

第5节.内容索引子系统

艾清遥

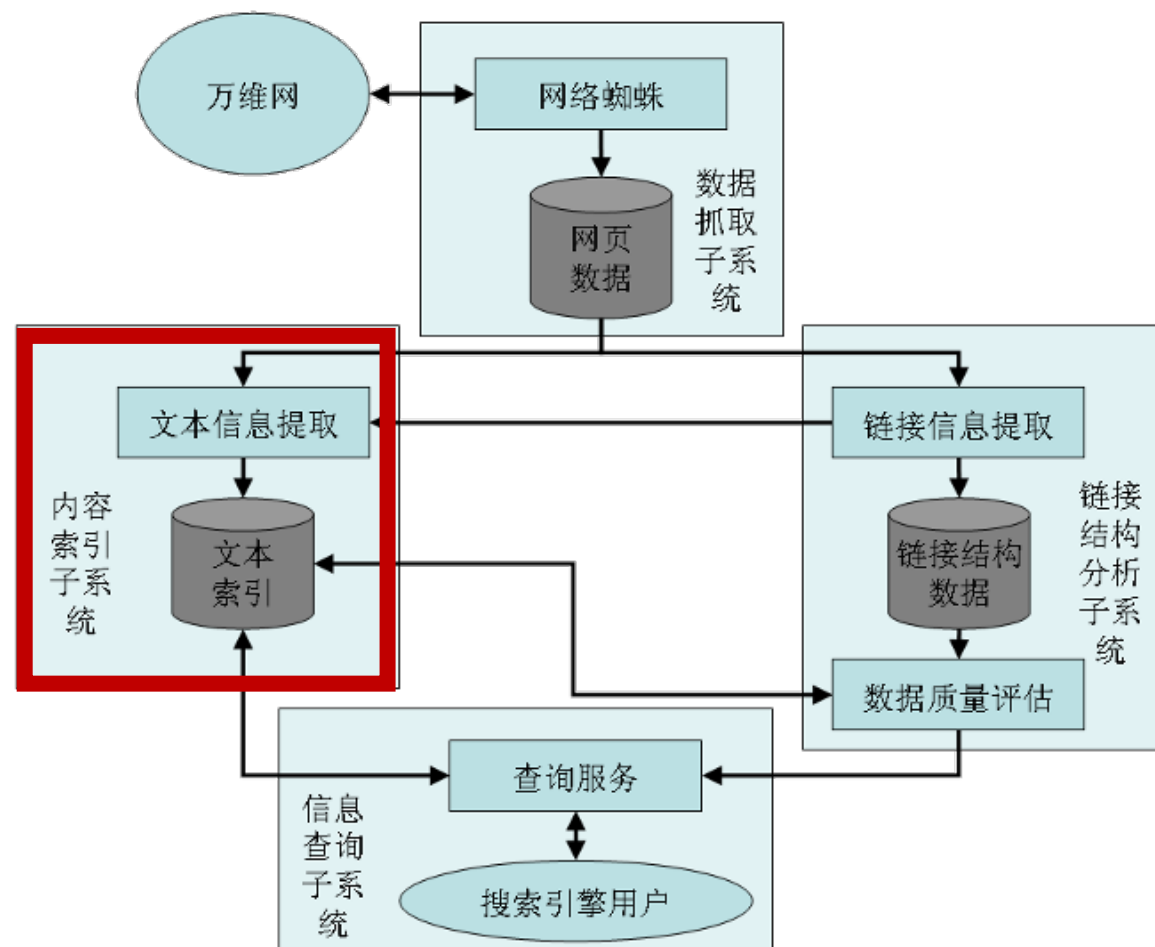
清华大学计算机系

清华大学互联网司法研究院

2023年3月21日

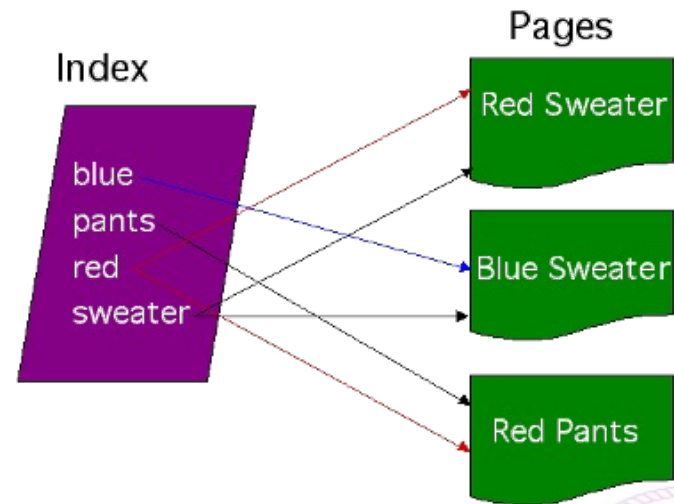


搜索引擎体系结构



引言：内容索引子系统

- 索引子系统的主要功能
 - 建立倒排索引
 - 建立正排索引
 - 提供索引查询服务
- 索引子系统面临的挑战
 - 高效利用系统资源
 - 提供可扩展的索引服务



引言：基本概念回顾

- 词项 (Term): 索引子系统处理的最小语义单位
- 倒排索引结构 (Inverted Index):
 - 以词项为核心的索引组织形式

Term 1	Doc 1, pos 1	Doc 1, pos 2	...	Doc p, pos q
Term 2	Doc 1', pos 1'	Doc 1', pos 2'	...	Doc p', pos q'
...				
Term N	Doc 1 ⁽ⁿ⁾ , pos 1 ⁽ⁿ⁾	Doc 1 ⁽ⁿ⁾ , pos 2 ⁽ⁿ⁾	...	Doc p ⁽ⁿ⁾ , pos q ⁽ⁿ⁾

