

上讲回顾：数据采集

- 文本分类方法
 - 网页抓取
 - 网页解析
 - Scrapy实践（选学）



中國人民大學
RENMIN UNIVERSITY OF CHINA

计算传播理论与实务

2019-2020秋季学期

第三讲

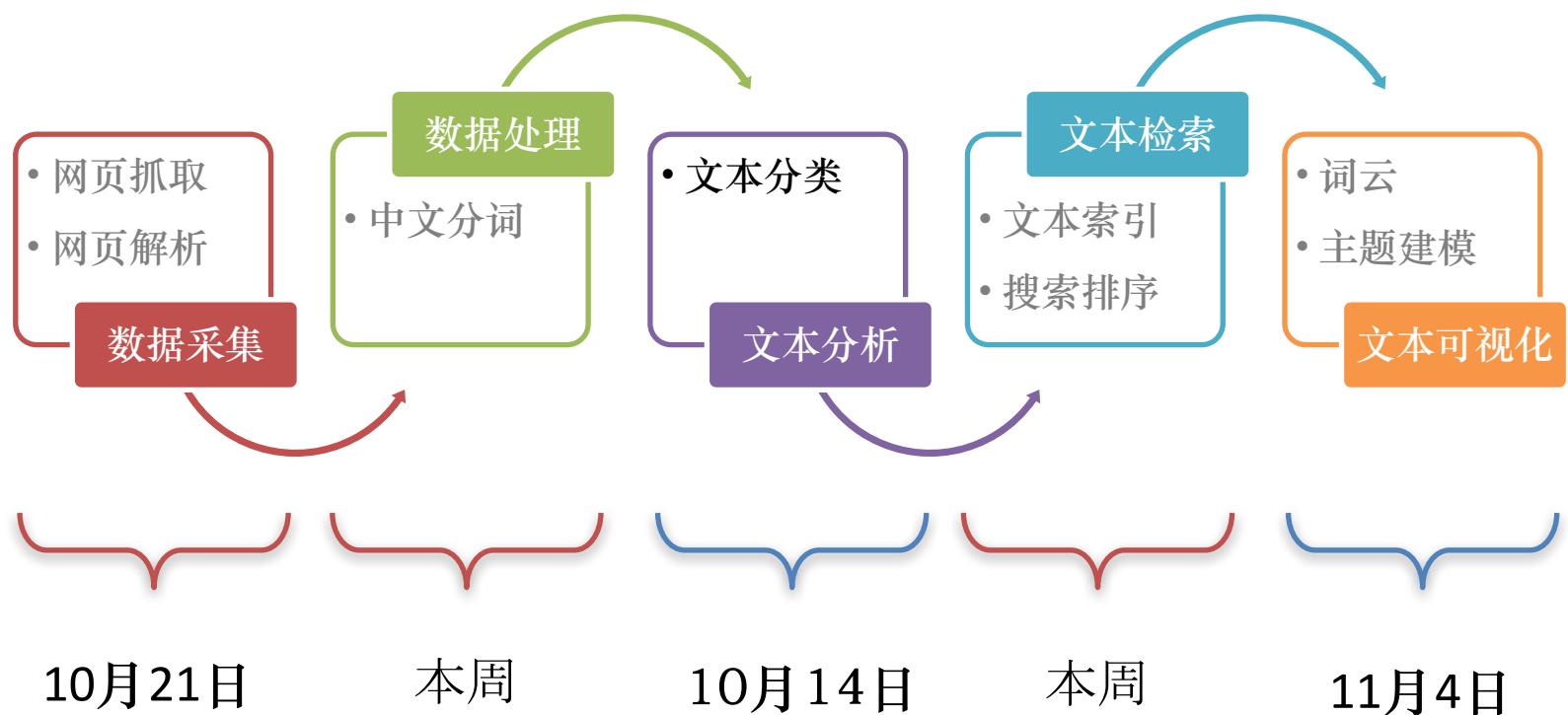
文本分析

授课教师：范举副教授、塔娜讲师

时间：2019年10月28日

文本模块主要内容

- 目标：自动分析出人们在“说”什么
 - 大多新闻与传播领域的数据以文本形态存在



第3.3节

文本分词 与检索

3.3.1 文本分词

- ✓ 问题背景
- ✓ 分词方法
- ✓ 分词工具

第3.3.1节 文本分词

问题背景

中文文本特点

- 英文（以及一些国家/地区语言文字）词与词之间有空格（分隔符），分词处理相对容易
例如：This is a book.
- 中文基于单字，中文书面表达方式以汉字作为最小单位的
 - 字与字之间、词与词之间紧密连接，词与词之间没有显性的界限标志
 - 词是最小并且能独立活动的语言成分，文章以词为基本单位来形成有意义的篇章
 - 添加合适的显性的词语边界标志使得所形成的词串反映句子的本意
 - 所以分词是汉语文本分析处理中首先要解决的问题

文本分析任务之词法分析

- 词法分析是将构成句子的字符序列转换为词的序列，并对每个词加上语法或语义标记
 - 分词：对句子进行分词，完成该功能的软件称为分词器(Tokenizer)。
 - 词性标注：Part-of-Speech Tagger，(简称POS Tagger) 分析某种语言的文本，然后针对每个词(Word或者Token)赋予POS标记，比如名词(Noun)、动词(Verb)、形容词(Adjective)等
 - 词义消歧

分词的意义

- 正确的机器自动分词是正确的中文信息处理的基础

- 文本检索

- 和服 | 务 | 于三日后裁制完毕，并呈送将军府中。
 - 王府饭店的设施 | 和 | 服务 | 是一流的。

如果不分词或者“和服务”分词有误，都会导致荒谬的检索结果。

- 文语转换

- 他们是来 | 查 | 金泰 | 撞人那件事的。（“查”读音为cha）
 - 行侠仗义的 | 查金泰 | 远近闻名。（“查”读音为zha）

分词面临的主要难题

- 如何面向大规模开放应用是汉语分词研究亟待解决的挑战
 - 如何识别未登录词
 - 如何利用语言学知识
 - 词语边界歧义处理
 - 实时性应用中的效率问题

南京市长江大桥

还好我一把把把住了

我也想过过儿过过的活

校长说衣服上除了校徽别别的

未登录词

- 虽然一般的词典都能覆盖大多数的词语，但有相当一部分的词语不可能穷尽地收入系统词典中，这些词语称为未登录词
- 分类：
 - 专有名词：中文人名、地名、机构名称、外国译名、时间词
 - 重叠词：“高高兴兴”、“研究研究”
 - 派生词：“一次性用品”
 - 与领域相关的术语：“互联网”

