

# 第二章信息检索模型



林 鸿 飞

# 提纲

- **Web**搜索与信息检索
- 布尔模型以及扩展布尔模型
- 向量空间模型以及潜在语义模型
- 概率模型以及基于语言模型的检索模型
- 知识模型以及基于本体论信息检索

# 提纲

- **Web搜索与信息检索**
- 布尔模型以及扩展布尔模型
- 向量空间模型以及潜在语义模型
- 概率模型以及基于语言模型的检索模型
- 知识模型以及基于本体论信息检索

# Web搜索与信息检索

- 1989年， Tim Berners-Lee在日内瓦欧洲离子物理研究所开发计算机远程控制时**首次**提出了Web概念，并在1990年圣诞节前推出了第一个浏览器。
- 随后，他又设计出HTTP、URL和HTML的规范，使网络能够为普通大众所应用。
- Ted Nelson 在1965年提出了超文本的概念
  - 超文本传输协议(HTTP, HyperText Transfer Protocol)是互联网上应用最为广泛的一种网络传输协议
  - 超文本标注语言 (HTML)

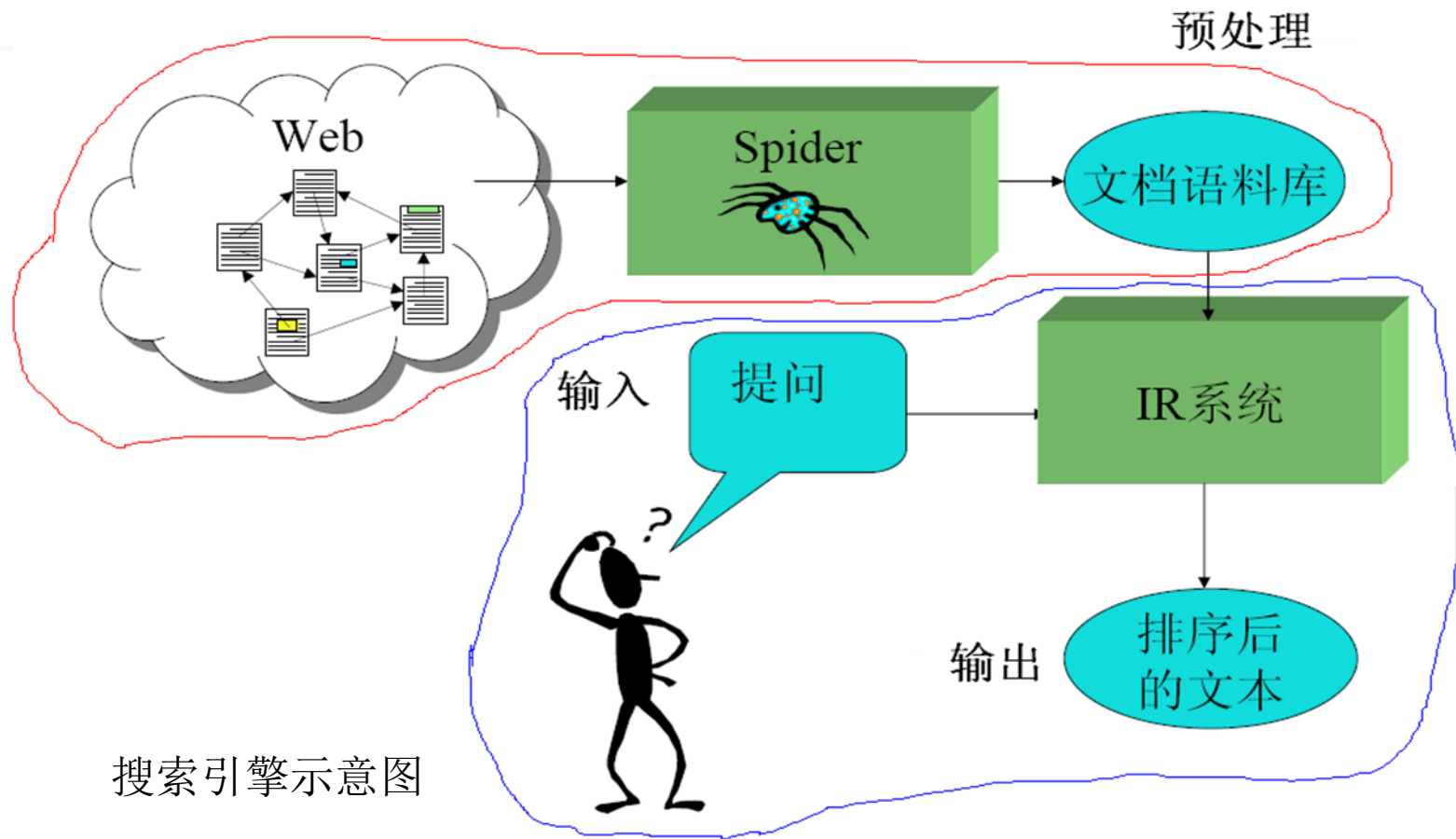
# Web搜索与信息检索

- 1993, Web robots 用于收集 URL, 例如: Wanderer、WWW Worm
- 1994, Stanford 博士生 David Filo and Jerry Yang 开发手工划分主题层次的雅虎网站.Yahoo.com
- 1994, WebCrawler是第一个全文搜索引擎
- 1995, Lycos (来自CMU) 还有Infoseek是搜索引擎史上又一个重要的进步。
- 1995, DEC的AltaVista是第一个支持自然语言搜索的搜索引擎, AltaVista是第一个实现高级搜索语法的搜索引擎 (如AND, OR, NOT等)
- 1997, 北大天网 由北大计算机系网络与分布式系统研究室开发, 于1997年10月29日正式在CERNET上提供服务
- 1998, Google的核心技术PageRank, Google公司则把1998年9月27日认作自己的生日
- 2000, 超链分析专利发明人、前Infoseek资深工程师李彦宏创立了百度公司, 2001年10月22日正式发布Baidu搜索引擎

# Web搜索与信息检索

- 史前时代：分类目录时代  
代表：Yahoo和Hao123
- 第一代：文本检索时代  
代表：AltaVista（布尔、向量空间、概率模型）
- 第二代：链接分析时代  
代表：Google和Baidu（PageRank、链接分析）
- 第三代：用户为中心的时代（知识图谱、意图检测、行为分析）  
正在方兴未艾。。。。。（2.5G）

# Web搜索与信息检索



搜索引擎示意图

# Web搜索与信息检索

## 搜索引擎:

能够接收用户通过浏览器提交的**查询词q**,在一个可以**接受的时间内**返回一个和用户查询**匹配**的网页信息**列表L**,其中L中每一条至少包括三个元素 (**标题, 网址, 摘要**)

[百度一下](#)[网页](#)[新闻](#)[贴吧](#)[知道](#)[音乐](#)[图片](#)[视频](#)[地图](#)[文库](#)[更多»](#)

百度为您找到相关结果约12,300,000个

搜索工具

[信息检索](#) [百度百科](#)



**信息检索** (Information Retrieval) 是用户进行**信息**查询和获取的主要方式, 是查找**信息**的方法和手段。狭义的**信息检索**仅指**信息**查询 (Information Search)。即用户根据需要, 采用...