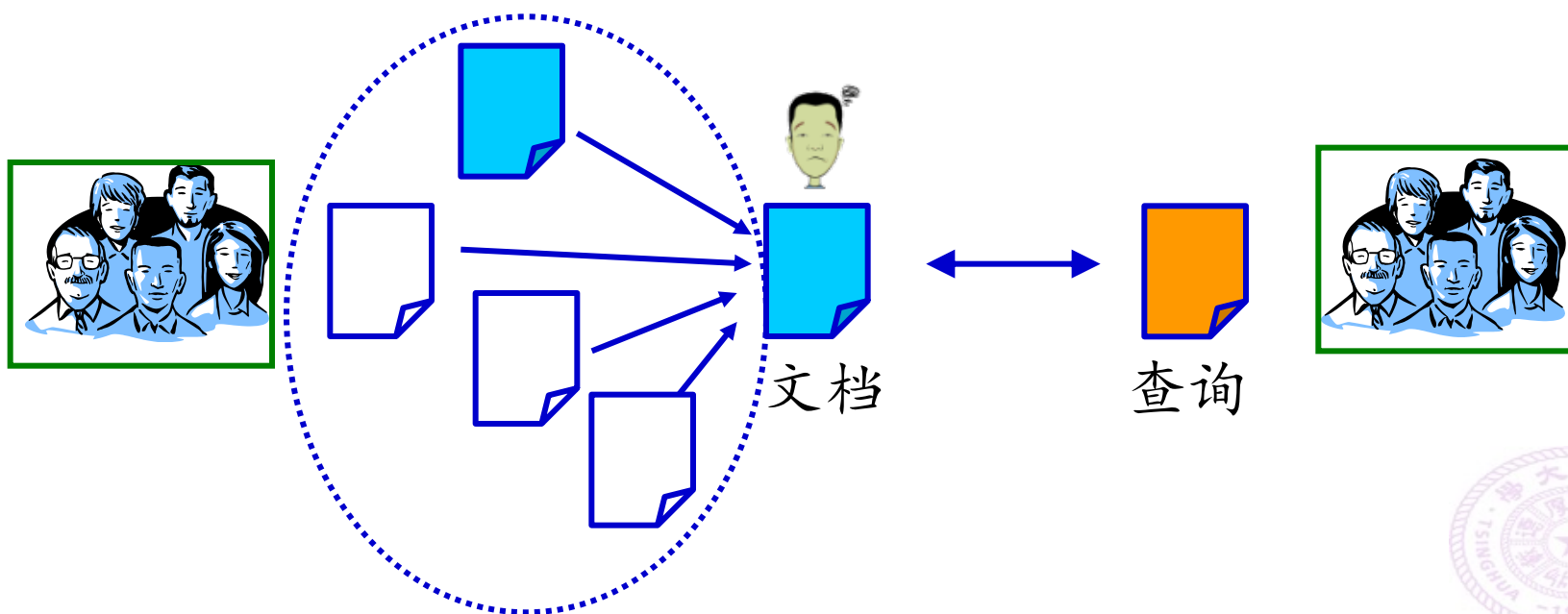
- 与人类的长期记忆原理类似
- 适合于主流商业搜索引擎的交互方式



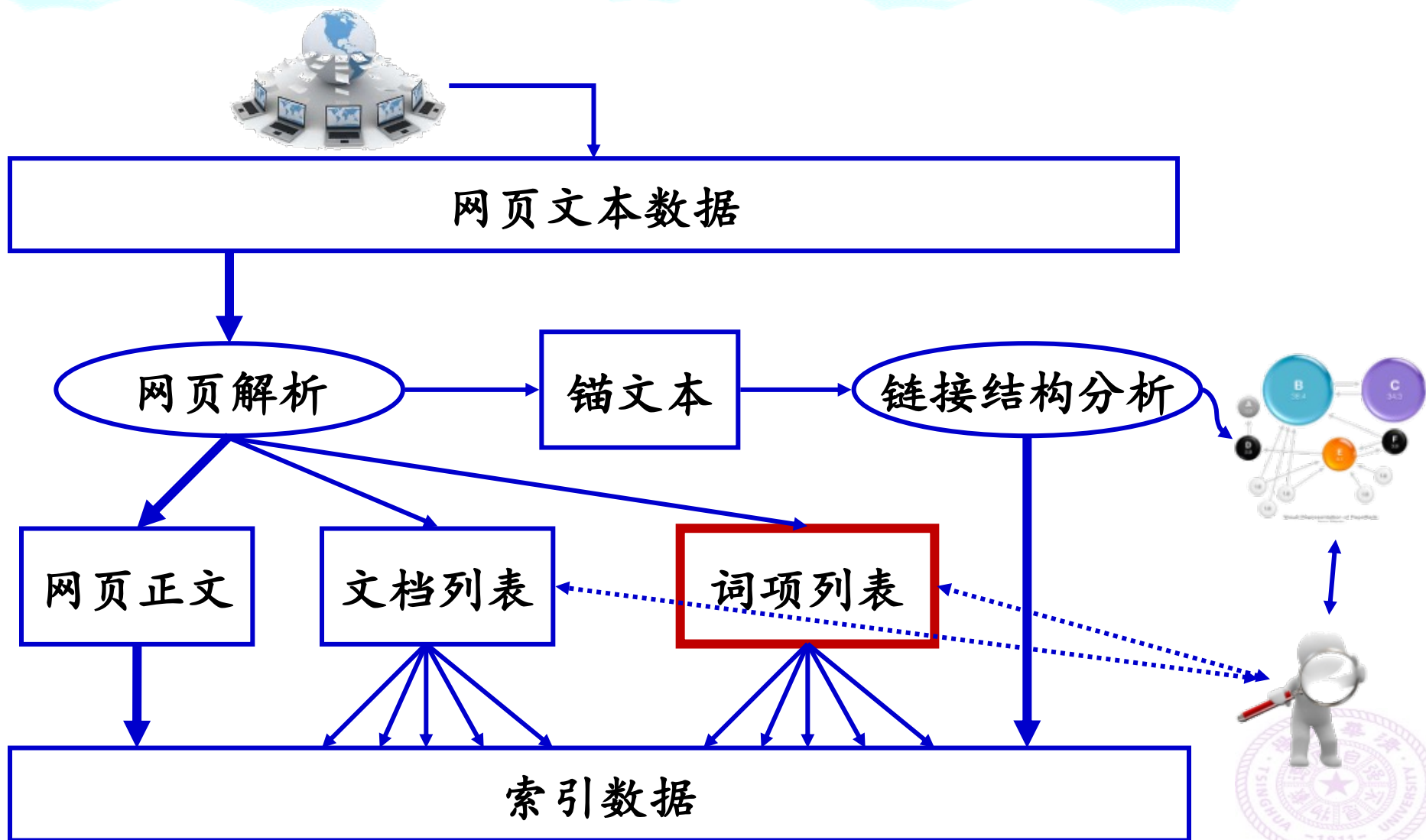
引言：基本概念回顾

- 锚文本：使网页内容与查询内容更好匹配
 - 对链接到网页的内容的描述

`北大`



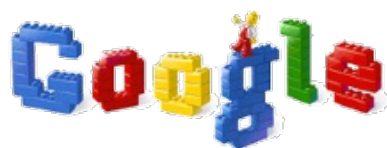
引言：内容索引子系统基本架构



本节概要

1. 词项列表构建
2. 索引数据结构
3. 索引建立过程
4. 索引查询过程

以**Google**早期
索引结构为例



1. 词项列表构建

• 1.1 词项的作用

- 作为语义的最小单位出现
 - 需要符合网络数据/用户的表述习惯
 - 根据热门程度调整词项列表：词库更新
- 作为倒排索引的索引项出现
 - 选取合理的词项数目以高效利用系统资源
 - 数目太大：词项编码长，冗余信息多，浪费空间；
 - 数目太小：词项对应的索引条目多，I/O压力大
- 案例：如何处理数字组成的词项？



1. 词项列表构建

- 案例：如何处理数字组成的词项？

Baidu 百度 新闻 网

3810020265

淘宝网 正品教师法治教育读本 01130 价
情况：- 浏览里：-...
www.361du.net/p1-3810020265.h

找到相关结果1个

Baidu 百度 新闻 网页 贴吧 知道 MP3 图片 视频 地图 更多▼

13810020265

 **手机归属地**
手机号码"13810020265" 北京 中国移动 GSM
手机归属地数据由[手机在线](#)提供
☒ [展开手机归属地查询](#)

[北京 北京1381002手机号段|移动动感地带卡|区号010 - 好想查号网](#)
北京 北京1381002手机号段的10000个号码： 13810028499,13810028498,13810028497,13810028496,13810028495,13810028494,13810028493,13810028492,13810028491,13810028490,13810028489...
hxcweb.com/1381002.html 2011-1-20 - [百度快照](#)

[北京市北京市1381002手机号段 1381002号段号码查询 手机号码归属...](#)
查询首页 北京市 北京手机号段北京 北京市，简称“京”，是中华人民共和国首都，四个中央直辖市之一，全国第二大城市及政治、交通和文化中心。北京位于华北...
www.btdxd.com/mobile/beijing_1381002.html 2011-3-10 - [百度快照](#)



1. 词项列表构建

- 1.2 英文词项处理

- 大小写统一化、命名实体识别(O'Neal, Los Angeles)
- 去除停用词(stop word)
 - 停用词：出现频率高，语义信息量小的词
- 在尽量保存有用内容的前提下，减少索引空间浪费，保证系统运行效率
- 极大精简索引结构：去除停用词后，倒排索引可以缩小40%，显著降低I/O负担



•1.2 英文词项处理



您搜索的内容: department of computer science and technology tsinghua



Copyright ©2009-2011 Depa

1. 词项列表构建

• 1.2 英文词项处理

- 取词干(stemming)
- 词干(stem): 删除词的词缀后剩余的部分
- Compute, computer, computing => comput
- 同一词根的不同变形缩减为同一个概念
- 合并索引项: 保证语义不变的情况下缩小索引规模, 提高召回率
- Porter's stemming algorithm



1. 词项列表构建

• 1.2 英文词项处理

- 取词干

- 词干(stem)

- Compu

- 同一词

- 合并索

- Porter's

可能的风险: inform != information

The screenshot shows a Google search interface. The search bar contains the text "inform retrieve". Below the search bar, the results are displayed. The first result is "Information Retrieval - Springer" with a link to "link.springer.com/journal/10791". The second result is "Information Retrieval - Wikipedia, the free encyclopedia" with a link to "en.wikipedia.org/wiki/Information_retrieval". The third result is "Information Retrieval - Wikipedia, the free encyclopedia" with a link to "en.wikipedia.org/wiki/Information_retrieval".

网页 图片 视频 新闻 更多 搜索工具

找到约 8,580,000 条结果 (用时 0.28 秒)

显示的是以下查询字词的结果: inform retrieval
仍然搜索: inform retrieve

Information Retrieval - Springer
link.springer.com/journal/10791 翻译此页
Subscription e-journal dedicated to theory and experimentation in information retrieval.
Sample copy available.

Information Retrieval – incl. option to publish open access - S...
www.springer.com/computer/database...information.../10791 翻译此页
The journal provides an international forum for the publication of theory, algorithms, and
experiments across the broad area of information retrieval. Topics of ...

Information retrieval - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Information_retrieval 翻译此页
Information retrieval is the activity of obtaining information resources relevant to an
information need from a collection of information resources. Searches can be ...