分词歧义

- 交集型切分歧义

- 汉字串AJB被称作交集型切分歧义，如果满足A、J、B同时为词(A、J、B分别为汉字串)。此时汉字串J被称作交集串。

- [例]“结合成分子”

- 结合 | 成分 | 子 |
 - 结合 | 成 | 分子 |
 - 结 | 合成 | 分子 |

- [例]“美国会通过对台售武法案”

- [例]“乒乓球拍卖完了”

- 组合型切分歧义

- 汉字串AB被称作组合型切分歧义，如果满足条件：A、B、AB同时为词

- [例]组合型切分歧义：“起身”

- 他站 | 起 | 身 | 来。

- 他明天 | 起身 | 去北京。

第3.3.1节 文本分词

分词方法

分词方法

- 优点：简单、可理解、结果可控
- 缺点：规则维护困难，分词精度欠佳

基于人工规则的分词

基于统计机器学习的分词

- 优点：数据驱动，应用广泛
- 缺点：高质量训练数据获取昂贵，特征工程

- 优点：精度高
- 缺点：可解释性差，需要海量训练数据

基于深度学习的分词

基于统计机器学习的方法

- 机器学习：研究一类算法，使之
 - 在某些任务上(task)
 - 通过已有的观测经验(数据)(experience)
 - 提升算法效果(performance)
- 两个过程
 - 离线训练：基于标注数据，发现规则（确定模型参数）
 - 在线预测：基于已发现的规则，对新数据进行预测（如：标注）
- 人工规则 → 从标注数据中自动发现规则

基于统计机器学习的分词流程

训练样本

市场/中/国有/企业/才能/发展

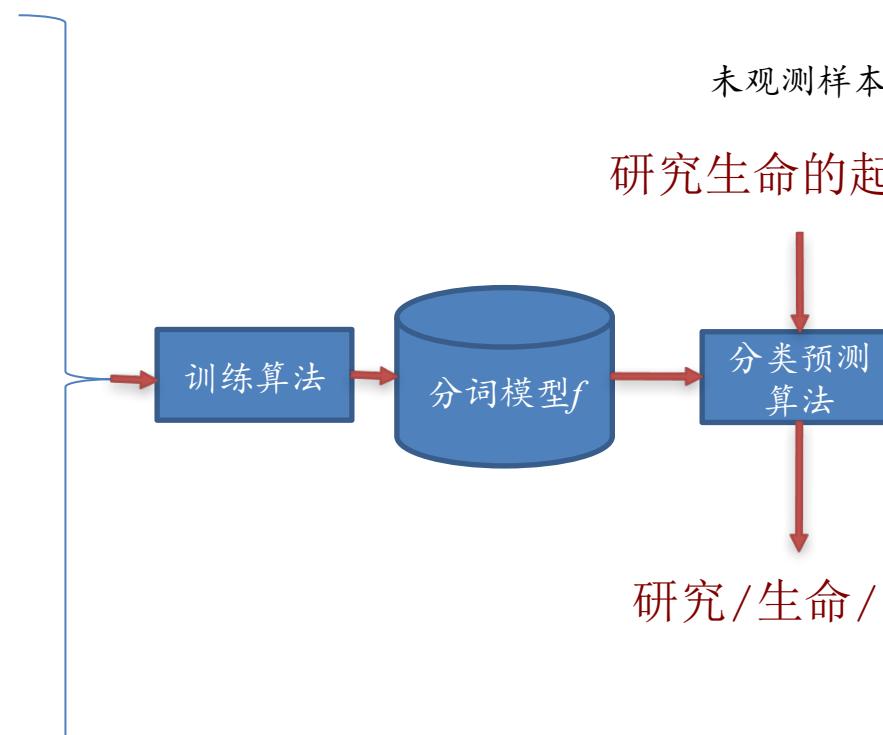
吃/两/顿/饭

跳/新疆/舞

... ...

时间/就/是/生命/

失败/是/成功/之/母



第3.3.1节 文本分词

分词工具

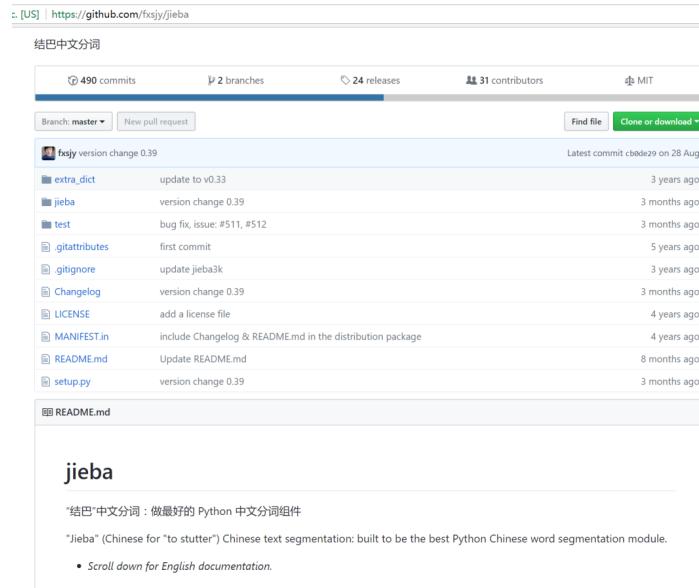
中文分词工具

资料来源：<https://github.com/crownpku/Awesome-Chinese-NLP>

- [Jieba 结巴中文分词](#) (Python及大量其它编程语言衍生) 做最好的 Python 中文分词组件
- [北大中文分词工具](#) (Python) 高准确度中文分词工具，简单易用，跟现有开源工具相比大幅提高了分词的准确率。
- [kcws 深度学习中文分词](#) (Python) BiLSTM+CRF与IDCNN+CRF
- [ID-CNN-CWS](#) (Python) Iterated Dilated Convolutions for Chinese Word Segmentation
- [Genius 中文分词](#) (Python) Genius是一个开源的python中文分词组件，采用 CRF(Conditional Random Field)条件随机场算法。
- [loso 中文分词](#) (Python)
- [yaha "哑哈"中文分词](#) (Python)
- [ChineseWordSegmentation](#) (Python) Chinese word segmentation algorithm without corpus (无需语料库的中文分词)

结巴分词

- <https://github.com/fxsjy/jieba>
- 广为流传的Python中文处理工具
- 最快捷的中文分词工具
- 安装： pip install jieba



结巴分词

- 支持三种分词模式：
 - 精确模式，试图将句子最精确地切开，适合文本分析；
 - 全模式，把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义；
 - 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。
- 支持繁体分词
- 支持自定义词典
- MIT 授权协议：免费使用和修改