[起源](#) [定义](#) [类型](#) [主要环节](#) [热点](#) [检索原因](#) [四个要素](#) [更多>>](#)

[查看“信息检索”全部6个含义>>](#)

[baike.baidu.com/](http://baike.baidu.com/) ▼ - ❏



# Web搜索与信息检索

- 收集：
  - 将网页爬取下来（爬虫，蜘蛛，机器人）
- 预处理：
  - 网页去重，去噪，正文提取，分词等
  - 建立索引
- 服务：
  - 接受用户请求，检索词串的处理，查询重构
  - 找到满足要求的列表
  - 根据连接和文本中的词进行排序输出

# Web搜索与信息检索

- Web搜索引擎系统
  - Web数据采集系统
  - 网页预处理和索引系统
  - 检索系统（IR检索模型）
  - 检索结果排序系统

# Web搜索与信息检索



# Web搜索与信息检索

## ■ 网页去重

- 网络上出现的多个域名对应同一网站的情况或者网站的互相转载、去除重复的网页是为了避免同一个网站的内容被多次采集和索引

## ■ 网页正文提取

- 由于网页中有很多对建立索引无用的信息，比如广告信息，一些无用的连接信息，还有一些脚本语言
- 所以在建立索引之前，需要先清理一下垃圾信息，这个过程被称为正文提取

## ■ 网页的索引（分词/停用词处理、倒排索引）

# Web搜索与信息检索

搜人搜物搜信息  
重情重义重认知

大连理工大学  
信息检索研究室



[首页](#) [学术研究](#) [成员介绍](#) [新闻动态](#) [科研项目](#) [学术报告](#) [多彩生活](#) [缤纷相册](#) [学术评测](#)

>>最受欢迎的情感词典, 欢迎点击下载! <<

### 研究方向

- ✦ 搜索引擎与自然语言处理
- ✦ 文本挖掘与机器学习
- ✦ 情感分析与观点挖掘
- ✦ 面向生物医学领域的文本挖掘

### 学术报告

- ✦ 06-05 罗凌 神经网络结构在命名
- ✦ 06-02 徐博 在CQA检索中建模未匹

## 欢迎来到信息检索研究室

正文部分

我们专注于互联网上内容的搜索、分析、理解和诠释, 挖掘出潜在的、有价值的、新颖的知识模式, 创造人机和谐的网络环境。我们的研究方向是信息检索、自然语言处理、推荐系统、社会计算、情感计算、面向生物医学领域的文本挖掘等。信息检索技术涉及到自然语言处理、机器学习、认知科学等诸多理论和技术, 是一个富有朝气和希望的研究领域。

信息检索研究室(DUTIR)在林鸿飞教授(微博)领导下, 坚持理论研究和实际应用相结合, 与国外大学和研究机构保持良好的合作关系。营造宽松和谐的研究环境, 悉心培养信息检索领域的优秀人才。鼓励学生积极参与各项学术活动, 同时举办丰富多彩的文体活动, 让学生受到多方面的熏陶。

**互联网上烽烟渐浓, 鏖战正急。欢迎各位青年才俊, 加入我们阵营, 创出一片天地!**

# Web搜索与信息检索

- 排序重要性
  - 65%—70%的网民点击搜索结果的第一页。
  - 20%-25%的网民点击搜索结果第二页
  - 3%-4%的网民点击量其他的网页
- 排序算法为各公司的机密
  - Google 拥有核心技术为PageRank技术
  - 百度拥有核心技术为超链分析
- SEO（搜索引擎优化）
  - 互相博弈



# Web搜索与信息检索

- 在**预处理阶段**，网页的重要性排序其依据只有链接因素。参照文献计量学的思想：

被引用多的网页重要

被重要的网页引用的网页重要

- 而在**服务阶段**，输出的网页按照与用户查询的相关程度排序，更多与内容相关。一般每个用户只会浏览前两页的搜索结果。

# 提纲

- Web搜索与信息检索
- 布尔模型以及扩展布尔模型
- 向量空间模型以及潜在语义模型
- 概率模型以及基于语言模型的检索模型
- 知识模型以及基于本体论信息检索