率



1. 词项列表构建

• 1.2 英文词项处理

• 预处理对词项列表规模的影响

	独立词项			词项-文档			词项-文档-位置		
	number	$\Delta\%$	T%	number	$\Delta\%$	T%	number	$\Delta\%$	T%
unfiltered	484,494			109,971,179			197,879,290		
no numbers	473,723	-2	-2	100,680,242	-8	-8	179,158,204	-9	-9
case folding	391,523	-17	-19	96,969,056	-3	-12	179,158,204	-0	-9
30 stop words	391,493	-0	-19	83,390,443	-14	-24	121,857,825	-31	-38
150 stop words	391,373	-0	-19	67,001,847	-30	-39	94,516,599	-47	-52
stemming	322,383	-17	-33	63,812,300	-4	-42	94,516,599	-0	-52

Manning *et. al.*, Reuters-RCV1

• “Rule of 30”

- Top 30 words \Rightarrow 30% appearances



1. 词项列表构建

• 1.3 中文词项处理

- 编码处理，全/半角处理等
- 去除停用词：的，是，和，在
- 风险：停用词的歧义性
 - 的确，和平，存在
- 案例：不同搜索引擎处理停用词的策略



1. 词项列表构建

• 1.3 中文

• 编码

• 去除

• 风险:

• 的确

• 案例:

瘦瘦的

瘦瘦的 - 百度百科



类型: 单曲
导演: 陈宏一
简介: 《瘦瘦的》是来自马来西亚的歌手梁静茹的一首歌, 由姚若龙作词, 陈小霞谱曲, 收录在2005年发行的专辑《丝路》中。
歌手简介 歌词 歌曲鉴赏
百度百科

瘦瘦的 - 视频大全 - 高清在线观看



【4K修复】梁静茹 - 瘦瘦的 M
哔哩哔哩



磁带试听梁静茹《瘦瘦的》
西瓜视频



多少人想减肥,然而瘦的却是心
西瓜视频



终于找到《瘦瘦的》最好听的版本,道尽爱情的...
好看视频



两性之间,为什么女人喜欢瘦瘦的男人,瘦男人有...
好看视频



人瘦是什么原因造成的? 太瘦也不是好事, 当心...
好看视频

瘦瘦的

搜狗已为您找到约2,005,024条相关结果

相关推荐: 瘦瘦app 瘦瘦app下架原因 瘦瘦果 瘦瘦的图片

安卓版瘦瘦下载

安卓版 iPhone版



瘦瘦
版本: 6.9.12 大小: 31.7 MB 更新: 2021-10-28
系统: Android4.0或更高版本 下载来源: 应用宝
立即下载



扫码下载

搜狗下载 - xiazai.sogou.com

瘦瘦的

5年前 - 今日碎碎念配乐: Boyzone 《No Matter What》 本公众号执行编辑: 四月不减肥五月徒伤悲的磊磊
张源来信 - weixin.qq.com - 2018-05-07



人很瘦是什么原因 - 有来医生

"人很瘦有很多原因,首先有些人群天生体质偏瘦,但是检查指标时都是健康无异常,这... 机体也会分解蛋白质和脂肪,所以也容易出现人很瘦的情况.人很瘦还应排除..."

点击播放 01'08"

罗莉 - 主任医师 - 内分泌科 - 安徽医科大学第一附属医院 三甲甲等
有来医生 - www.youlai.cn/a... - 2021-4-24 - 快照

1. 词项列表构建

• 1.3 中文词项处理

- 中文分词问题：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
- 基于词典的分词方法
- 基于理解的分词方法
- 基于统计的分词方法



1. 词项列表构建

• 1.3 中文词项处理：基于词典的分词方法

- 由苏联专家在1950年代末提出
- 将待分析的汉字串与一个“充分大的”机器词典中的词条进行匹配
- 按照扫描方向的不同：
正向匹配、逆向匹配、双向匹配
- 按照不同长度优先匹配的情况：
最大（最长）匹配和最小（最短）匹配



1. 词项列表构建

分词 就是
连续 字序
字序列 词序列
序列
重新 组合
合成 过程

- 中文分词举例（最大匹配）

分词就是将连续的字序列重新组合成词序列的过程

正向匹配：

分词就是将连续的字序列重新组合成词序列的过程

反向匹配：

分词就是将连续的字序列重新组合成词序列的过程

双向匹配：

分词就是将连续的字序列重新组合成词序列的过程

词频规则、
上下文规则等



1. 词项列表构建

- 1.3 中文词项处理：分词面临的技术挑战
 - 交集型歧义 (overlapping ambiguity)
 - 字符串ABC中，AB，BC同时为词
 - 例：组合成，结合成
 - 例：使用语言的过程就是选词组句的过程
 - 一些解决方案
 - 从语言习惯统计规律上看，哪种分割的可能性大
 - 在语境中，哪种分割方式造成的孤立字少
 - 构建新词，扩充词项列表



1. 词项列表构建

• 1.3 中文词项处理：分词面临的技术挑战

- 覆盖型歧义 (combination ambiguity)

- 字符串AB中，AB, A, B同时为词

- 例：中华人民共和国，中国科学院

- 如何处理：多粒度分词

- 未登录词识别

- 词典不可能包括所有的词

- 人名、地名、机构名、新出现的搭配、旧词新意

- 如何处理：新词发现（语言现象统计）



1. 词项列表构建

- 1.3 中文词项处理：面向搜索需求的分词
 - 分词算法的时间性能要比较高
 - 不能采用复杂语义理解的方式
 - 多粒度分词，适当引入冗余信息
 - 民进党团 => 民进党团，民进党，党团
 - **用户查询与索引数据需要采用同样的分词方式**
 - 查询：/奥尼尔/；索引：奥/尼/尔 => 查询无结果
 - 查询：/胡戎睿/；索引：胡戎/睿智/力/超群
=> 查询“胡戎睿”无结果



1. 词项列表构建

- 1.3 中文词项处理：面向搜索需求的分词
 - 采用与网络数据环境相匹配的词典
 - 词典质量直接影响分词精度
 - “众包”方式构建词典：细胞词库



1. 词项列表构建

• 案例：搜狗输入法的诞生故事

当时我用的输入法



当时我用的搜索



2005-8-6 12:15 PM

主题： 关于百度拼音输入法创意的补充



我昨天提了一个关于百度拼音输入法的建议，由于很仓促，落了几句。如果把百度拼音输入法里面低调的加入百度搜索的话，人们也会乐于接受，因为人们习惯输入法每天都开着。这时如果再加上只有文件名索引的桌面搜索的话（这样占用空间很少），人们肯定会非常喜欢的。

在天空软件站，仅紫光拼音的下载量就有400万左右，紫光的总下载量应该有至少2000万……

Re: 关于百度拼音输入法创意的补充
2005-8-6 10:06:59

您好:

谢谢您的建议.

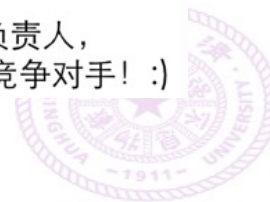
感谢您使用百度

百度

马先生:

好想法，好创意！佩服！

如果搜狗输入法的设想实现，您的功劳当属NO.1,我马上反馈给相关负责人，感谢您的支持，搜狗有您这样忠诚的网民的关注和投入，肯定能超越竞争对手！:)



1. 词项列表构建

• 1.4 词项列表的规模

Word ID	Number of docs	Pointer
Word ID	Number of docs	Pointer
.....		
Word ID	Number of docs	Pointer

} N words

- 词项数目规模庞大
- 受到内存规模限制
 - 索引建立、内容检索模块客观需要



1. 词项列表构建

- 1.4 词项列表的规模：Heaps' Law

- 对于规模为 L 个词的语料库，其独立词项数 V 为：

$$V = k \cdot L^b$$

- 其中，参数 k ：[30,100]；参数 $b \approx 0.5$ 。
- 参数 k 的取值取决于预处理方法
- 结论1：词项的规模会随着文档数目的增加持续增长，而不会稳定在某个阈值。
- 结论2：Web规模文档集合的词汇量必将非常巨大



1. 词项列表构建

- 1.4 词项列表的规模：Zipf's Law

- 语料库中的词项按频度进行排序后，记其频度为 P_1, P_2, \dots, P_n ，序号 i 为 $1, 2, \dots, n$ ，则近似有：

$$P_i \propto 1/i$$

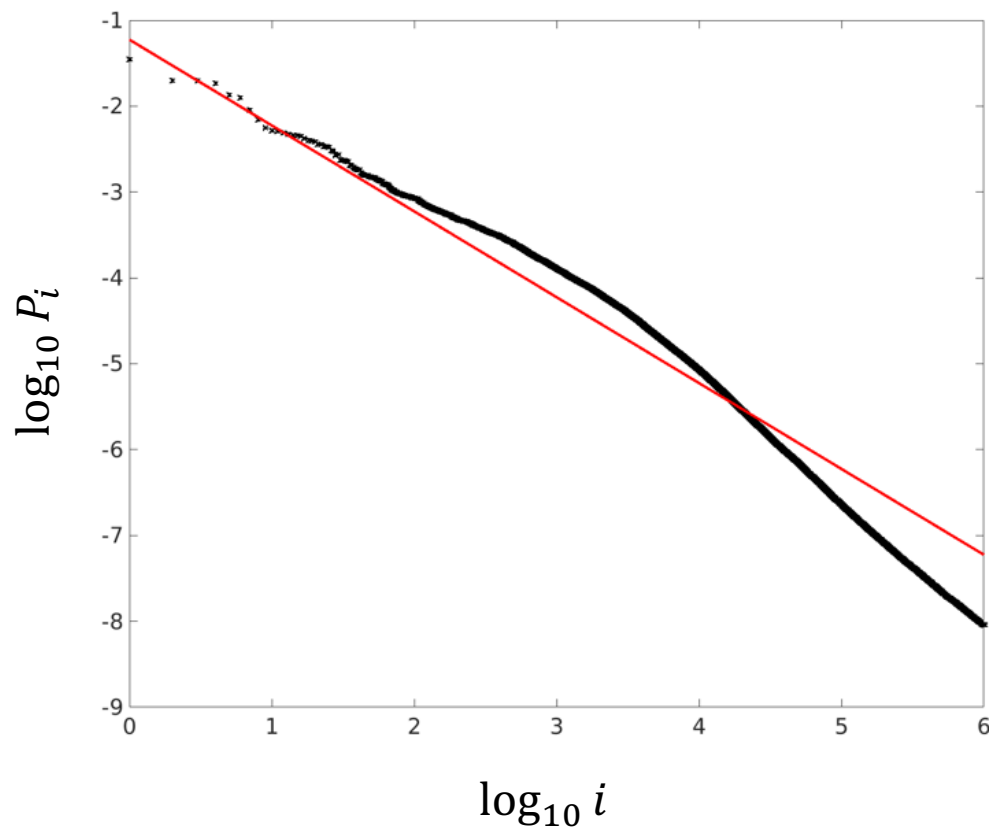
- George K. Zipf: Human behavior and the principle of least-effort. 1949. Harvard University.
- Zipf's law 反映的是人类语言规律的客观特性，在不同语言之间具有普适性。
- 是幂律 (Power Law) 的一种特殊形式