Python函数调用

- 直接使用无需准备训练数据
- 分词函数：`jieba.cut`
 - 参数1：需要分词的字符串；
 - 参数2：`cut_all` 参数用来控制是否采用全模式
 - 参数3：`HMM` 参数用来控制是否使用 `HMM` 模型
- `jieba.cut_for_search`
 - 参数1：需要分词的字符串；
 - 参数3：`HMM` 参数用来控制是否使用 `HMM` 模型
- `jieba.lcut` 以及 `jieba.lcut_for_search` 直接返回 `list`

普通分词

第一次运行：

```
# encoding=utf-8
import jieba

seg_list = jieba.cut("我来到中国人民大学", cut_all=False)
print("Full Mode: " + "/ ".join(seg_list)) # 精确模式

Building prefix dict from the default dictionary ...
Dumping model to file cache /var/folders/67/81q9qf7d7cv3vkydc03wfjv80000gn/T/jieba.cache
Loading model cost 0.780 seconds.
Prefix dict has been built successfully.

Full Mode: 我/ 来到/ 中国人民大学
```

返回generator：

```
# encoding=utf-8
import jieba

seg_list = jieba.cut("我来到中国人民大学", cut_all=False)
print(seg_list)
for w in seg_list:
    print(w)

<generator object Tokenizer.cut at 0x10eeb5660>
我
来到
中国人民大学
```

全模式分词

- 全模式：把句子中所有的可以成词的词语都扫描出来

```
# encoding=utf-8
import jieba

seg_list = jieba.cut("我来到中国人民大学", cut_all=True)
print("精确模式: " + "/ ".join(seg_list))
```

精确模式： 我/ 来到/ 中国/ 中国人民大学/ 国人/ 人民/ 人民大学/ 大学

搜索引擎模式分词

- 适合用于搜索引擎构建倒排索引的分词，粒度比较细

```
# encoding=utf-8
import jieba

seg_list = jieba.cut_for_search("我来到中国人民大学")
print(seg_list)
for w in seg_list:
    print(w)
```

```
<generator object Tokenizer.cut_for_search at 0x10ec26408>
我
来到
中国
国人
人民
大学
中国人民大学
```

lcut和lcut_for_search

- jieba.lcut 以及 jieba.lcut_for_search 直接返回 list

```
# encoding=utf-8
import jieba

seg_list = jieba.lcut("我来到中国人民大学", cut_all=False)
print(seg_list)
```

['我', '来到', '中国人民大学']

直接使用已有模型的局限

- 分词模型快速，但是不可避免会出现错误

```
# encoding=utf-8
import jieba

seg_list = jieba.cut("中区食堂和北区食堂都是我喜欢的中国人民大学的食堂", cut_all=False)
print("/".join(seg_list))
```

中/区/食堂/和/北区/食堂/都/是/我/喜欢/的/中国人民大学/的/食堂

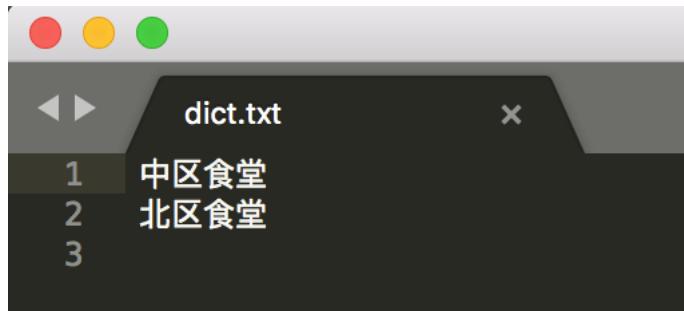
- 未登录词：
 - 中区食堂
 - 北区食堂

添加自定义词典

- 开发者可以指定自己自定义的词典，以便包含 jieba 词库里没有的词
 - 虽然 jieba 有新词识别能力，但是自行添加新词可以保证更高的正确率
- 用法： `jieba.load_userdict(file_name)` # `file_name` 为文件类对象或自定义词典的路径
 - 词典格式和 `dict.txt` 一样，一个词占一行；每一行分三部分：词语、词频（可省略）、词性（可省略），用空格隔开，顺序不可颠倒。
 - `file_name` 若为路径或二进制方式打开的文件，则文件必须为 UTF-8 编码。
- 词频省略时使用自动计算的能保证分出该词的词频。

词典示例

- 增加dict.txt（必须为存为UTF-8格式）



```
: import jieba  
jieba.load_userdict("/Users/junxu/Documents/Work/上课/数据科学导论课程/程序/jieba分词/dict.txt")  
seg_list = jieba.cut("中区食堂和北区食堂都是我喜欢的中国农业大学的食堂。", cut_all=False)  
print("/".join(seg_list))
```

中区食堂/和/北区食堂/都/是/我/喜欢/的/中国农业大学/的/食堂/。

动态增加单词

- 用户词典文件适合提前批量增加
- 在线动态增加单词：jieba.add_word

```
import jieba
#jieba.load_userdict("/Users/junxu/Documents/Work/上课/数据科学导论课程/程序/jieba分词/dict.txt")
seg_list = jieba.cut("中国人民大学数据科学导论本科生课程。", cut_all=False)
print("/".join(seg_list))
jieba.add_word("数据科学导论")
seg_list = jieba.cut("中国人民大学数据科学导论本科生课程。", cut_all=False)
print("/".join(seg_list))
```

中国人民大学/数据/科学/导论/本科生/课程/。

中国人民大学/数据科学导论/本科生/课程/。

给文件分词

```
1 # encoding=utf-8
2 import jieba
3 fnInput = "/Users/junxu/Documents/Work/上课/数据科学导论课程/程序/jieba分词/不带标签短信.txt"
4 fnOutput = "/Users/junxu/Documents/Work/上课/数据科学导论课程/程序/jieba分词/不带标签短信-seg.txt"
5 fin = open(fnInput, "rb")
6 fout = open(fnOutput, "w+")
7 numline = 0
8 for line in fin:
9     seg_list = jieba.cut(line, cut_all=False)
10    fout.write(" ".join(seg_list))
11    numline = numline + 1
12 fin.close()
13 fout.close()
14 print("Processed %d lines" % numline)
```