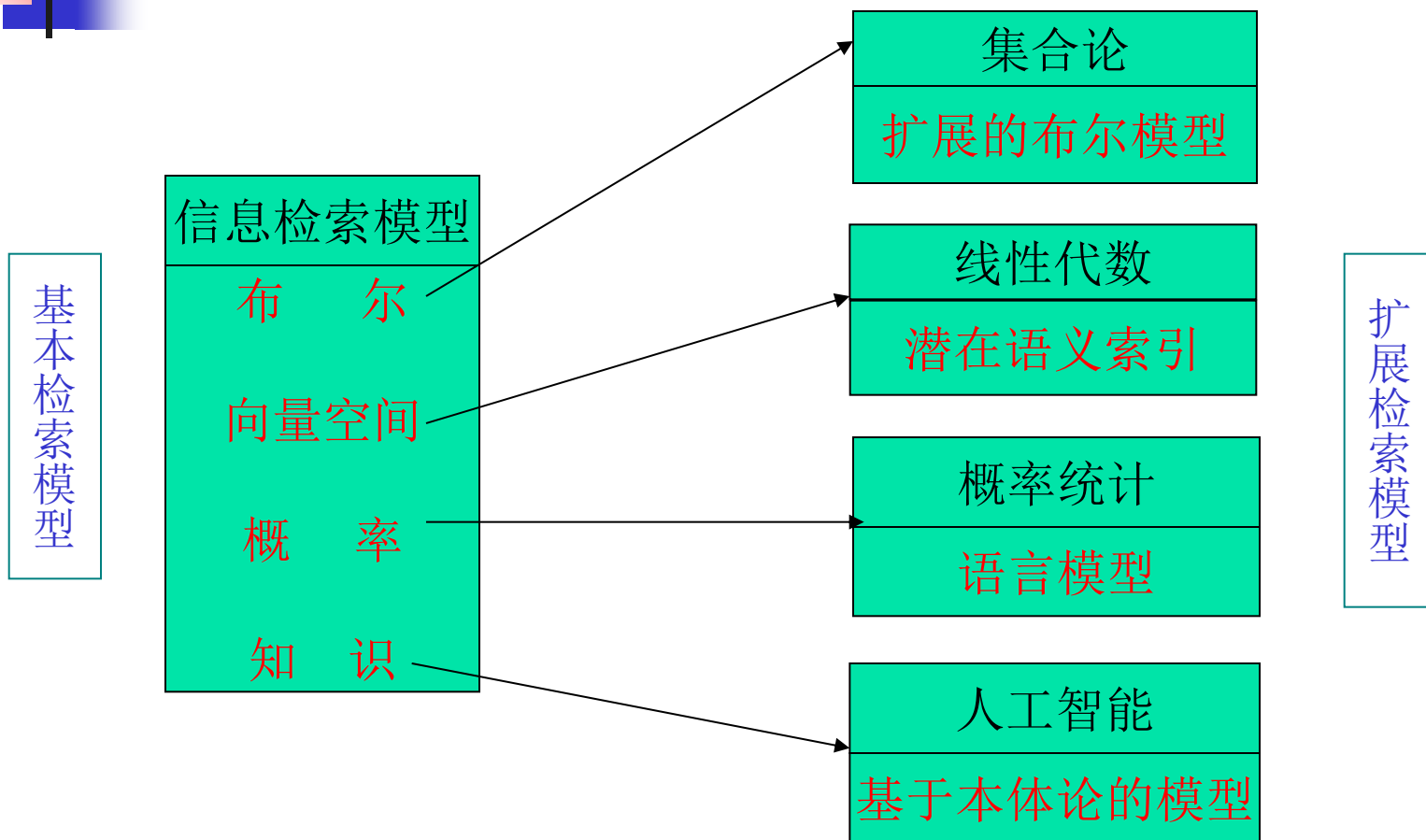
# 什么是模型？

- 模型是采用数学工具，对现实世界某种事物或某种运动的抽象描述
- 面对相同的输入，模型的输出应能够无限地逼近现实世界的输出
  - 举例：天气的预测模型
- 信息检索模型
  - 是表示文档，用户查询以及查询与文档的关系的框架

# 信息检索模型

- 信息检索模型是一个四元组 $[D, Q, F, R(q_i, d_j)]$ 
  - D: 文档集的机内表示
  - Q: 用户需求的机内表示
  - F: 文档表示、查询表示和它们之间的关系的模型框架(Frame)
  - $R(q_i, d_j)$ : 排序函数, 给query  $q_i$  和document  $d_j$ 评分
- 信息检索模型取决于:
  - 从什么样的视角去看待查询和文档
  - 基于什么样的理论去看待查询和文档的关系
  - 如何计算查询和文档之间的相似度

# 检索模型分类



# 布尔模型(Boolean Model)

# 布尔模型

- 最早的IR模型，也是应用最广泛的模型
- 目前仍然应用于商业系统中
- Lucene是基于布尔（Boolean）模型的

# 布尔模型描述

- 文档D表示
  - 一个文档被表示为特征项的集合
- 查询Q表示
  - 查询(Queries)被表示为特征项的布尔组合，用“与、或、非”连接起来，并用括弧指示优先次序
- 匹配F
  - 一个文档当且仅当它能够满足布尔查询时，才将其检索出来
  - 检索策略基于二值判定标准
- 算法R
  - 根据匹配框架F判定相关

# 查询表示

- 在布尔模型中，所有特征项的权值变量和文档 $d_j$ 与查询 $q$ 的相关度都是二值的
- 查询 $q$ 被表述成一个常规的布尔表达式，为方便计算查询 $q$ 和文档 $d$ 的相关度，一般将查询 $q$ 的布尔表达式转换成析取范式 $q_{DNF}$

## 示例

- 文档集包含两个文档：  
文档1: a b c f g h  
文档2: a f b x y z  
查询: 文档中出现a或者b, 但必须出现z。
- 将查询表示为布尔表达式  $q = (a \vee b) \wedge z$
- 并转换成析取范式  $q_{DNF} = (1, 0, 1) \vee (0, 1, 1) \vee (1, 1, 1)$
- 文档1和文档2的三元组对应值分别为(1,1,0)和(1,1,1)
- 经过匹配, 将文档2返回



# 优点

- 布尔模型是最常用的检索模型：
  - 由于查询简单，因此容易理解
  - 通过使用复杂的布尔表达式，可以很方便地控制查询结果
- 相当有效的实现方法
  - 相当于识别包含了一个某个特定term的文档
- 经过某种训练的用户可以容易地写出布尔查询
- 布尔模型可以通过扩展来包含排序的功能，即“扩展的布尔模型”

# 问题

- 布尔模型被认为是功能最弱的方式，其主要问题在于不支持部分匹配，而完全匹配会导致太多或者太少的结果文档被返回
  - 非常刚性：“与”意味着全部；“或”意味着任何一个
- 很难控制被检索的文档数量
  - 原则上讲，所有被匹配的文档都将被返回
- 很难对输出进行排序
  - 不考虑索引词的权重，所有文档都以相同的方式和查询相匹配
- 很难进行自动的相关反馈
  - 如果一篇文档被用户确认为相关或者不相关，怎样相应地修改查询呢？

## 2 扩展的布尔模型

# 传统布尔模型和向量空间模型的优缺点

模 型 \ 优缺点	优 点	缺 点
传统布尔模型	检索式的结构化—用布尔算法明确的揭示了特征项之间的关系。	不能按相似度进行排序； 不能控制返回文档的数量； 不能进行相关性反馈。
向量空间模型	<ol style="list-style-type: none"> <li>(1)检索结果的相关性排序；</li> <li>(2)可以控制输出结果的数量；</li> <li>(3)能够进行相关性反馈。</li> </ol>	认为特征项相互独立，未能揭示词语之间的关系。

# 基本模糊集合模型

## ■ 布尔检索的检索式与文档相似度计算公式

布尔检索式	赋值公式
$sim(d_j, t_m \text{ and } t_n)$	$\min\{w_{m,j}, w_{n,j}\}$ $w_{m,j} \bullet w_{n,j}$
$sim(d_j, t_m \text{ or } t_n)$	$\max\{w_{m,j}, w_{n,j}\}$ $w_{m,j} + w_{n,j} - w_{m,j} \bullet w_{n,j}$
$sim(d_j, t_m \text{ andnot } t_n)$	$w_{m,j} \bullet (1 - w_{n,j})$

## 举例

- 假设有一查询  $q = (t_1 \text{ or } t_2) \text{ andnot } t_3$  , 文档为d, 其中  $w_{1,d}=0.7, w_{2,d}=0.2, w_{3,d}=0.1$ , 计算文档d与q的相似度:

- 内层检索子式:

$$\text{sim}(d, t_1 \text{ or } t_2) = 0.7$$

- 外层检索式

$$\text{sim}(d, (t_1 \text{ or } t_2) \text{ andnot } t_3) = \text{sim}(d, t_1 \text{ or } t_2) \times (1 - \text{sim}(d, t_3)) = 0.7 \times (1 - 0.1) = 0.63$$