George Kingsley Zipf
(1902-1950)



Zipf's Law: Google 英文词表



$$\log_{10} \frac{P_i}{N} + \log_{10} i = 0.59$$

N : 数据集中所有英文单词频数之和

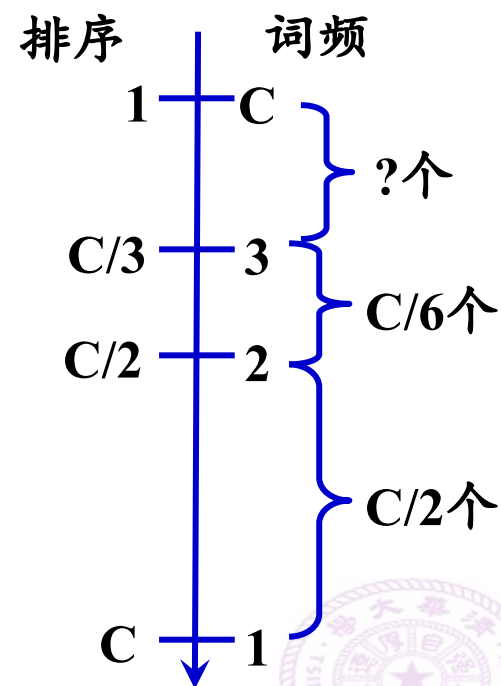


1. 词项列表构建

• 1.4 词项列表的规模：Zipf's Law

$$P_i \propto 1/i \Rightarrow i \propto 1/P_i \Rightarrow i = C/P_i$$

- $P_i = 1$ 则 $i = C$
- $P_i = 2$ 则 $i = C/2$,
词频为2的词项数为 $C/2$
- $P_i = 3$ 则 $i = C/3$,
词频为3的词项数为 $C/6$
- 低频词是词项列表的主要组成部分



1. 词项列表构建

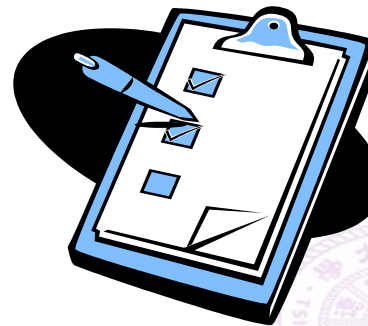
• 1.4 词项列表的规模

- Heaps' Law: 词项列表的规模与语料规模共同增长
- Zipf's Law: 大规模语料中, 数目较少的高频词占大部分语料内容, 但低频词占有词项数目的绝大部分
 - Brown Corpus: 包含1,014,312词, 其中135个词覆盖语料的 50%, 2000个词覆盖语料的80%。
 - 低频词数目众多, 存储空间和编码空间的浪费
- 有损优化: 不必将所有词项加入内存中的词项列表, (在内存中)忽略最不常用的词项



本节概要

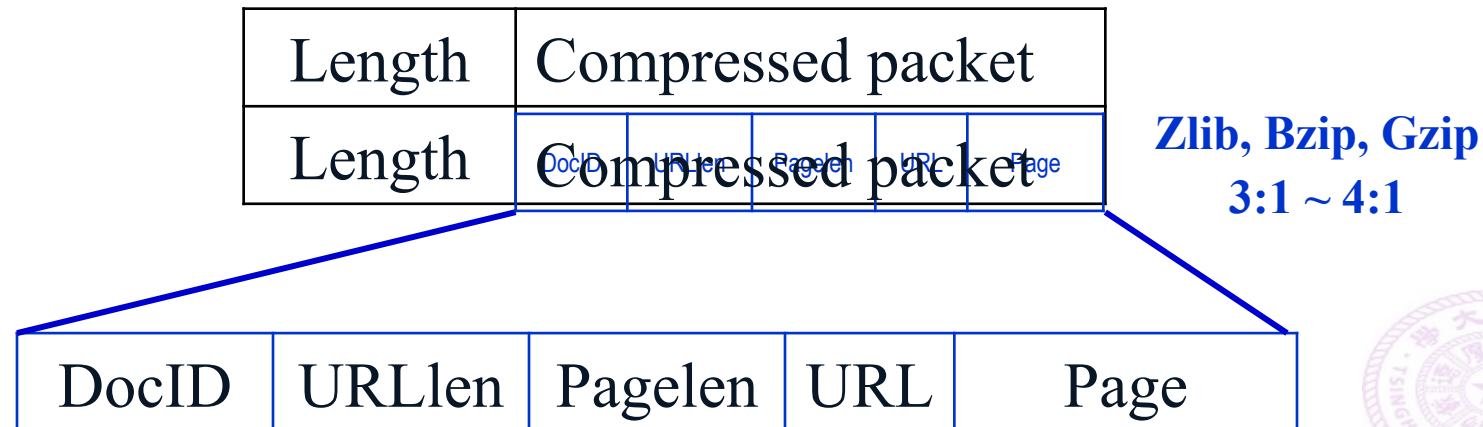
1. 词项列表构建
2. 索引数据结构
 - 2.1 原始数据存放
 - 2.2 词项出现记录
 - 2.3 倒排/正排索引
3. 索引建立过程
4. 索引查询过程



2. 索引数据结构

• 2.1 原始数据存放

- 目的：保证具备重建索引的能力
 - 硬件崩溃时有发生，累计式抓取周期过长
 - Archive the Web: WebInfoMall
- 存储结构与压缩



2. 索引数据结构

• 2.2 词项出现信息记录(Hit record, token, entry)

- 索引存储的基本单位 (正排/倒排)
- 词项在该文档中某次特定出现的信息(2 bytes)

是否大写(1 bit)	字体(3 bits)	位置(12 bits, 4096)
0	000	0000 0000 0010
0	000	0000 0000 0101
0	010	0000 0001 0001

/当/网络/经济/在/此次/经济/危机/中/再度/被/抛向/空中/
时/，/众多/的/互联网/企业/都/在/思考/如何/“过冬”/。/对
于/企业/来讲/，/网站/推广/和/市场营销/的/作用/日益/凸显/。
/经济/危机/正/促使/整个/行业/发生/着/变化/。



2. 索引数据结构

• 2.2 词项出现信息记录

111为保留位

是否大写(1 bit)	字体(3 bits)	位置(12 bits, 4096)
-------------	------------	-------------------

HTML规范中，规定了页面中的六级标题格式：

<html>

<h1>This is the main header</h1>

Some initial text

<h2>This is a level 2 header</h2>

Paragraph and sentences.

<h3>This is a level 3 header</h3>

.....

</html>

This is the main header

Some initial text

This is a level 2 header

Paragraph and sentences. Paragraph and sentences.
Paragraph and sentences. Paragraph and sentences.

This is a level 3 header

Paragraph and sentences. Paragraph and sentences.
Paragraph and sentences. Paragraph and sentences.
Paragraph and sentences. Paragraph and sentences.
Paragraph and sentences. Paragraph and sentences.

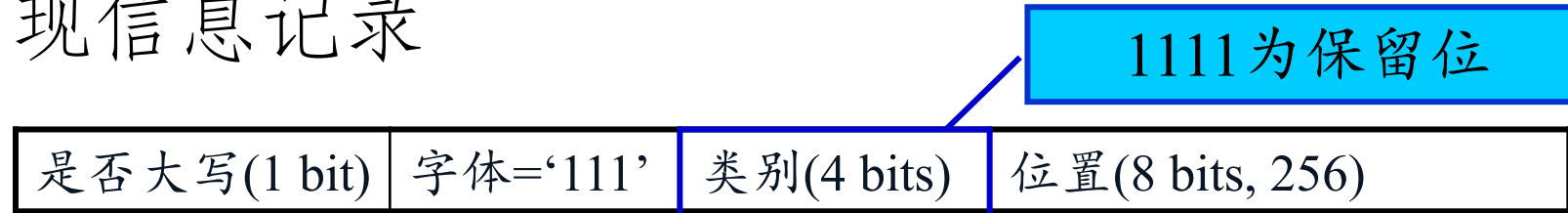
This is a level 3 header

Paragraph and sentences. Paragraph and sentences.
Paragraph and sentences. Paragraph and sentences.
Paragraph and sentences. Paragraph and sentences.
Paragraph and sentences. Paragraph and sentences.