Processed 200000 lines

分词结果

不带标签短信.txt 不带标签短信-seg.txt

1 .x月xx日推出凭证式国债x年期x.xx.xx%，x年期x.xx%到期一次还本付息。真情邮政，为您竭诚服务！ 咨询电话xxxx-xx
2 x强度等级水泥的必要性和可行性进行深入研究
3 Don'tSellProduct
4 以上比赛规则由江苏科技大学教职工摄影协会负责解释
5 坐12个小时飞机身体已经疲惫不堪
6 为什么不能是你③以多数人的努力程度
7 地址位于天津市滨海新区响罗湾旷世国际大厦A座1801室
8 它是由AlexanderStepanov、MengLee和DavidRMusser在惠普实验室工作时所开发出来的
9 前首席执行官迪克·科斯特洛或将离开
10 zuzu气垫BB拍上去过几分钟后就会和皮肤越来越贴
11 年薪20万以上的工作岗位普遍较少
12 适当运用收纳设计把客厅改造成书房
13 被扭曲的独白拼凑折射出人性自私的阴暗面
14 命运永远会偏袒勇者...加油...
15 庆x'x节本会所优惠活动，为答谢新老顾客的支持与厚爱，，面部特卡：xxx元/xx次，身体活动，带脉减小肚腩：xxxx元/xx次，，肠胃
16 斯柯达对外发布了全新FabiaR5概念版
17 开头先夸一下自己：最近有不少人跟我说话都是这么开头的：哎呀
18 这样的ladybeard给吓坏了崩坏吧

不带标签短信.txt 不带标签短信-seg.txt

1 . x 月 xx 日 推出 凭证式 国债 x 年期 x . xx . xx% , x 年期 x . xx% 到期 一 次 还本付息 。 真情 邮政 ， 为 您 竭诚服务 ! 咨询电话 xxxx - xx
2 x 强度 等级 水泥 的 必要性 和 可行性 进行 深入研究
3 Don ' tSellProduct
4 以上 比赛 规则 由 江苏 科技 大学 教职工 摄影 协会 负责 解释
5 坐 12 个 小时 飞机 身体 已经 疲惫不堪
6 为 什么 不能 是 你 ③ 以 多数 人 的 努力 程度
7 地址 位 于 天津市 滨海 新区 响 罗湾 旷世 国际 大厦 A座 1801 室
8 它 是 由 AlexanderStepanov 、 MengLee 和 DavidRMusser 在 惠普 实验室 工作 时 所 开发 出来的
9 前 首席 执行官 迪克 · 科斯特 洛 或 将 离开
10 zuzu 气垫 BB 拍 上去 过 几分钟 后 就 会 和 皮 肤 越来越 贴
11 年 薪 20 万 以上 的 工作 岗位 普遍 较 少
12 适 当 运用 收纳 设计 把 客厅 改造 成 书 房
13 被 扭 曲 的 独白 拼凑 折射 出 人 性 自私 的 阴 暗 面
14 命 运 永远 会 偏 袒 勇 者 ... 加 油 ...
15 庆 x ' x 节 本 会 所 优 惠 活 动 ， 为 答 谢 新 老 顾 客 的 支 持 与 厚 爱 ， ， 面 部 特 卡 : xxx 元 / xx 次 ， 身 体 活 动 ， 带 脉 减 小 肚 脩 : xxxx 元 / xx 次 ， ， 肠 胃
16 斯 柯 达 对 外 发 布 了 全 新 FabiaR5 概 念 版

第3.3节

文本分词 与检索

3.3.2 文本检索

- ✓ 信息检索
- ✓ 倒排索引
- ✓ 排序模型

第3.3.2节 文本检索

信息检索

信息检索

- 信息检索(Information Retrieval, IR)是指从大规模的非结构化数据集中(通常指文本文档)寻找满足用户信息需求的过程
- 互联网搜索引擎是目前最常见的信息检索系统，但信息检索不局限于互联网搜索：
 - 企业搜索(如SharePoint Search)
 - 特定领域文档搜索(Scholar, Patent等)
 - 桌面搜索、Email搜索

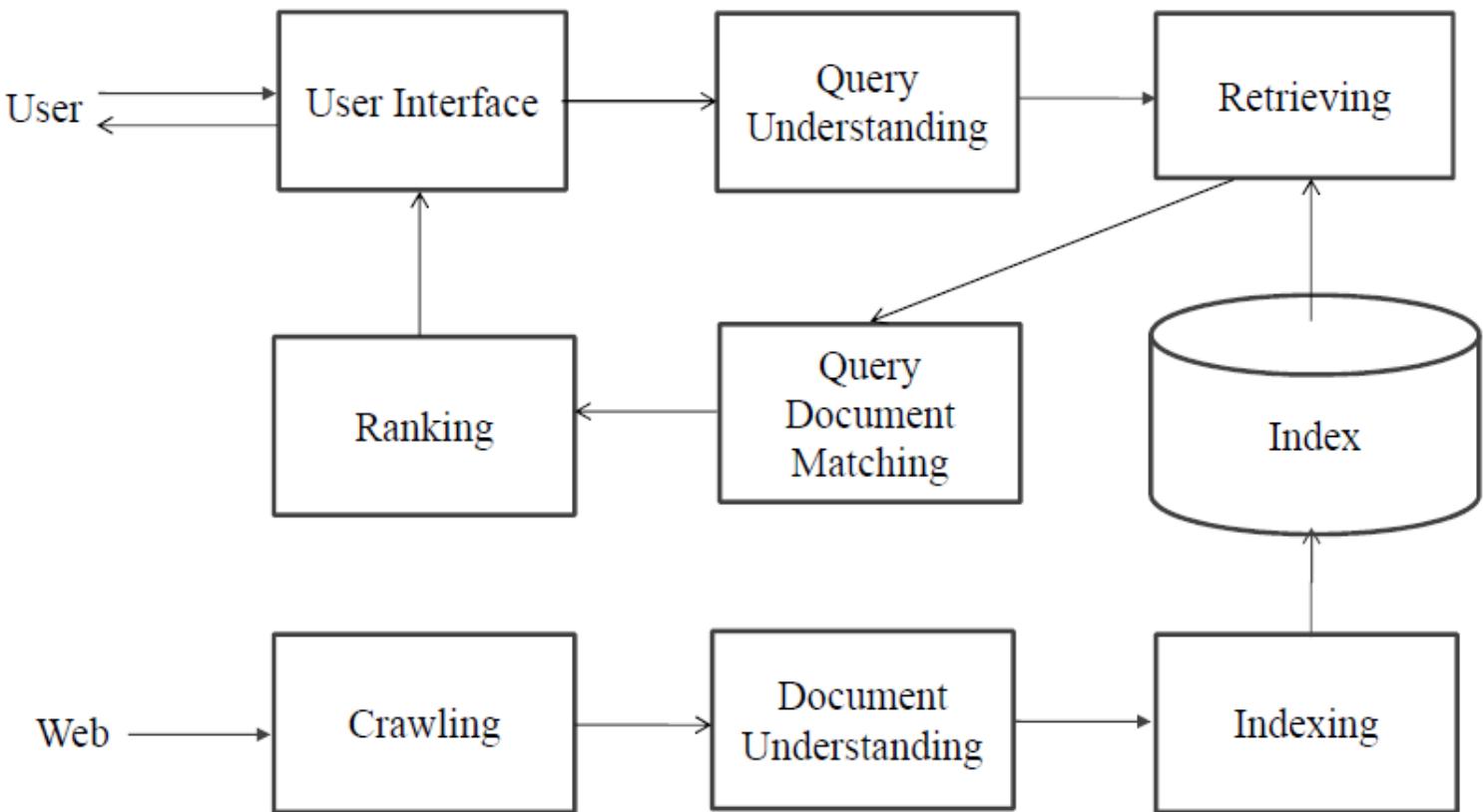
关于几个关键词

- **寻找信息**: 与构造新的信息内容(如统计归纳)不同, 信息检索只负责提供已有的信息给用户
- **非结构化数据**: 与数据库中关系数据不同, 非结构化数据不容易被计算机处理
- **信息需求**: 通常通过查询词进行表达
- **大规模数据**: 例如互联网网页、企业内部网数据等, 数据量大, 处理数据的方法需要足够高效且可扩展