- 最后得到文档d与查询q的相似度为0.63

# 分析

- 该模型保留了传统布尔检索模型的结构化特点，能够对检索结果按相似度进行降序排列
- 能够控制输出结果的数量
- 没有对查询中的特征项赋予权值，而是直接认为查询中的特征项权值和文档中的特征项权值相等

# 扩展模糊集合模型

- 在基本模糊集合模型中，没有对查询中的特征项赋予权值，而是假设查询中的特征项权值和文档中的特征项权值相等。
- 针对这个问题，扩展模糊集合模型进行了改进，给文档中的词和检索式中的词赋予不同的权值。
- 一种有效的方式是将查询特征项的权值与文档特征项的权值相乘



## 举例

- 给定文档 $d=\{t_1, t_2, t_3, t_4, t_5\}$ , 查询 $q=((t_1 \text{ and } t_2) \text{ or } t_3) \text{ and } (t_4 \text{ or } t_5)$
- 查询特征项和文档特征项的权重

文档特征项权值	查询特征项权值	计算得到的相似度值
$w_{1,d} = 0.6$	$w_{1,q} = 0.5$	$\text{sim}(d, t_1) = 0.3$
$w_{2,d} = 0.3$	$w_{2,q} = 0.9$	$\text{sim}(d, t_2) = 0.27$
$w_{3,d} = 0.7$	$w_{3,q} = 0.2$	$\text{sim}(d, t_3) = 0.14$
$w_{4,d} = 0.8$	$w_{4,q} = 0.6$	$\text{sim}(d, t_4) = 0.48$
$w_{5,d} = 0.9$	$w_{5,q} = 0.4$	$\text{sim}(d, t_5) = 0.36$

- 假设查询特征项和文档特征项的权重如表所示, 这个扩展只是改变了权值, 相似度的计算方法并没有改变

## 举例（续）

- 文档 $d$ 与各个层次查询的相似度分别如下：

$$\text{sim}(d, q_3) = \text{sim}(d, t_1 \text{ and } t_2) = \text{sim}(d, t_1) \times \text{sim}(d, t_2) = 0.3 \times 0.27 = 0.081$$

$$\text{sim}(d, q_1) = \text{sim}(d, (q_3 \text{ or } t_3)) = \text{sim}(d, q_3) + \text{sim}(d, t_3) - \text{sim}(d, q_3) \times \text{sim}(d, t_3) = 0.21$$

$$\text{sim}(d, q_2) = \text{sim}(d, t_4 \text{ or } t_5) = \text{sim}(d, t_4) + \text{sim}(d, t_5) - \text{sim}(d, t_4) \times \text{sim}(d, t_5) = 0.67$$

$$\text{sim}(d, q) = \text{sim}(d, q_1 \text{ and } q_2) = \text{sim}(d, q_1) \times \text{sim}(d, q_2) = 0.21 \times 0.67 = 0.14$$

# 分析

- 通过上述的例子，可以清楚地看到采用扩展的模糊集合计算相似度的方法。
- 同时，也要注意利用上述方法计算查询与文档的相似度时，有两点不足：
  - 每篇文档与查询相似度的大小依赖于查询的长度和乘数操作的次数，相似度数值的稳定性；
  - 文档与查询相似度的数值大小有时受**and**子查询中一些不重要的词（也即权值很小的词）控制。

# 提纲

- Web搜索与信息检索
- 布尔模型以及扩展布尔模型
- 向量空间模型以及潜在语义模型
- 概率模型以及基于语言模型的检索模型
- 知识模型以及基于本体论信息检索

# 向量空间模型 (VSM)

# 模型的提出

- Salton在上世纪60年代提出的向量空间模型进行特征表达
- 成功应用于SMART ( System for the Manipulation and Retrieval of Text) 文本检索系统
- 这一系统理论框架到现在仍然是信息检索技术研究的基础

# 模型的描述

- **文档D(Document)**: 泛指文档或文档中的一个片段（如文档中的标题、摘要、正文等）。
- **特征项t (Term)**: 指出现在文档中能够代表文档性质的基本语言单位（如字、词等），也就是通常所指的特征项，这样一个文档D就可以表示为 $D(t_1, t_2, \dots, t_n)$ ，其中n就代表了检索字的数量。
- **特征项权重 $W_k$  (Term Weight)**: 表示特征项 $t_n$ 能够代表文档D能力的大小，体现了项在文档中的重要程度。
- **相似度S (Similarity)**: 指两个文档内容相关程度的大小

# 模型的特点

- 基于特征项(一个文本由一个特征项列表组成)
- 根据特征项的出现频率计算相似度
  - 例如：文档的统计特性
- 用户设定项(**term**)集合，可以给每个项附加权重：  
 $Q = \langle \text{database } 0.5; \text{ text } 0.8; \text{ information } 0.2 \rangle$
- 根据相似度对输出结果进行排序
- 支持自动的相关反馈
  - 有用的项被添加到原始的查询中
  - 例：  $Q \Rightarrow \langle \text{database}; \text{ text}; \text{ information}; \text{ *document* } \rangle$



## 项的权重

- 根据项在文档( $tf$ )和文档集( $idf$ )中的频率(frequency)计算项的权重
  - $tf_{ij}$  = 项j在文档i中的频率
  - $df_j$  = 项j的文档频率 = 包含项j的文档数量
  - $idf_j$  = 项j的反文档频率 =  $\log_2 (N / df_j)$ 
    - $N$ : 文档集中文档总数
    - 反文档频率用项区别文档