2. 索引数据结构

• 2.2 词项出现信息记录



<HTML> (HTML特殊域)

<HEAD>

<TITLE>...</TITLE>

<Meta>

</HEAD>

<BODY>

...

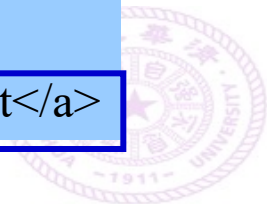
</BODY>

</HTML>

<meta name="keyword" contents="key1, key2">
<meta name="description" contents="description....">

粗体
<i>斜体</i>
<u>加下划线</u>
红色的字
.....

anchor text

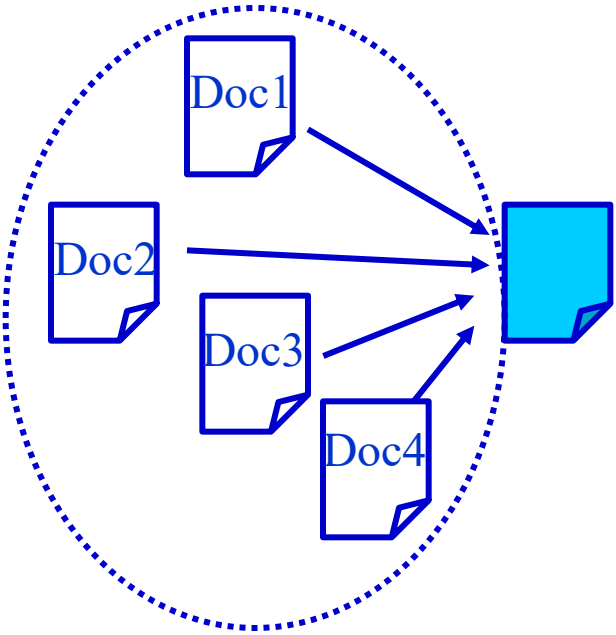


2. 索引数据结构

•2.2 词项出现信息记录

文档编号的Hash值

是否大写(1 bit)	字体='111'	类别='1111'	编号(4 bits)	位置(4 bits)
-------------	----------	-----------	------------	------------



(锚文本) (锚文本来源)

Doc 1: 清华/大学/主页

0	111	1111	0000	0000
---	-----	------	------	------

Doc 2: 清华/主页

0	111	1111	0001	0000
---	-----	------	------	------

Doc 3: 世纪/清华

0	111	1111	0010	0001
---	-----	------	------	------

Doc 4: 清华/我/爱/清华

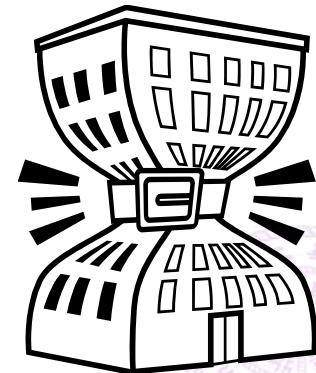
0	111	1111	0011	0000
0	111	1111	0011	0011



2. 索引数据结构

• 2.2 词项出现信息记录

- 需要记录的基本信息：位置、字体、类别
- 方案1：simple encoding (3 integers)
- 方案2：simple encoding + zip
- 方案3：hand optimized allocation of bits
- 优势：节约每一个比特
- 问题：有损优化，可能的位置信息丢失，可能的锚文本来源丢失



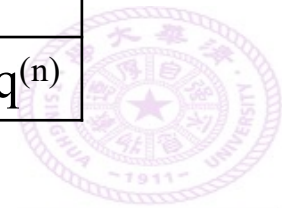
2. 索引数据结构

• 2.3 倒排索引与正排索引

- 均以词项出现信息记录(HIT record)为最小单位

Term 1	Doc 1, pos 1	Doc 1, pos 2	...	Doc p, pos q
Term 2	Doc 1', pos 1'	Doc 1', pos 2'	...	Doc p', pos q'
...				
Term N	Doc 1 ⁽ⁿ⁾ , pos 1 ⁽ⁿ⁾	Doc 1 ⁽ⁿ⁾ , pos 2 ⁽ⁿ⁾	...	Doc p ⁽ⁿ⁾ , pos q ⁽ⁿ⁾

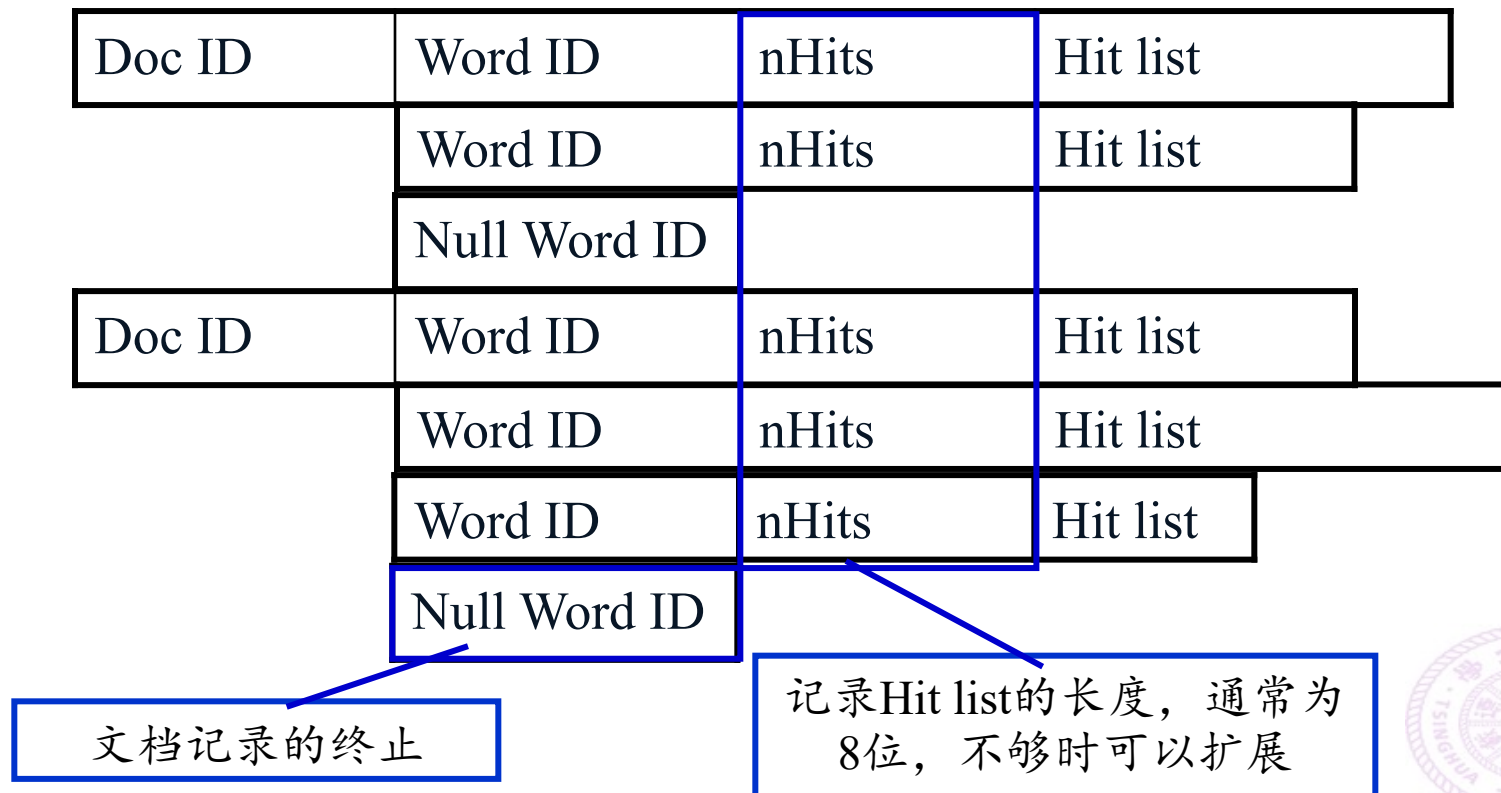
Doc 1	Term 1, pos 1	Term 1, pos 2	...	Term p, pos q
Doc 2	Term 1', pos 1'	Term 1', pos 2'	...	Term p', pos q'
...				
Doc N	Term 1 ⁽ⁿ⁾ , pos 1 ⁽ⁿ⁾	Term 1 ⁽ⁿ⁾ , pos 2 ⁽ⁿ⁾	...	Term p ⁽ⁿ⁾ , pos q ⁽ⁿ⁾