对信息检索系统的基本假设

- **静态文档集合**
 - 假设在用户搜索的时刻，文档集合不发生变化
- **检索目的**
 - 从文档集合中检索出与用户的信息需求相关的文档，从而帮助用户完成某一特定任务

搜索引擎主要模块



互联网搜索引擎发展

Archie FAQ
(1990)
精确FTP文件名
搜索

World Wide
Web Wanderer
(1993)
第一个网络爬虫程序


(1993)
网站主动提交检索信息


(1993)
分析字词关系
概念搜索


(1994)
提供简单目录搜
索


(1994)
全文搜索引擎


(1994)
网页自动摘要


infoseek
(1994)
网页自动摘要，
同时提供网页目
录等其他服务


(1995)
支持自然语言搜
索和高级搜索语
法


(1995)
Inktomi公司, 抓取索引|1
千万页/天, 储存用户搜索
喜好


(1996)
自然语言提问, 优先
提供答案



(1998)


(1999)
Fast公司, 利用ODP自
动分类改善搜索


(2000)
搜索结果自动聚类


(1997)
第一个中文搜索引擎


(2000)
目前为止最成功的中
文搜索引擎

互联网搜索引擎发展

- 第一代 (1994—1998)
 - 基于语法的查询-内容匹配 (syntactic matching)
- 第二代 (1998—约2008)
 - 不仅仅考虑网页内容与查询的匹配(beyond “on-page” content)
 - 同时考虑链接分析、用户点击路径等
- 第三代 (2008—约2015)
 - 结果页面不仅仅显示网页链接 (Beyond 10 blue links)
 - User intension, short cut, rich content
- 第四代
 - 移动 ? 信息流 ? 个性化 ?
 - 搜索 + 推荐 + 广告 ?

文本检索挑战#1

- 如何从大规模集合快速找到包含指定关键词的文档（候选集）？
 - 大规模文档集合
 - 字典规模庞大
 - 无法提前预知用户输入的查询
 - 快速(< 0.1s)
- 为一个查询遍历文档集合不是一个可行的选择

信息检索 大数据

All News Images Videos Maps More Settings Tools

About 28,900,000 results (0.46 seconds)

Google 学术: 信息检索 大数据
大数据时代的图书馆与数据素养教育 - 张晨 - Cited by 39
知识咨询: 大数据时代图书馆的知识服务增长点 - 王天泥 - Cited by 102
大数据视角下的情报研究与情报研究技术 - 李广建 - Cited by 73

[PDF] 面向网络大数据的信息检索与挖掘 - ChinaXiv
chinavix.org/user/download.htm?id=6537 ▾ Translate this page
本文介绍了面向网 络**大数据**的深度检索与挖掘的一系列关键技术，包括用户查询理解与处理、文档建模与理解及检索模型等。关键词：**信息检索** 数据挖掘查询理解话题 ...

视频信息检索与数据挖掘- 程序园
www.voidcn.com/article/p-vkkyldsuc.html ▾ Translate this page
Nov 1, 2012 - 引子-**信息检索** **信息检索**是用来处理文本数据的技术, **信息检索**领域的传统 ... 发展结构,它为**大数据**量视频的导航和浏览提供了一种非常好的手段。

互联网大规模数据分析技术
<https://www.xuetangx.com/courses/course-v1:WUT...T1/about> ▾ Translate this page
如今我们处于**大数据**的时代, 互联网大规模数据分析这门课程带大家进入分析和处理 ... 接下来解剖**信息检索**和推荐系统两大Web主流应用的原理和模型, 并通过例子 ...

文本检索挑战#2

- 如何以一种合适的方式把候选集展示给用户?
- 传统展示：展示所有结果集合
 - 文档太多：难以浏览
 - 文档太少：找不到满意结果
- 排序
 - 按照相关度从上往下排序
 - 辅助展示手段：（动态）摘要与剽

The screenshot shows a search results page with the query '信息检索' entered in the search bar. The results are listed below:

- 信息检索（一种信息技术）_百度百科**
baike.baidu.com/view/45496.htm ▾ Translate this page
信息检索（Information Retrieval）是指信息按一定的方式组织起来，并根据信息用户的需要找出有关的信息的过程和技术。狭义的**信息检索**就是**信息检索**过程的后半 ...
- 信息检索- 维基百科，自由的百科全书**
https://zh.wikipedia.org/zh/信息检索 ▾ Translate this page
資訊檢索（英语：Information Retrieval）是指搜尋資訊的科學，如在文件中搜尋資訊、搜尋文件本身、搜尋描述文件的metadata或是在資料庫中進行搜尋，無論是在相關 ...
- 文本信息检索- 维基百科，自由的百科全书**
https://zh.wikipedia.org/zh/文本信息检索 ▾ Translate this page
文本**信息检索**是针对文本的**信息检索**技术。在技术社区中，文本**信息检索**常常被等同于**信息检索**技术本身。相对视频、音频检索而言，文本**信息检索**是发展较快也较 ...

第3.3.2节 文本检索

倒排索引

文本检索挑战#1

- 如何从大规模集合快速找到包含指定关键词的文档（候选集）？
 - 大规模文档集合
 - 字典规模庞大
 - 无法提前预知用户输入的查询
 - 快速(< 0.1s)
- 为一个查询遍历文档集合不是一个可行的选择

The screenshot shows a search results page from Google Scholar. The search bar at the top contains the query "信息检索 大数据". Below the search bar, there are tabs for All, News, Images, Videos, Maps, More, Settings, and Tools. The "All" tab is selected. A message indicates "About 28,900,000 results (0.46 seconds)". The results list includes several entries:

- [PDF] 面向网络大数据的信息检索与挖掘 - ChinaXiv
chinaxiv.org/user/download.htm?id=6537 ▾ Translate this page
- 知识咨询: 大数据时代图书馆的知识服务增长点 - 王天泥 - Cited by 102
- 大数据视角下的情报研究与情报研究技术 - 李广建 - Cited by 73
- [PDF] 视频信息检索与数据挖掘- 程序园
www.voidcn.com/article/p-vkkyldsuc.html ▾ Translate this page
- Nov 1, 2012 - 引子-信息检索 信息检索是用来处理文本数据的技术, 信息检索领域的传统 ... 发展结构, 它为大数据量视频的导航和浏览提供了一种非常好的手段。
- 互联网大规模数据分析技术
<https://www.xuetangx.com/courses/course-v1:WUT...T1/about> ▾ Translate this page
- 如今我们处于大数据的时代, 互联网大规模数据分析这门课程带大家进入分析和处理 ... 接下来解剖信息检索和推荐系统两大Web主流应用的原理和模型, 并通过例子 ...