# 由特征项构成向量空间

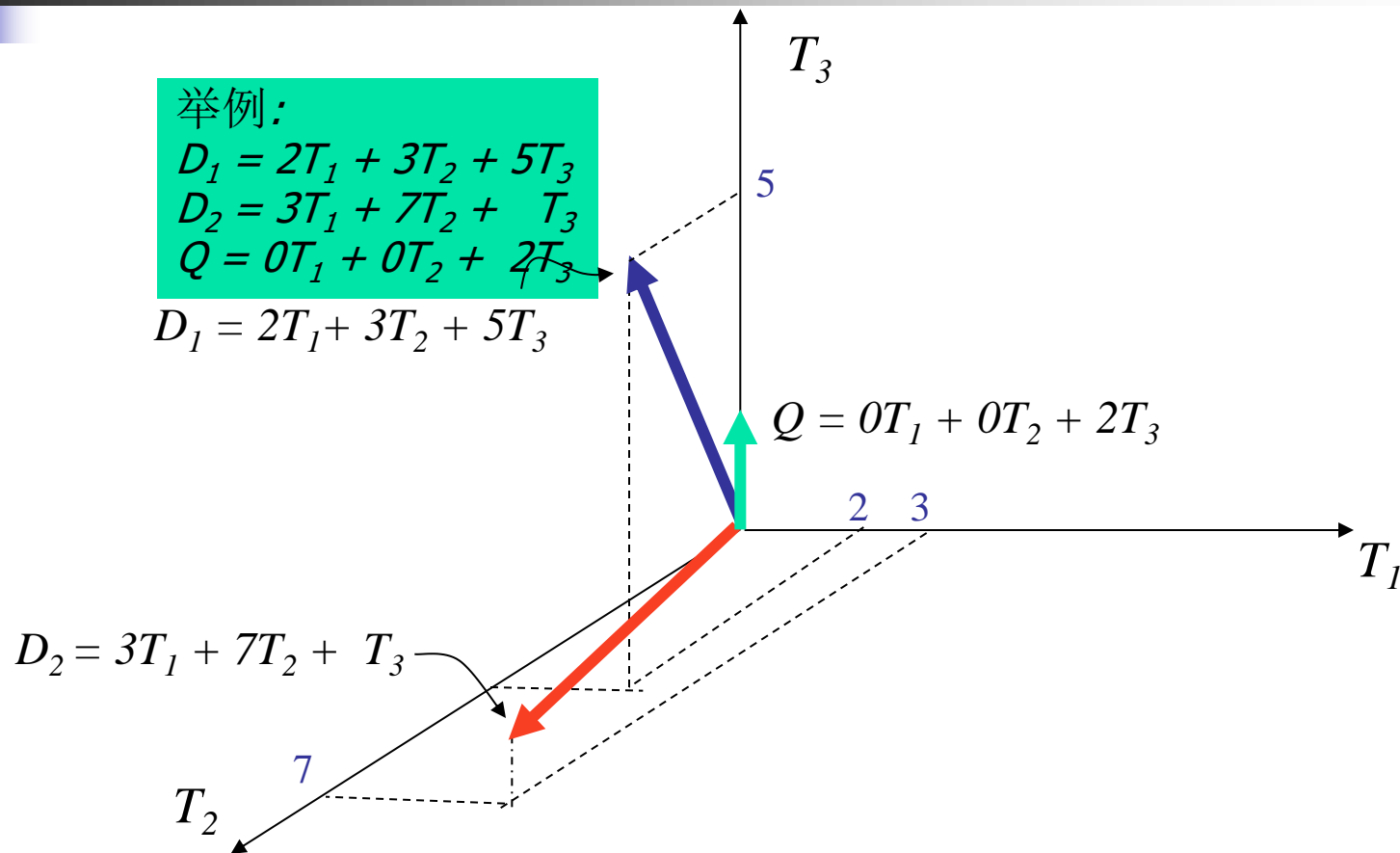
- 2个特征项构成一个二维空间，一个文档可能包含0, 1 或2个特征项
  - $d_i = \langle 0, 0 \rangle$  (一个特征项也不包含)
  - $d_j = \langle 0, 0.7 \rangle$  (包含其中一个特征项)
  - $d_k = \langle 1, 2 \rangle$  (包含两个特征项)
- 类似的，3个特征项构成一个三维空间，n个特征项构成n维空间
- 一个文档或查询可以表示为n个元素的线性组合

## 文档集 — 一般表示

- 向量空间中的N个文档可以用一个矩阵表示
- 矩阵中的一个元素对应于文档中一个项的权重。“0”意味着该项在文档中没有意义，或该项不在文档中出现。

$$\begin{array}{ccccc} & T_1 & T_2 & \dots & T_t \\ \begin{array}{c} D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_n \end{array} & \begin{array}{c} d_{11} \\ d_{21} \\ \vdots \\ \vdots \\ d_{n1} \end{array} & \begin{array}{c} d_{12} \\ d_{22} \\ \vdots \\ \vdots \\ d_{n2} \end{array} & \begin{array}{c} \dots \\ \dots \\ \vdots \\ \vdots \\ \dots \end{array} & \begin{array}{c} d_{1t} \\ d_{2t} \\ \vdots \\ \vdots \\ d_{nt} \end{array} \end{array}$$

# 图示



# 相似度计算

- 相似度是一个函数，它给出两个向量之间的相似程度，查询和文档都是向量，各类相似度存在于：
  - 两个文档之间（分类，聚类）
  - 两个查询之间（FAQ）
  - 一个查询和一个文档之间（检索）
- 人们曾提出大量的相似度计算方法，因为最佳的相似度计算方法并不存在。

## 计算查询和文档之间的相似度

- 可以根据预定的重要程度对检索出来的文档进行排序
- 可以通过强制设定某个阈值，控制被检索出来的文档的数量
- 检索结果可以被用于相关反馈中，以便对原始的查询进行修正。（例如：将文档向量和查询向量进行结合）

# 相似度度量 – 内积(Inner Product)

- 文档 $D$ 和查询 $Q$ 可以通过内积进行计算:

$$\text{sim} ( D, Q ) = \sum_{k=1}^t (d_{ik} \bullet q_k)$$

- $d_{ik}$  是文档 $d_i$ 中的项 $k$ 的权重,  $q_k$ 是查询 $Q$ 中项 $k$ 的权重
- 对于二值向量, 内积是查询中的项和文档中的项相互匹配的数量
- 对于加权向量, 内积是查询和文档中相互匹配的项的权重乘积之和

# 内积 - 举例

- 二值 (Binary) :
  - $D = 1, 1, 1, 0, 1, 1, 0$
  - $Q = 1, 0, 1, 0, 0, 1, 1$
  - $\text{sim}(D, Q) = 3$
- 向量的大小 = 词表的大小 = 7
- 0 意味着某个项没有在文档中出现, 或者没有在查询中出现

- 加权

$$D_1 = 2T_1 + 3T_2 + 5T_3 \quad D_2 = 3T_1 + 7T_2 + T_3$$

$$Q = 0T_1 + 0T_2 + 2T_3$$

$$\text{sim}(D_1, Q) = 2*0 + 3*0 + 5*2 = 10$$

$$\text{sim}(D_2, Q) = 3*0 + 7*0 + 1*2 = 2$$



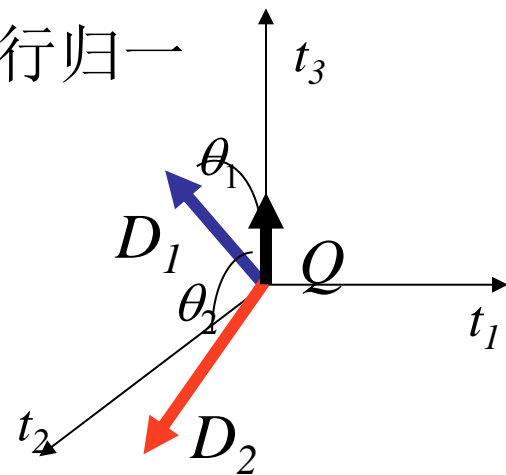
# 内积的特点

- 内积值没有界限
  - 不象概率值，要在 $(0,1)$ 之间
- 对长文档有利
  - 内积用于衡量有多少项匹配成功，而不计算有多少项匹配失败
  - 长文档包含大量独立项，每个项均多次出现，因此一般而言，和查询中的项匹配成功的可能性就会比短文档大。