2. 索引数据结构

• 2.3 倒排索引与正排索引：正排索引

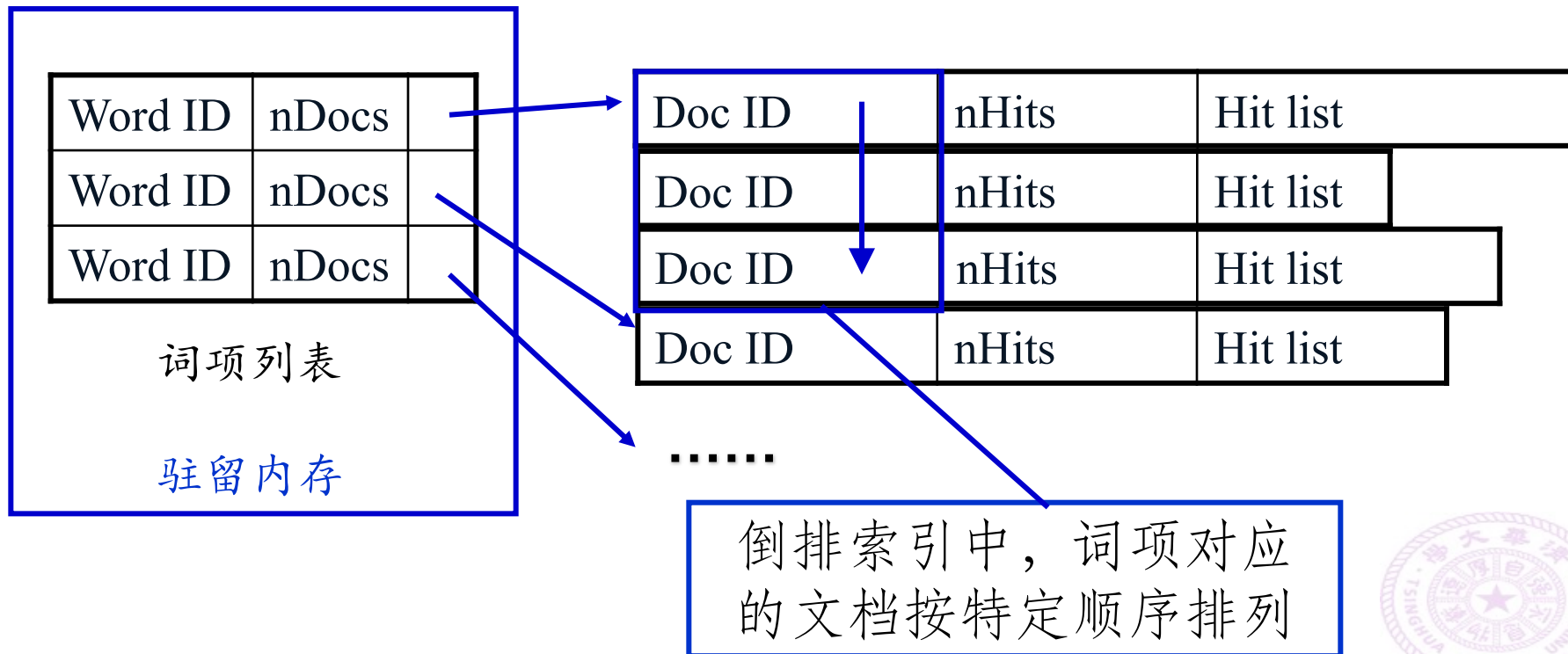
- 输入Doc ID，输出WordID列表及对应的Hit lists



2. 索引数据结构

• 2.3 倒排索引与正排索引：倒排索引

- 输入 WordID，输出 DocID 列表及对应的 Hit lists



2. 索引数据结构

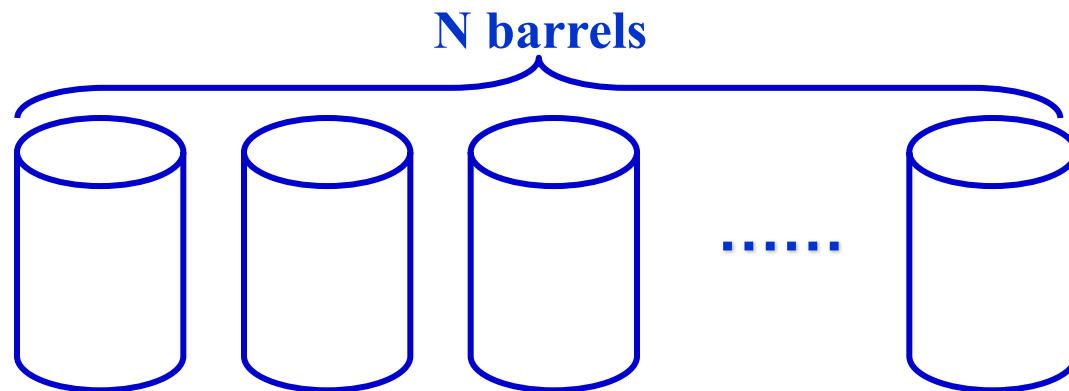
• 2.4 索引的并行存储结构

- 在有限的磁盘存储量限制下增加存储总量
- 在有限的磁盘I/O性能限制下增加索引吞吐效率
- 并行存储单元
 - 物理或虚拟存储单位：
“桶” (barrel) / “环” (circle) / “碎片” (shard)
- 如何划分存储单元
 - 按文档划分
 - 按词项划分



2. 索引数据结构

- 2.4 索引的并行存储结构：按词项划分
 - 每个桶包含 $1/N$ 的词项，以及这部分词项对应的所有文档
 - 倒排索引：直接访问对应 **WordID** 的桶
 - 正排索引：访问所有桶，获得对应 **DocID** 的词项列表(不完整)和 **Hit lists**



2. 索引数据结构

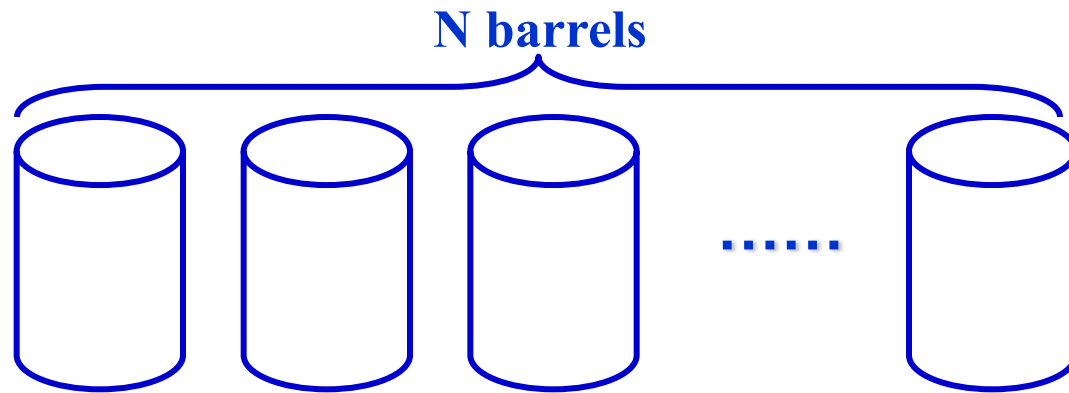
•2.4 索引的并行存储结构：按词项划分

- 优点：对包含K个词项的查询，至多需要K个桶共同进行处理，只需进行O(K)次磁盘查找操作
- 缺点：
 - 网络传输量大(需要传输整个词项对应的倒排表)
 - 需要在索引结构之外存储文档的其他信息
 - 扩展性较差：更新/增加文档

[illegible]

2. 索引数据结构

- 2.4 索引的并行存储结构：按文档划分
 - 每个桶包含 $1/N$ 的文档，以及这部分文档包含的所有词项
 - 正排索引：直接访问对应 **DocID** 的桶
 - 倒排索引：访问所有桶，获得对应 **WordID** 的文档列表(不完整)和 **Hit lists**



2. 索引数据结构

• 2.4 索引的并行存储结构：按文档划分

- 优点：可扩展性较好

- 每个桶可以独立处理查询请求
- 每个桶可以保存文档的其他信息(如PageRank)
- 网络传输量小(只需传输查询需求和每个桶的查询结果)

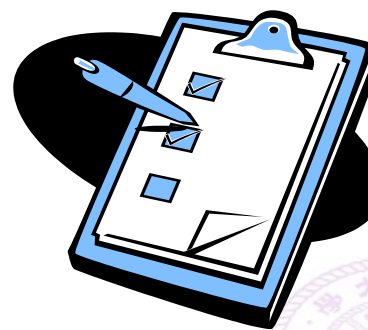
- 缺点：

- 需要等待**所有桶**处理完查询请求，以避免重要文档丢失
- 处理包含K个词项的查询，需进行 $O(K*N)$ 次查找操作
- 可能的解决方式：创建多个重要文档的镜像