Text data in 1650: Shakespeare



- 用户查询问题：莎士比亚的哪部作品包含单词 Brutus 和 Caesar？
- 最直接的想法：逐个浏览莎士比亚的作品，找出所有符合条件的作品集合
- 但是对于文本检索而言，这不是一个好主意
 - 慢！！！ 因为文档集合可能很大

单词-文档共现矩阵

	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	
...							

- 每一行表示一个单词，每一列表示一个文档
 - 1：单词在文档中出现过至少一次。例如：单词BRUTUS在文档Hamlet中出现
 - 0：单词未在文档中出现。例如：单词BRUTUS未在文档Othello中出现

回答上述查询

- 莎士比亚的哪部作品包含单词 Brutus 和 Caesar?
 - 找出Brutus对应的向量: 110100
 - 找出Caesar对应的向量: 110111
 - Bitwise AND: 110100 AND 110111 = 110100
- 答案: 110100
 - Anthony and Cleopatra
 - Julius Caesar
 - Hamlet

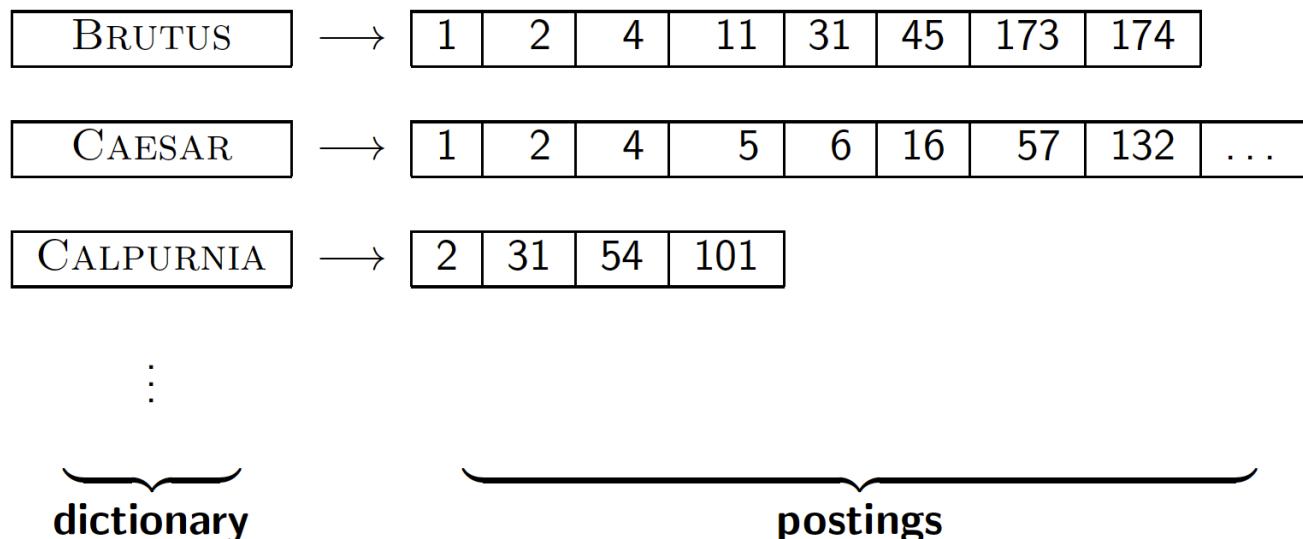
	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	1	1	0	0	0	1	
BRUTUS	1	1	0	1	0	0	
CAESAR	1	1	0	1	1	1	
CALPURNIA	0	1	0	0	0	0	
CLEOPATRA	1	0	0	0	0	0	
MERCY	1	0	1	1	1	1	
WORSER	1	0	1	1	1	0	
...							

集合规模很大，这个方法还有效吗？

- 例如 $500,000 * 1,000,000$ 规模的矩阵
 - 100万个文档，50万个不同的单词
 - 矩阵规模： $5 * 10^{11} ! !$
 - 直接存取已经不现实
- 好消息：矩阵中只有很少的值为1，绝大部分都为0
 - 假设平均文档长度为500
 - 矩阵中1的个数为： $500 * 1,000,000 = 5 * 10^8 \ll 5 * 10^{11}$
 - 平均1000个位置才会出现一次1
- 如何利用上述性质？
 - 稀疏矩阵存储方法：只存1

倒排索引

- 对字典中的每一个单词t，只存储包含了t的文档列表

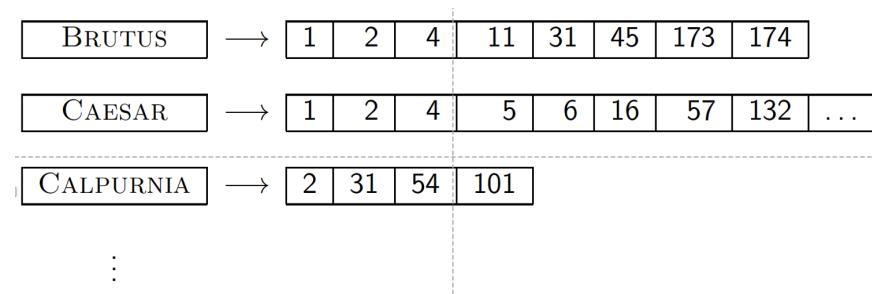


为什么叫倒排？

- 正排：通过文档ID找到文档内容

	Brutus	Caesar	Calpurnia	Anthony
D1	1	1	0	0		
D2	1	1	1	0		
D3	0	0	0	0		
D4	0	1	0	0		
...						

- 倒排：通过文档的内容找到文档ID



构建倒排索引的步骤

- 爬取所需要索引的文档集合

Friends, Romans, countrymen. So let it be with Caesar ...

- 分词

Friends Romans countrymen So ...

- 对词进行进一步处理（语言相关），如：小写、找词根、去除停用词

friend roman countryman so ...