# 余弦(Cosine)相似度度量

- 余弦相似度计算两个向量的夹角
- 余弦相似度是利用向量长度对内积进行归一化的结果

$$\text{CosSim}(D_i, Q) = \frac{\sum_{k=1}^t (d_{ik} \cdot q_k)}{\sqrt{\sum_{k=1}^t d_{ik}^2 \cdot \sum_{k=1}^t q_k^2}}$$



$$\begin{aligned} D_1 &= 2T_1 + 3T_2 + 5T_3 & \text{CosSim}(D_1, Q) &= 5 / \sqrt{38} = 0.81 \\ D_2 &= 3T_1 + 7T_2 + T_3 & \text{CosSim}(D_2, Q) &= 1 / \sqrt{59} = 0.13 \\ Q &= 0T_1 + 0T_2 + 2T_3 \end{aligned}$$

用余弦计算， $D_1$  比  $D_2$  高6倍；  
用内积计算， $D_1$  比  $D_2$  高5倍

# 其它相似度量方法

- 存在大量的其它相似度量方法

Jaccard Coefficient:

$$\frac{\sum_{k=1}^t (d_{ik} \bullet q_k)}{\sum_{k=1}^t d_{ik}^2 + \sum_{k=1}^t q_k^2 - \sum_{k=1}^t (d_{ik} \bullet q_k)}$$

$$\begin{aligned} D_1 &= 2T_1 + 3T_2 + 5T_3 & \text{Sim}(D_1, Q) &= 10 / (38+4-10) = 10/32 = 0.312 \\ D_2 &= 3T_1 + 7T_2 + T_3 & \text{Sim}(D_2, Q) &= 2 / (59+4-2) = 2/61 = 0.033 \\ Q &= 0T_1 + 0T_2 + 2T_3 \end{aligned}$$

- $D_1$  比  $D_2$  高9.5倍

# 二值化的相似度度量

Inner Product:  $\sum_{k=1}^t (d_{ik} \bullet q_k)$   $|d_i \cap q_k|$

Cosine:  $\frac{\sum_{k=1}^t (d_{ik} \bullet q_k)}{\sqrt{\sum_{k=1}^t d_{ik}^2} \cdot \sqrt{\sum_{k=1}^t q_k^2}}$   $\frac{|d_i \cap q_k|}{\sqrt{|d_i|} \times \sqrt{|q_k|}}$

Jaccard :  $\frac{\sum_{k=1}^t (d_{ik} \bullet q_k)}{\underbrace{\sum_{k=1}^t d_{ik}^2 + \sum_{k=1}^t q_k^2 - \sum_{k=1}^t (d_{ik} \bullet q_k)}_{d_i \text{ 和 } q_k \text{ here are vector}}}$   $\frac{|d_i \cap q_k|}{\underbrace{|d_i| + |q_k| - |d_i \cap q_k|}_{d_i \text{ and } q_k \text{ here are sets of keywords}}}$

# 向量空间优点

- 特征项权重的算法提高了检索的性能
- 部分匹配的策略使得检索的结果文档集更接近用户的检索需求
- 可以根据结果文档对于查询的相关度通过 **Cosine Ranking** 等公式对结果文档进行排序

## 不足

- 随着Web页面信息量的增大、Web格式的多样化，这种方法查询的结果往往会与用户真实的需求相差甚远，而且产生的无用信息量会非常大
- 特征项之间被认为是相互独立
- 潜在语义索引模型是向量空间模型的延伸

# 潜在语义索引(LSI)

# 问题引出

- 自然语言文本中的词汇歧义性.
- 由于一词多义, 基于精确匹配的检索算法会报告许多用户不要的东西
  - 处理
    - 什么地方处理旧家具?
    - 你去把那个叛徒处理了
    - 处理自然语言很难
- 由于一义多词, 基于精确匹配的检索算法又会遗漏许多用户想要的东西
  - 互联网、万维网、因特网、国际互联网等
- 说明以后部分把有代表性词汇称之为特征项



# 特征项 - 文档矩阵

- LSI(Latent Semantic Indexing)将自然语言中的每个文档视为以特征项为维度的空间中的一个点，认为一个包含语义的文档出现在这种空间中，它的分布绝对不是随机的，而是服从某种语义结构。
- 同样地，也将每个特征项视为以文档为维度的空间中的一个点。文档是由特征项组成的，而特征项放到文档中去理解，体现了一种“特征项—文档”双重概率关系。

# LSI的意义

当然,如果能基于自然语言理解来做这件事,那一切问题就都没有了。问题是:

- 自然语言理解的目前水平还是有限度的;
- 即使用自然语言理解,效率也会很低
- 我们希望找到一种办法,既能反映特征项之间内在的相关性,又具有较高的效率.
- 1995年, Berry M.W., Dumais S.T. 等提出了潜在语义分析 (Latent Semantic Indexing), 缩写为LSI) 这一自然语言处理的方法。  
( Dumais S.T.获得了SIGIR2009颁发的Salton奖)

# 算法步骤

- 以特征项(**terms**)为行, 文档(**documents**)为列做一个矩阵(**matrix**)。设为 **t** 行 **d** 列, 矩阵名为 **A**. 矩阵的元素为特征项在文档中的出现频度.
- 数学上可以证明:
  - **A**可以分解为三个矩阵**T0**, **S0**, **D0<sup>T</sup>**(**D0**的转置)的积.
  - 这种分解叫做奇异值分解(singular value decomposition)简称**SVD**
  - $A = T0 * S0 * D0^T$

## 算法步骤

- 一般要求 $T_0$ ,  $S_0$ ,  $D_0$ 都是满秩的. 不难做到把 $S_0$ 的元素沿对角线从大到小排列.
- 现在, 把 $S_0$ 的 $m$ 个对角元素的前 $k$ 个保留, 后 $m-k$ 个置0, 我们可以得到一个新的近似的分解:
  - $\hat{X} = T * S * D^T$
- 奇妙的是,  $\hat{X}$ 在最小二乘意义下是 $X$ 的最佳近似! 这样, 我们实际上有了一个"降维"的途径.
- $K$ 值的选择
  - $k$ 越大失真越小, 但开销越大
  - $k$ 的选择是按实际问题的要求进行平衡的结果

