0	1
BRUTUS	1	1	0	1	0	0
CAESAR	1	1	0	1	1	1
CALPURNIA	0	1	0	0	0	0
CLEOPATRA	1	0	0	0	0	0
MERCY	1	0	1	1	1	1
WORSER	1	0	1	1	1	0
. . .						

- 每个文档用一个二值向量表示 $\in \{0,1\}^{|V|}$

词项-文档词频关联矩阵

- 考虑词项在文档中出现的频率
  - 将每个文档看成是一个词频向量：矩阵中的一列

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	157	73	0	0	0	1
BRUTUS	4	157	0	2	0	0
CAESAR	232	227	0	2	1	0
CALPURNIA	0	10	0	0	0	0
CLEOPATRA	57	0	0	0	0	0
MERCY	2	0	3	8	5	8
WORSER	2	0	1	1	1	5
. . .						

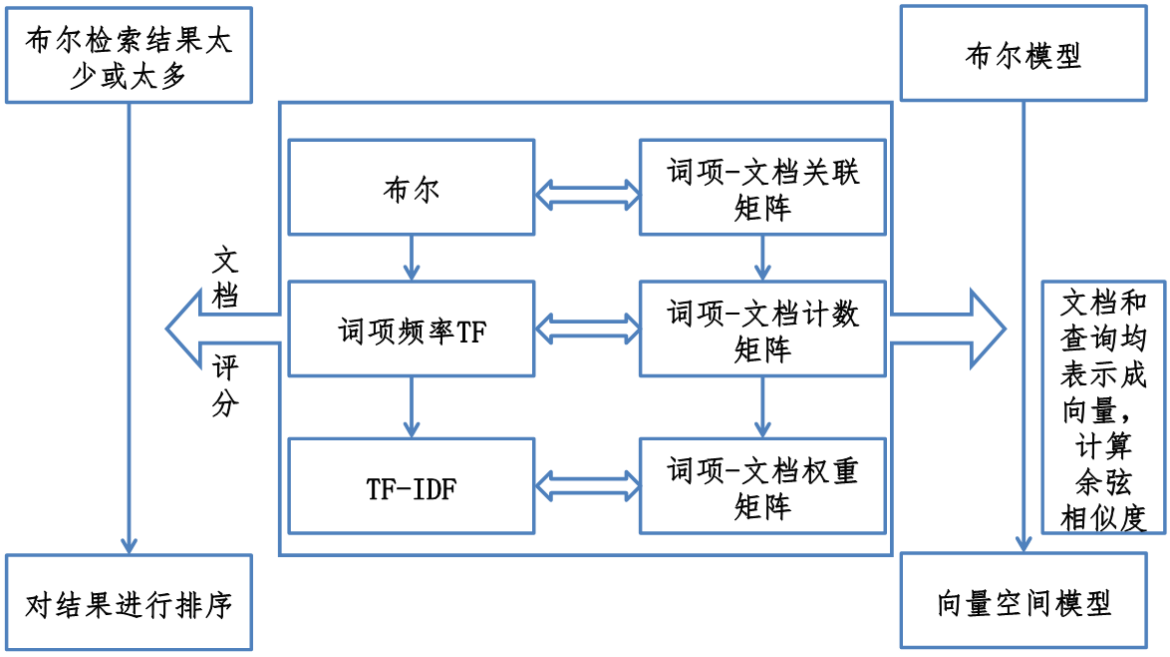
二值 → 词频 → tf-idf矩阵

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	5.25	3.18	0.0	0.0	0.0	0.35
BRUTUS	1.21	6.10	0.0	1.0	0.0	0.0
CAESAR	8.59	2.54	0.0	1.51	0.25	0.0
CALPURNIA	0.0	1.54	0.0	0.0	0.0	0.0
CLEOPATRA	2.85	0.0	0.0	0.0	0.0	0.0
MERCY	1.51	0.0	1.90	0.12	5.25	0.88
WORSER	1.37	0.0	0.11	4.15	0.25	1.95
. . .						