- 构建倒排索引，包括： dictionary和postings

倒排索引构建步骤

Doc 1. I did enact Julius Caesar: I was killed i' the Capitol; Brutus killed me.
Doc 2. So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:



Doc 1. i did enact julius caesar i was killed i' the capitol brutus killed me
Doc 2. so let it be with caesar the noble brutus hath told you caesar was ambitious

term	docID
i	1
did	1
enact	1
julius	1
caesar	1
i	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

1. 文本分词和预处理

2. 生成单词-文档表

term	docID
i	1
did	1
enact	1
julius	1
caesar	1
i	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2



term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
i	1
i	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	1
so	2
the	2
the	2
told	2
you	1
you	2
was	1
was	2
with	2

⇒

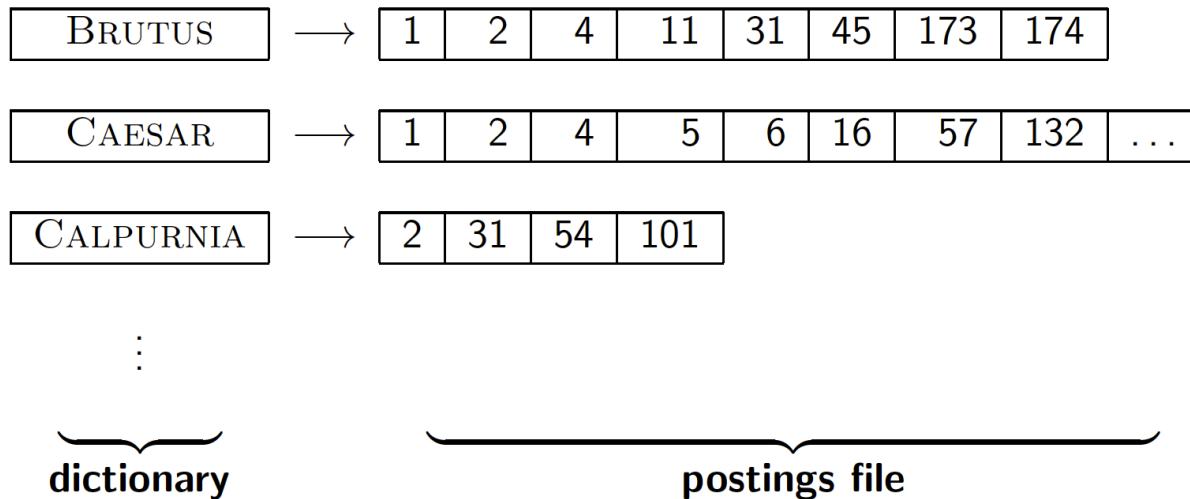
term	doc. freq.
ambitious	1
be	1
brutus	2
capitol	1
caesar	1
caesar	2
did	1
enact	1
hath	1
i	1
i	1
i'	1
it	1
julius	1
killed	1
killed	1
let	1
me	1
noble	1
so	1
the	2
the	2
told	1
you	1
you	2
was	2
with	1

→ postings list:

ambitious	1	→ [2]
be	1	→ [2]
brutus	2	→ [1] → [2]
capitol	1	→ [1]
caesar	2	→ [1] → [2]
caesar	2	→ [1]
did	1	→ [1]
enact	1	→ [1]
hath	1	→ [2]
i	1	→ [1]
i	1	→ [1]
i'	1	→ [1]
it	1	→ [2]
julius	1	→ [1]
killed	1	→ [1]
killed	1	→ [2]
let	1	→ [2]
me	1	→ [1]
noble	1	→ [2]
so	1	→ [2]
the	2	→ [1] → [2]
the	2	→ [2]
told	1	→ [2]
you	1	→ [2]
you	2	→ [1] → [2]
was	2	→ [2]
with	1	→ [2]

3. 排序

4. 计算文档频率，构建postings list



5. 形成字典和postings file

倒排索引如何快速检索？

- 莎士比亚的哪部作品包含单词 Brutus和 Calpurnia?
 - 在字典中找到Brutus
 - 读入Brutus对应的postings list （注意posting list存在文件中）
 - 在字典中找到Calpurnia
 - 读入Calpurnia对应的postings list
 - 求两个postings list的交集
 - 返回求交的结果

举例

BRUTUS → [1] → [2] → [4] → [11] → [31] → [45] → [173] → [174]

CALPURNIA → [2] → [31] → [54] → [101]

Intersection ⇒



BRUTUS → [1] → [2] → [4] → [11] → [31] → [45] → [173] → [174]

CALPURNIA → [2] → [31] → [54] → [101]

Intersection ⇒ [2]



BRUTUS → [1] → [2] → [4] → [11] → [31] → [45] → [173] → [174]

CALPURNIA → [2] → [31] → [54] → [101]

Intersection ⇒ [2] → [31]

- 要求posting lists已排序

第3.3.2节 文本检索

排序模型

文本检索挑战#2

- 如何以一种合适的方式把候选集展示给用户?
- 传统展示：展示所有结果集合
 - 文档太多：难以浏览
 - 文档太少：找不到满意结果
- 排序
 - 按照相关度从上往下排序
 - 辅助展示手段：（动态）摘要与剽

The screenshot shows a search results page for the query "信息检索". The search bar at the top contains the text "信息检索". Below the search bar, there are three search results listed:

- 信息检索（一种信息技术）_百度百科**
baike.baidu.com/view/45496.htm ▾ Translate this page
信息检索（Information Retrieval）是指信息按一定的方式组织起来，并根据信息用户的需要找出有关的信息的过程和技术。狭义的**信息检索**就是**信息检索**过程的后半 ...
- 信息检索- 维基百科，自由的百科全书**
https://zh.wikipedia.org/zh/信息检索 ▾ Translate this page
資訊檢索（英语：Information Retrieval）是指搜尋資訊的科學，如在文件中搜尋資訊、搜尋文件本身、搜尋描述文件的metadata或是在資料庫中進行搜尋，無論是在相關 ...
- 文本信息检索- 维基百科，自由的百科全书**
https://zh.wikipedia.org/zh/文本信息检索 ▾ Translate this page
文本**信息检索**是针对文本的**信息检索**技术。在技术社区中，文本**信息检索**常常被等同于**信息检索**技术本身。相对视频、音频检索而言，文本**信息检索**是发展较快也较 ...