# 三个问题

- 基于A可以计算三类相似度计算
  - 特征项i和j有多相似?
    - 即特征项的类比和聚类问题
  - 文档i和j有多相似?
    - 即文档的分类和聚类问题
  - 特征项i和文档j有多相关?
    - 即特征项和文档的关联问题

# 三个问题的答案

- 比较两个特征项
  - 做"正向"乘法:
  - $\hat{X} \hat{X}^T = T * S * D^T * D * S * T^T = T * S^2 * T^T = (TS) * (TS)^T$
  - $D^T * D = I$ , 因为D已经是正交归一的,  $s = s^T$
  - 它的第i行第j列表明了特征项i和j的相似程度
- 比较两个文档做"逆向"乘法:
  - $\hat{X}^T \hat{X} = D * S * T^T * T * S * D^T = D * S^2 * D^T = (SD) * (SD)^T$
  - $T^T * T = I$ , 因为T已经是正交归一的,  $s = s^T$
  - 它的第i行第j列表明了文档i和j的相似程度
- 比较一个文档和一个特征项恰巧就是Xhat本身.
  - 它的第i行第j列表明了特征项 I 和文档 j 的相似程度.

# 示例

## ■ 原始矩阵 $A$

$$X = \begin{pmatrix} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \text{计算机} & \textcircled{1} & 0 & \textcircled{1} & 0 & 0 & 0 \\ \text{电 脑} & 0 & \textcircled{1} & 0 & 0 & 0 & 0 \\ \text{程 序} & \textcircled{1} & \textcircled{1} & 0 & 0 & 0 & 0 \\ \text{书 桌} & 1 & 0 & 0 & 1 & 1 & 0 \\ \text{办公桌} & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

# 示例

## ■ SVD分解:

$$T = \begin{pmatrix} & \text{cosm.} & \text{astr.} & \text{moon} & \text{car} & \text{truck} \\ \text{Dimension 1} & -0.44 & -0.13 & -0.48 & -0.70 & -0.26 \\ \text{Dimension 2} & -0.30 & -0.33 & -0.51 & 0.35 & 0.65 \\ \text{Dimension 3} & 0.57 & -0.59 & -0.37 & 0.15 & -0.41 \\ \text{Dimension 4} & 0.58 & 0.00 & 0.00 & -0.58 & 0.58 \\ \text{Dimension 5} & 0.25 & 0.73 & -0.61 & 0.16 & -0.09 \end{pmatrix}$$

$$S = \begin{pmatrix} 2.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.59 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.28 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.39 \end{pmatrix}$$

$$D = \begin{pmatrix} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \text{Dimension 1} & -0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\ \text{Dimension 2} & -0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\ \text{Dimension 3} & 0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\ \text{Dimension 4} & 0.00 & 0.00 & 0.58 & 0.00 & -0.58 & 0.58 \\ \text{Dimension 5} & -0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22 \end{pmatrix}$$

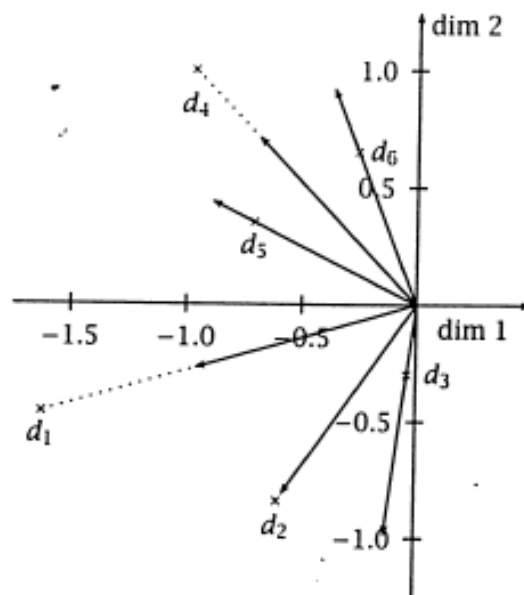


# 示例

- $A$ 降维处理:  $B = S_{2 \times 2} D^T_{2 \times d}$

	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$
Dimension 1	-1.62	-0.60	-0.04	-0.97	-0.71	-0.26
Dimension 2	-0.46	-0.84	-0.30	1.00	0.35	0.65

- 图示:



# 示例

- 向量夹角余弦值:

$$\text{CosSim}(D_i, Q) = \frac{\sum_{k=1}^t (d_{ik} \cdot q_k)}{\sqrt{\sum_{k=1}^t d_{ik}^2 \cdot \sum_{k=1}^t q_k^2}}$$

- 文本之间相似度矩

矩阵

	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$
$d_1$	1.00					
$d_2$	0.78	1.00				
$d_3$	0.40	0.88	1.00			
$d_4$	0.47	-0.18	-0.62	1.00		
$d_5$	0.74	0.16	-0.32	0.94	1.00	
$d_6$	0.10	-0.54	-0.87	0.93	0.74	1.00

## 降维前后的对比

- 表中列出了文档在新空间的相似度， $d_1$ 和 $d_2$ 之间的相似度为0.78， $d_4, d_5$ 和 $d_6$ 为0.94，0.93，0.74，而在原空间上两者的值是相等的
- 在原空间中， $d_2, d_3$ 没有共同的单词，相似度为0，但是在新空间中的相似度为0.88之所已有这种结果，在于它们之间存在着同现模式

# 查询处理

- 如何在降维空间中表示查询字段和新增文档
  - 查询可以作为一个伪文档
- 每次重新计算SVD，计算量太大
- 解决方案： $A = TSD^T, T^T A = T^T TSD^T, T^T A = SD^T$
- 新的查询 $q$ ，再降维后新空间表示为 $T_{t \times k}^T q$ （可以理解为一种映射）

# 对LSI的理解

## ■ 最佳近似矩阵

- 从数据压缩的角度看， $\hat{X}$ 是秩为 $k$ 的前提下矩阵 $X$ 的全局最佳近似矩阵。

## ■ 降维

- LSI不同于向量空间模型（VSM）中文档和词汇的高维表示，而是将文档和词汇的高维表示投影在低维的潜在语义空间（Latent Semantic Space）中，缩小了问题的规模，得到词汇和文档的低维表示。

## ■ 语义关联的发现

- 对应于小奇异值的奇异向量被忽略后，噪声被大量消减，而使语言单元之间的意义上的相关性显示出来。
- 潜在语义空间中（不论是文档空间，还是词汇空间），每个维度代表了一个潜概念（Latent Concept）