每篇文档表示成一个基于tf-idf权重的实值向量 $\in R^{|V|}$

向量空间模型

- **|V|维实向量空间**
  - **V是词项集合**，|V|表示词项个数
  - 空间的每一**维**都对应一个**词项**
- 每篇文档表示成一个基于**tf-idf权重的实值向量**$\in R^{|V|}$
  - 文档是空间中的点或者向量



Vector Space Model, VSM

- **文本内容的处理简化为向量空间中的向量**
- **以空间上的相似度表达语义的相似度**
  相比于布尔模型要求的准确匹配, VSM模型采用了"部分匹配"的检索策略（即：出现部分索引词也可以出现在检索结果中）。通过给查询或文档中的索引词分配非二值权值来实现。 有序

向量空间模型特点

- 优点:
  - 帮助改善了检索结果。

    - 部分匹配的文档也可以被检索到。
    - 可以基于向量cosine 的值进行排序，提供给用户。
- 缺点:
    - 这种方法假设标记词是相互独立的，但实际可能不是这样，如同义词、近义词等往往被认为是不相关的词
    - 维度非常**高**：特别是互联网搜索引擎，空间可能达到千万维或更高
    - 向量空间非常**稀疏**：对每个向量来说大部分都是0

11.6 向量空间模型的价值

VSM模型的价值：将无结构化文本表示为向量，各种数学处理成为可能

jieba分词系统中的TF-IDF抽取关键词：预处理，进行分词和词性标注，将满足指定词性的词作为候选词；分别计算每个词的TF-IDF值根据每个词的TF-IDF值降序排列，并输出指定个数的词汇作为可能的关键词

使用gensim tf-idf模型求文本相似度

第12讲 概率检索模型

12.3 BM25

BM25是信息索引领域用来计算query与文档相似度得分的经典算法。
不同于TF-IDF，BM25的公式主要由三个部分组成：

- query中每个单词t与文档d之间的相关性
- 单词t与query之间的相似性
- 每个单词的权重

传统的TF值理论上是可以无限大的。而BM25与之不同，它在TF计算方法中增加了一个常量k，用来限制TF值的增长极限
BM25的TF Score会被限制在0~k+1之间

- 如果查询很长，那么对于查询词项也可以采用类似的权重计算方法。

$$\frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}$$

- tf_{tq}是词项t在查询q中的权重。
- k3 是另一个取正值的调优参数，用于对查询中的词项tq 频率进行缩放控制。

单词权重 idf　　单词和文档相关性　　单词和query相关性

$$RSV_d=\sum_{t\in q}\log\left[\frac{N}{df_t}\right]\cdot\frac{(k_1+1)tf_{td}}{k_1[(1-b)+b\times(L_d/L_{ave})]+tf_{td}}\cdot\frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}$$

第14讲 检索排序

14.2 链接分析

图数据

- 将整个静态Web看成是静态HTML网页通过超链接互相连接而成的有向图，其中每个网页是图的顶点，而每个超链接则代表一个有向边。

- 顶点和有向边集合称为Web图
- 链接分析是指源于对Web结构中超链接的多维分析

链接分析算法

- Pagerank 算法
  - 强调链接数量与质量整体关系
- HITS算法
  - 强调权威页与枢纽页的 相互增强关系
  - （相关性）

14.3 PageRank算法

- 计算互联网网页重要度的算法
  - 对每个网页给出一个正实数，表示网页的重要程度， 整体构成一个向量
  - PageRank值越高，网页就越重要，在互联网搜索的排序中可能就被排在前面
- PageRank算法是图的链接分析（link analysis）的代表性算法，属于图数据上的无监督学习方法。
  - PageRank可以定义在任意有向图
    - 图数据: 互联网、社交网络、交通网络
  - 应用：社会影响力分析、文本摘要、交通流量预测等
    - Textrank
    - simrank

14.3.1 PageRank算法的基本想法

- 随机游走（Random Walk，缩写为 RW
- 数学统计模型： 它是一连串的轨迹所组成，其中每一次都是随机的。
- 能用来表示不规则的变动形式，
- 随机游走的形式有：
  - 马尔可夫链或马可夫过程
  - 醉汉走路（drunkard’s walk）

网页浏览行为

- 随机游走模型，
- 一阶马尔可夫链
  - 从一个随机的页面开始
  - 每一步从当前页等概率地选择一个链接，进入链接所在页面
- 在稳定状态下，每个页面都有一个访问概率 – 用这个概率作为页面的分数
  - 当冲浪者在 Web 上进行节点间的随机游走时，某些节点的访问次数会比其它节点更多。
- 直观地看，这些访问频繁的节点具有很多从其它频繁访问节点中指向的入链接。

直观上， PageRank值高，这个网页重要
指向该网页的超链接越多，随机跳转到该网页的概率也就越高
指向该网页的PageRank值越高，指向该网页的PageRank值被分走的越少

PageRank值依赖于网络的拓扑结构，一旦网络的拓扑（连接关系）确定，PageRank值就确定

在一定条件下，极限情况访问每个结点的概率收敛到平稳分布，这时各个结点的平稳概率值就是其PageRank值，表示结点的重要度。
PageRank表示这个马尔可夫链的平稳分布，每个网页的PageRank值就是平稳概率。
PageRank是递归定义的，PageRank的计算可以 通过迭代算法进行。