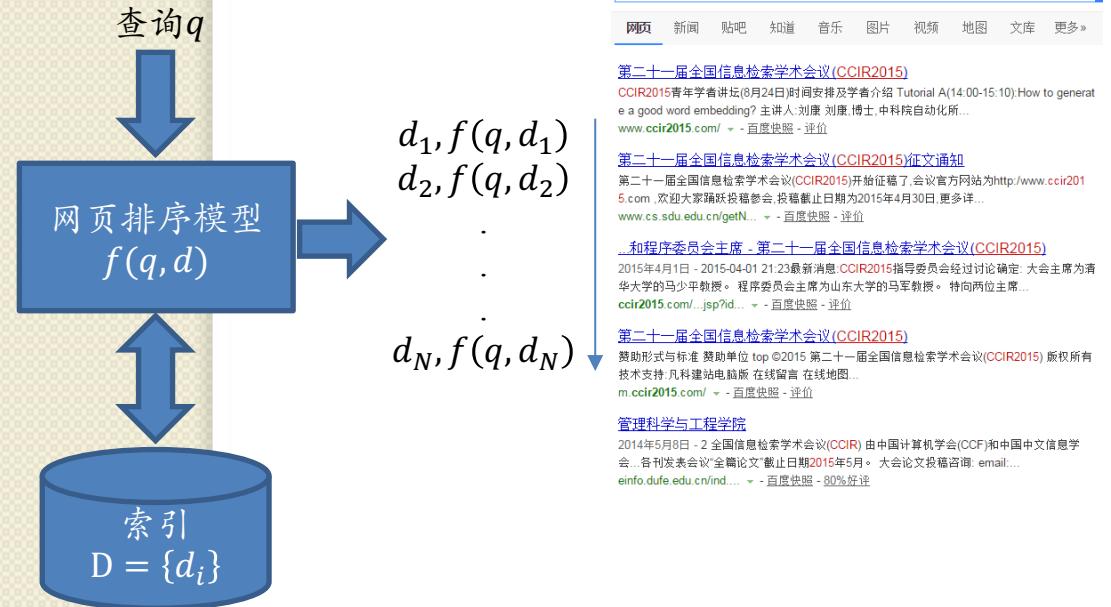
A large blue arrow points downwards from the bottom of the slide content towards the third search result.

展示集合的优劣

- 优势
 - 专家用户：他们精确了解自己的需求和文档集合的内容
 - 程序访问：对于程序而言，浏览1000甚至更多的文档没有任何压力
- 劣势：普通用户难以使用
 - 普通通常很难精确描述自己的需求，对文档结合也缺乏准确的了解
 - 用户不可能逐个浏览1000个文档来得到信息

互联网搜索时代，上述劣势变得更加明显

排序所带来的优势



- 大的候选集合不再成为大的阻碍
- 通常一页只展示10个结果(ten blue links)
- 实践表明, 排序提供了一种更加平滑的交互方式
 - 用户不会被大量的信息淹没
 - 用户可以浏览更多的结果

排序的准则

- 在不同的搜索应用中有不同的排序准则
- 明确的排序准则
 - 时间（如学术搜索、Email搜索、新闻搜索）
 - 引用量(学术搜索)
 - 评论数、成交量、下载量 (商品搜索、apps搜索)
 -
- 模糊的排序准则
 - 相关度
 - 重要性

网页搜索排序准则

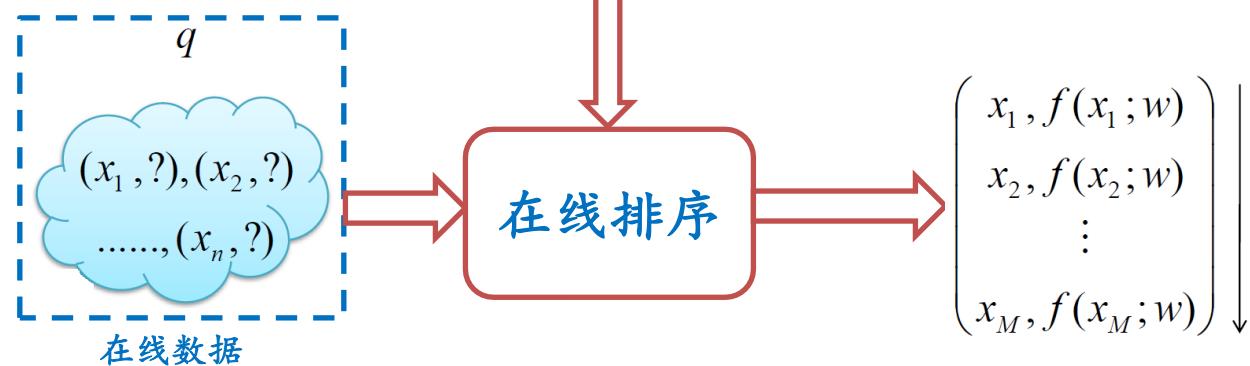
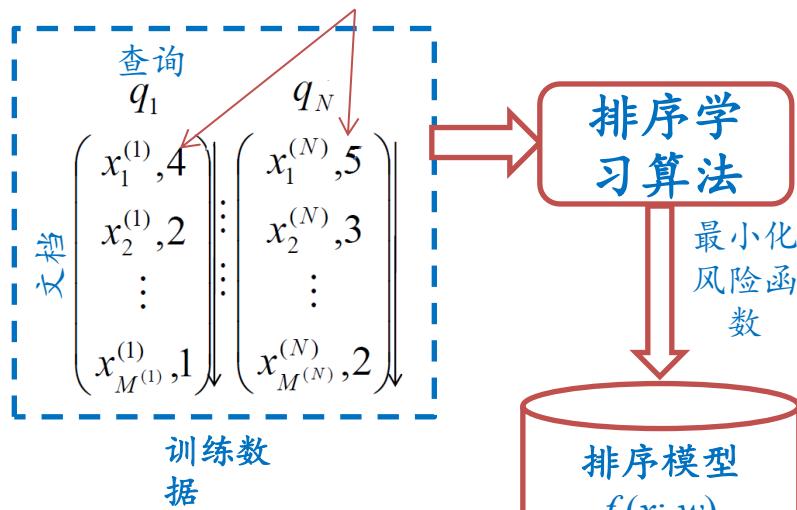
- 相关性排序
 - 模糊的排序准则，如何精确定义？
- 研究者们试图从查询和文档性质，以及它们中的词的共现关系，计算查询-文档的相关度
 - 共现次数(次数越多越相关)
 - 词的重要性
 - 文档长度
 - 文档重要性(微软主页和苹果主页谁更重要？)

基于分值的排序模型

- 如何实现“相关性”排序?
 - 对每一个查询-文档对进行打分
 - 分值体现查询与文档的“匹配”程度
 - 按照分值从大到小对文档进行排序
- 关键问题：如何计算分值?
 - 如果查询词不在文档中出现，分值为0
 - 查询词出现的次数越多，分值越高
 - 文档中包含不同的查询词越多，分值越高
 - 排序模型通常也被称为**检索模型**

排序学习

人工标注





Thanks!