本节概要

1. 词项列表构建
2. 索引数据结构
3. 索引建立过程
4. 索引查询过程



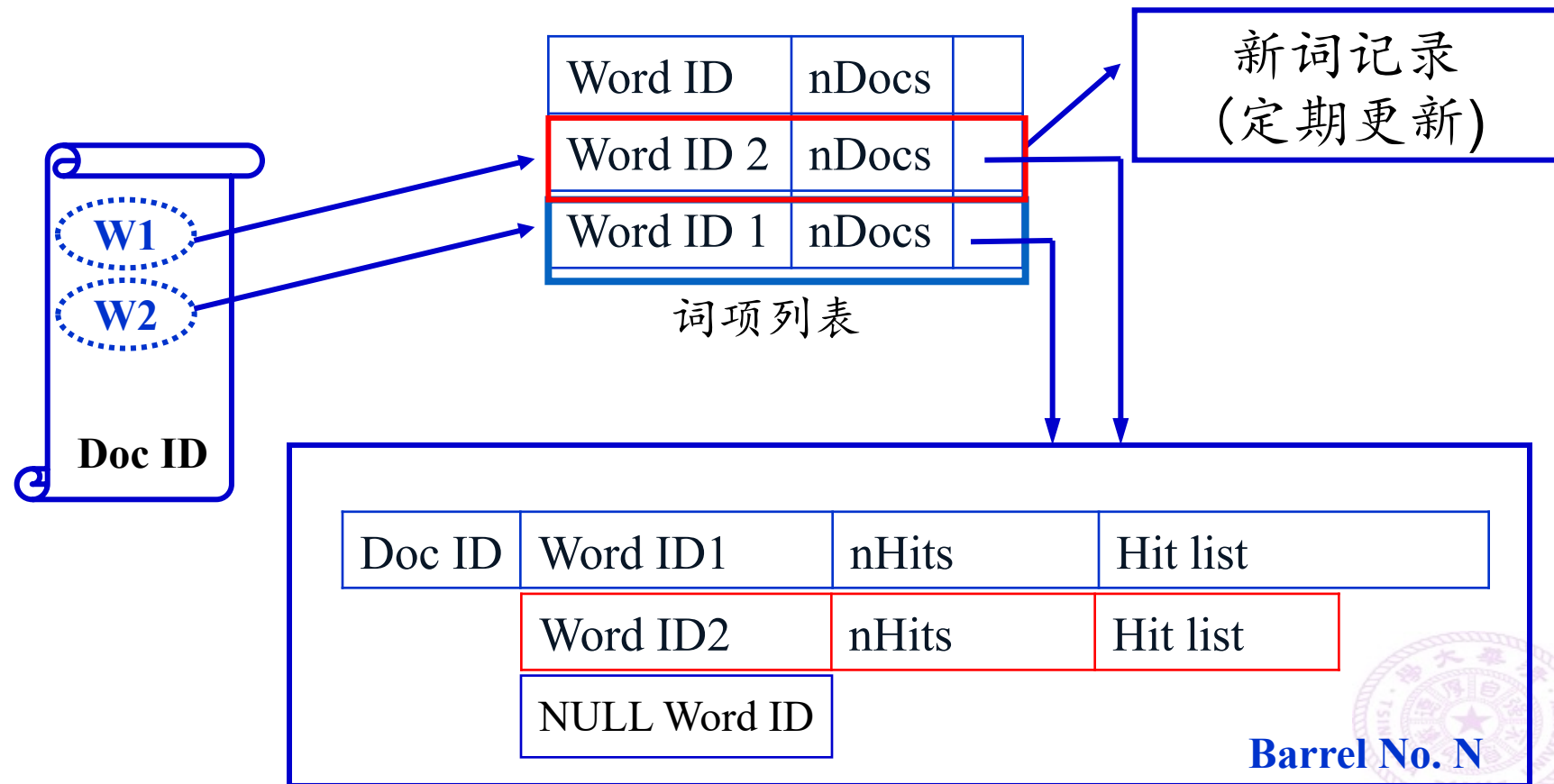
3. 索引建立过程

•3.1 文档预处理



3. 索引建立过程

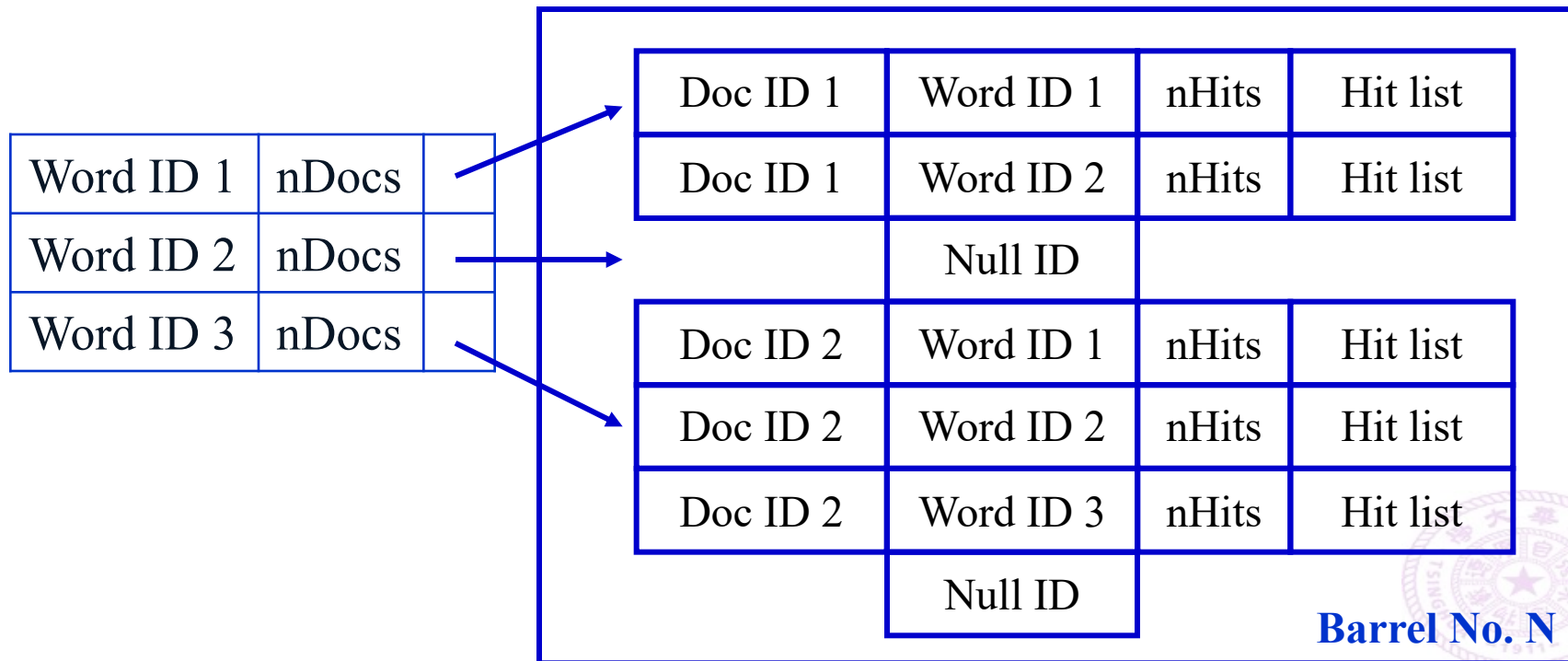
- 3.2 构建正排索引：以按文档分桶为例



3. 索引建立过程

• 3.3 构建倒排索引

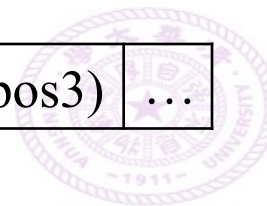
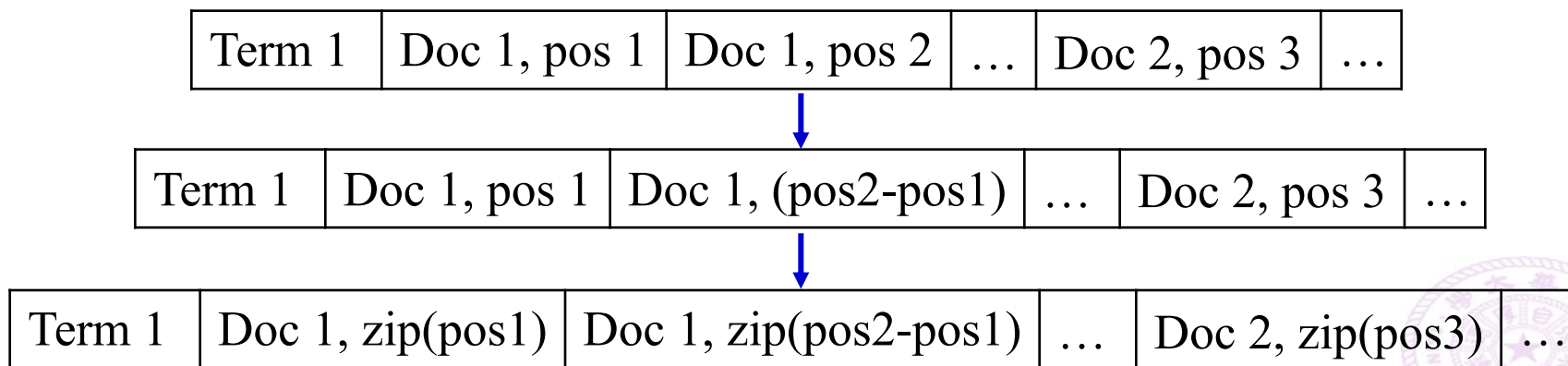
- 在每个桶内，对正排索引按**Word ID**进行排序，重构成倒排索引