14.3.2 PageRank的一般定义

- PageRank一般定义的想法是在基本定义的基础上导入平滑项，给定一个含有n个结点vi, i=1,2,...,n，的任意有向图
- 假设考虑一个在图上随机游走模型，即一阶马尔可夫链，其转移矩阵是M，从一个结点到其连出的所有结点的转移概率相等。这个马尔可夫链未必具有平稳分布
- 假设考虑另一个完全随机游走的模型，其转移矩阵的元素全部为1/n，也就是说从任意一个结点到任意一个结点的转移概率都是1/n
- 两个转移矩阵的线性组合又构成一个新的转移矩阵，在其上可以定义一个新的马尔可夫链。
- 容易证明这个马尔可夫链一定具有平稳分布，且平稳分布满足

$$R = dMR + \frac{1-d}{n}\mathbf{1} \qquad (21.10)$$

- R 是n维向量
  - R表示的就是有向图的一般PageRank

    $PR(v_i),\ i = 1, 2, \cdots, n$ 表示结点v_i的PageRank值

$$R = \begin{bmatrix} PR(v_1) \\ PR(v_2) \\ \vdots \\ PR(v_n) \end{bmatrix}$$

- 1是所有分量为1的n维向量
- d称为阻尼因子（damping factor）
  - 系数
  - 0≤d ≤ 1

第一项表示（状态分布是平稳分布时）依照转移矩阵M访问各个结点的概率，
第二项表示完全随机访问各个结点的概率，第二项称为平滑项，由于采用平滑项，所有结点的PageRank值都不会为0
阻尼因子d取值由经验决定

PageRank 算法实现

- 迭代算法
- 幂法
- 代数法

14.4 其他排序算法

14.4.1 HITS算法

- Hyperlink- Induced Topic Search (HITS)超链导向的主题搜索
- HITS算法专注于改善泛指主题检索的结果
  - 比如一个以程序开发为主题网页，指向另一个以程序开发为主题的网页，则另一个网页的重要性就可能比较高，但是指向另一个购物类的网页则不一定。

网页分类

- 权威Authoritive页 A
  - 一个网页被多次引用，则它可能是很重要的；
  - 一个网页的重要性被平均的传递到它所引用的网页。
    - 一个网页虽然没有被多次引用，但是被重要的网页引用，则它也可能是很重要的；
- 列表Hub页H

    - 它本身可能并不重要，或者说没有几个网页指向它，但是它提供了指向就某个主题而言最为重要的站点的链接集合，比如一个课程主页上的推荐参考文献列表

PageRank算法和HITS算法比较

- 都是基于链接分析的搜索引擎排序算法，并且在算法中两者都利用了特征向量作为理论基础和收敛性依据。
- HITS算法计算的authority值只是相对于某个检索主题的权重，因此HITS算法也常被称为Query-dependent算法；
- 而PageRank算法是独立于检索主题，因此也常被称为Query-independent算法。

第15讲 图片检索-颜色特征

15.1 图像检索

TBIR基于文本的图像检索

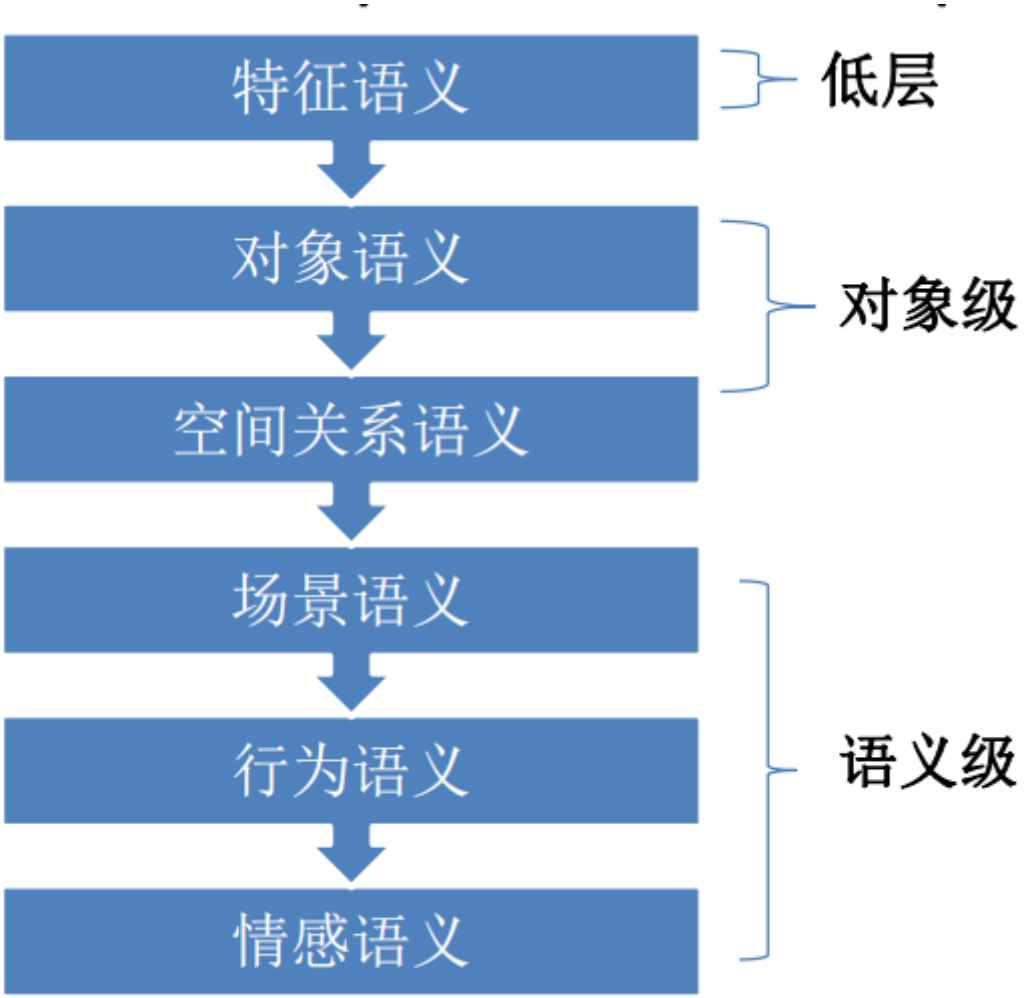
- 查询词：文本
- 搜索引擎
    - 爬虫　图片
    - 索引　图片对应的文字，锚文本，URL
    - 基于图像周围文本的检索
    - 基于链接锚文本的检索

基于内容的图像检索CBIR（Content-based image retrieval）

- 用户输入一张图片，以查找具有相同或相似内容的其他图片。
- **CBIR 的关键技术:图像特征提取和匹配**

15.2 图像特征

图像的特征主要包括低层特征和语义特征



15.4 颜色空间

RGB

HSV(色调、饱和度、亮度)

颜色特征

Color Features
颜色是彩色图像最底层、最直观的物理特征，通常对噪声，图像质量的退化，尺寸、分辨率和方向等的变化具有很强的鲁棒性，是绝大多数基于内容的图像和视频检索的多媒体数据库中使用的特征之一。

15.4.1 颜色直方图（ColorHistogram）

- 最简单也是最常用的颜色特征
- 核心思想
  - **统计学，在颜色空间中采用一定的量化方法对颜色进行量化**，然后**统计每一个量化通道在整幅图像中所占的比重。**
  - 每种颜色Ci , Hci(I) 表示图片I 中，颜色是Ci的**像素数目**
- **描述的是不同色彩在整幅图像中所占的比例，统计分布特性**
- 具有平移、尺度、旋转不变性，特别适于描述那些难以进行自动分割的图像。
- 没有空间信息

15.4.2 颜色相关图(ColorCorrelogram)

- 用**颜色对相对于距离的分布**来描述信息，
- 它反映了像素对的空间相关性，以及局部像素分布和总体像素分布的相关性。
    - 只考虑共同颜色之间的空间关系->颜色自相关图。
- 既有颜色信息，又有空间信息。

15.4.3 颜色矩(ColorMoment)

- 其基本思想
    - 在**颜色直方图的基础上计算出每个颜色的矩估计**
        - 颜色信息主要分布于低阶矩中
        - 一阶矩(均值,mean),二阶矩(方差,viarance),三阶矩(斜度,skewness),四阶，峭度
    - 用**这些统计量替代颜色的分布来表示颜色特征**
    - 它具有特征量少，处理简单的特点。
- 该方法的优点：
    - 不需要颜色空间量化，特征向量维数低；
    - 但实验发现该方法的检索效率比较低，因而在实际应用中往往用来过滤图像以缩小检索范围，常和其他特征结合

15.4.4 MPEG-7 主颜色描述子

描述了图像中最常出现、处于支配地位的颜色信息。

主导颜色所占比重、其颜色方差、主导颜色的空间一致性。