3. 索引建立过程

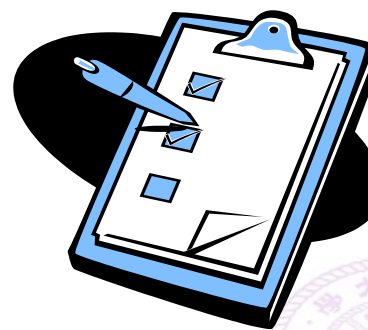
•3.4 性能优化：索引压缩

- 优点：节约空间，有效利用I/O，增加硬盘寿命
- 缺点：额外占用计算资源，传统的压缩算法可能导致索引随机读写的困难
- 方案：主要针对位置信息进行压缩



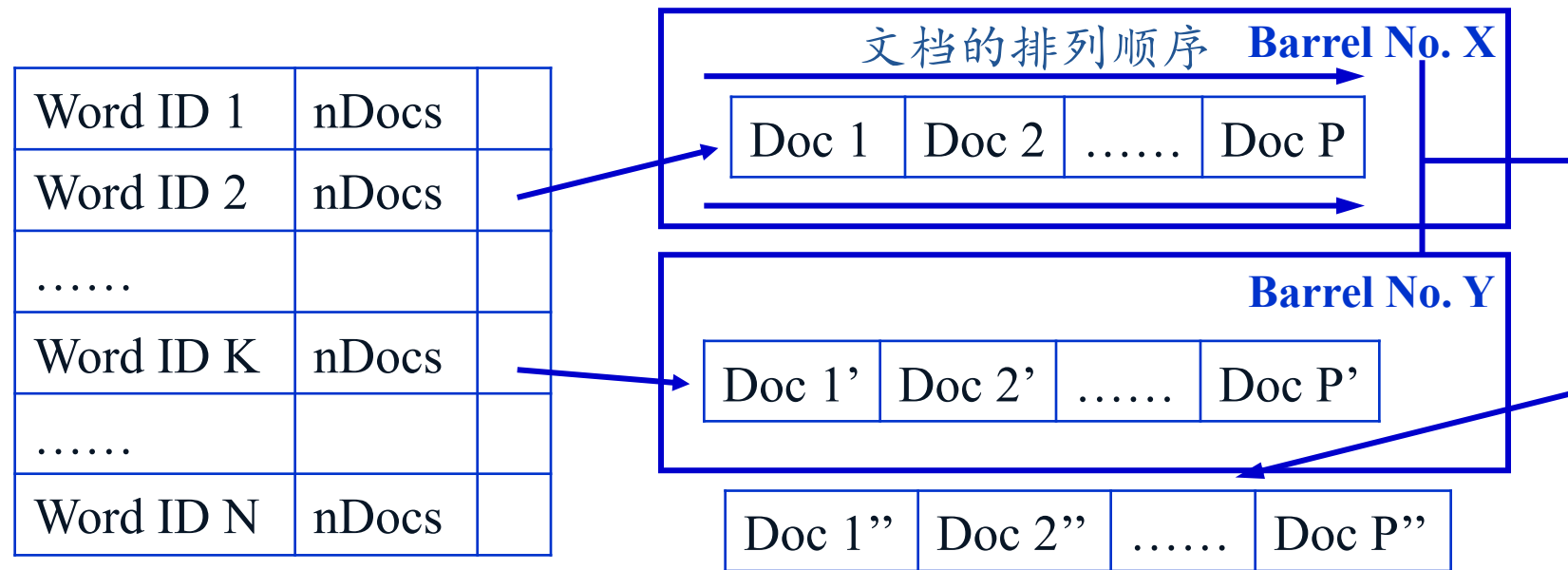
本节概要

1. 词项列表构建
2. 索引数据结构
3. 索引建立过程
4. 索引查询过程



4. 索引查询过程

- 4.1 查询过程：以按词项分桶为例
 - 用户输入：(Word ID 2, Word ID K)



- 系统输出：包含以上查询词的(部分)文档列表



4. 索引查询过程

- 4.2 性能优化：缓存服务器
 - 保存常用索引项
 - 保存常用同现词项的文档列表求交结果
 - Google: 缓存命中率30-60%,
 - 本质：时间局部性原理
 - 容易被缓存命中的查询词，经常是热门的，同时也是资源消耗量巨大的（网络资源数量多）



4. 索引查询过程

•4.2 性能优化：关键位置倒排索引

- 关键字
- 对于
- 在HT
- 更为



The screenshot shows a Google search interface with the query 'index'. The search results indicate approximately 8,470,000,000 results found in 0.13 seconds. The left sidebar includes navigation options like '所有结果', '图片', '视频', '新闻', and '更多'. The main content area displays a list of search results, including a link to 'INDEX: Design to Improve Life' and a Wikipedia entry for 'Index'. The bottom of the page shows a list of search results from Baidu, including links to 'index.baidu.com/' and 'indexaward.dk/'.

Google

index

Google 搜索

找到约 8,470,000,000 条结果 (用时 0.13 秒)

高级搜索

小提示: [只搜索中文\(简体\)结果](#), 可在 [设置](#) 指定搜索语言

相关搜索: [index.dat](#) [index是什么](#) [oracle index](#)

[INDEX: Design to Improve Life](#) - [翻译此页]

INDEX: works globally to promote and apply both design and design processes that have the capacity to improve the lives of people worldwide.

[Index-award - About-index - Contact - People](#)

[www.indexaward.dk/](#) - 网页快照 - 类似结果

[Index - Wikipedia, the free encyclopedia](#) - [翻译此页]

An **index** is a system used to make finding information easier. **Index** may also refer to: Bibliographic **index**, a regularly updated print periodical publication ...

[en.wikipedia.org/wiki/Index](#) - 网页快照 - 类似结果

[百度指数](#)

1, 2012, 120597. 2, 新少林寺, 76232. 3, 3d肉蒲团, 63983. 4, 日本沉没, 52102. 5, 神奇侠侣, 42152. 完整榜单. 电视剧. 1, 回家的诱惑, 1294811. 2, 回家的欲望 ...

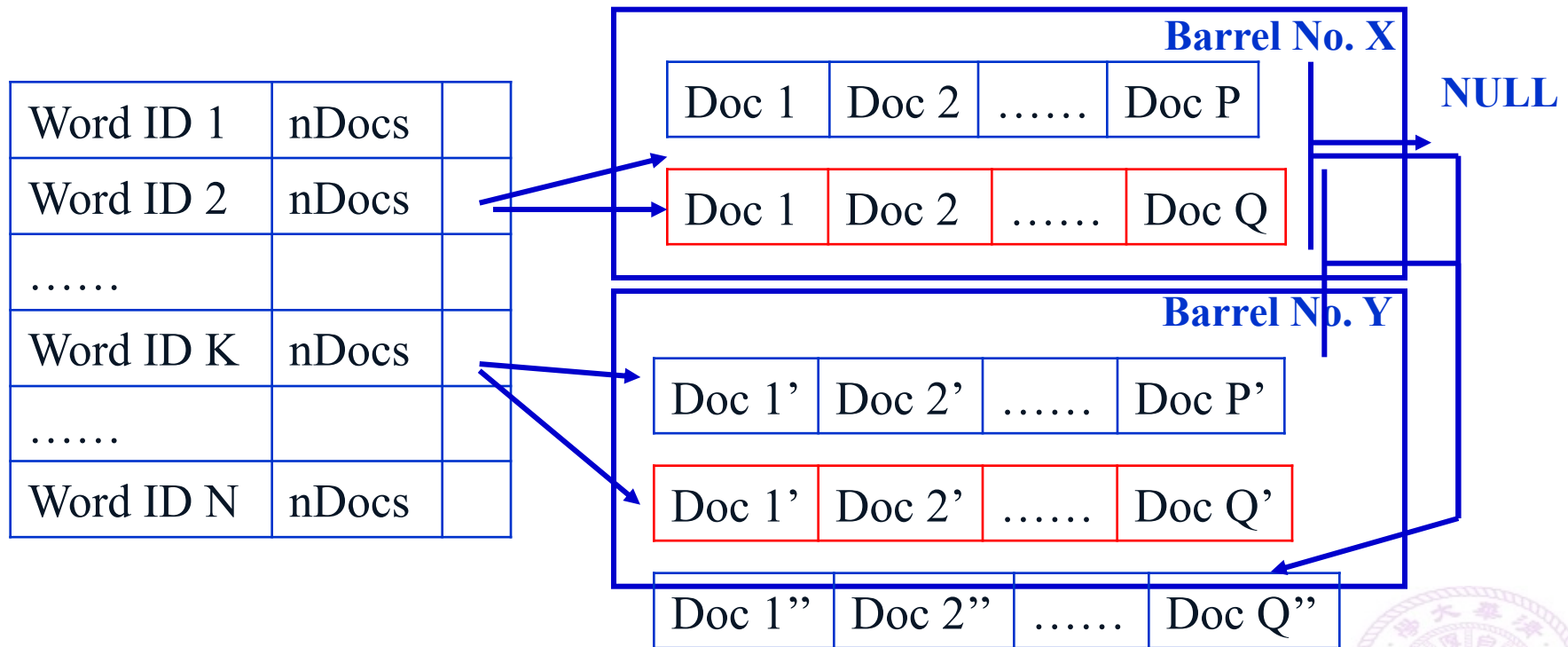
[index.baidu.com/](#) - 网页快照 - 类似结果

的文档



4. 索引查询过程

• 4.2 性能优化：关键位置倒排索引



总结

- 词项列表构建
 - 分词算法/词项列表构建
- 索引数据结构
 - 原始数据存放/词项出现信息记录/
正排索引与倒排索引/并行索引结构
- 索引系统运行方式
 - 索引建立/索引使用/性能优化