# 利用LSI进行检索

## ■ 对查询的要求

- 和传统的基于特征项的查询不同，潜在语义检索允许用户提交类似于自然语言的查询条件，而不一定必须是几个分离的词汇。
- 查询越长，提供的信息需求越充分，越明确

## ■ 检索过程

- 检索过程就是把查询的集合视为是一个虚拟的文档，检索的任务是把这个虚拟的文档和其他文档做相似性比较，挑选最相似的出来
- 相似度计算方法可以采用线性代数理论中的各种方法，比如向量夹角等，根据实际情况而定

# 适用性

- 多数情况下，潜在语义索引的性能好于向量空间模型，因为利用了同现度
- 潜在语义索引的应用依赖于具体的文档集合
- 适用于词汇异构度很高的文档集合
- 从应用角度，计算量太大
- 框架定义完整，优化准则清楚

# 提纲

- Web搜索与信息检索
- 布尔模型以及扩展布尔模型
- 向量空间模型以及潜在语义模型
- 概率模型以及基于语言模型的检索模型
- 知识模型以及基于本体论信息检索

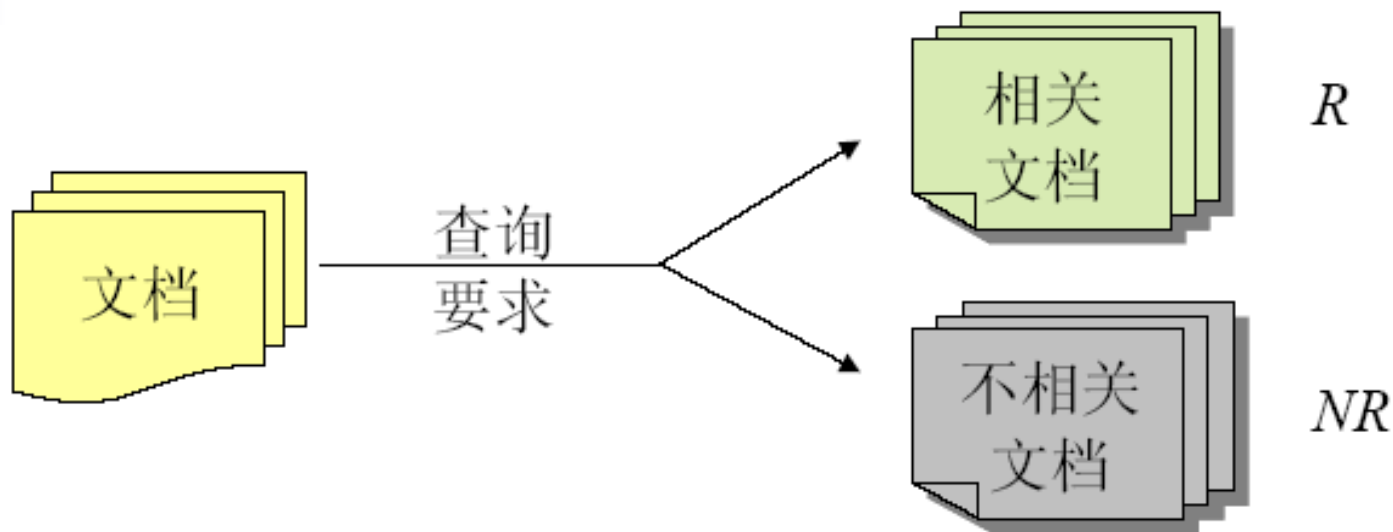


# 概率模型

# 背景

- 概率模型是在布尔逻辑模型的基础上为解决检索中存在的一些不确定性而引入的，试图在概率论的框架下解决信息检索的问题
- 信息检索系统内在存在很多的不确定性
  - 比如对某一信息需求既没有一个查询是唯一的
  - 文档与查询是否“相关”也即文档是否能满足用户的需求也没有一个明确的定义和判定标准
- 信息检索的过程具有的不确定性是概率模型应用到信息检索中的重要前提

# 概率模型



检索问题即求条件概率问题

If  $\text{Prob}(R|d_i, q) > \text{Prob}(NR|d_i, q)$  then  $d_i$ 是检索结果，否则不是检索结果

# 检索的理想结果

- 理想答案集(ideal answer set)
  - 给定一个用户的查询串，相对于该串存在一个包含所有相关文档的集合
  - 我们把这样的集合看作是一个理想的结果文档集
  - 信息检索的过程被看成是描述理想文档集的过程
- 用特征项刻画理想答案集的属性
  - 把查询处理看作是对理想结果文档集属性的处理
  - 我们并不能确切地知道这些属性，我们所知道的是用特征项的语义来刻画这些属性

# 实际策略

## ■ 初始估计

- 由于在查询期间这些属性都是不可见的，这就需要在初始阶段来估计这些属性。
- 这种初始阶段的估计允许我们对首次检索的文档集合返回理想的结果集，并产生一个初步的概率描述。

## ■ 相关反馈(relevance feedback)

- 为了提高理想结果集的描述概率，系统需要与用户进行交互式操作，具体处理过程如下：
  - 用户浏览结果文档，决定哪些是相关的，哪些是不相关的；
  - 然后系统利用该信息重新定义理想结果集的概率描述；
  - 重复以上操作，就会越来越接近真正的结果文档集。

# 概率模型的理论

- 概率模型是基于以下基本假设：
  - 文档与一个查询 的相关性与文档集合中的其他文档是没有关系的，这点被称为概率模型的相关性独立原则；
  - 文档和查询中特征项与特征项之间是相互独立的；
  - 文档和查询中的特征项权重都是二元的；
  - 文档相关性是二值的,即只有相关和不相关两种，也就是说，一篇文档要么属于理想文档集，要么不属于理想文档集。
- 正是由于这些假设，概率模型也被称为二值独立检索模型（Binary Independent Retrieval, BIR）。

# 查询与文档的相关度概率定义

- 在概率模型中特征项的权重都是二值的
  - $w_{i,j} \in \{0,1\}$ ,  $w_{i,q} \in \{0,1\}$ ,
- 查询 $q$ 是特征项集合的子集
- 设 $R$ 是相关文档集合（初始的猜测集合）， $\bar{R}$ 是 $R$ 的补集（非相关文档的集合）
- $P(R|d_j)$ 表示文档 $d_j$ 和查询 $q$ 相关的概率；
- $P(\bar{R}|d_j)$ 表示文档 $d_j$ 和查询 $q$ 不相关的概率；

# 查询与文档的相关度概率定义

- 文档 $d_j$ 对于查询串 $q$ 的相关度值定义为：文档与查询相关的概率和文档与查询不相关概率的比值

$$Sim(d_j, q) = P(R | \vec{d_j}) / P(\bar{R} | \vec{d_j})$$

- 根据贝叶斯原理

$$Sim(d_j, q) = P(\vec{d_j} | R)P(R) / P(\vec{d_j} | \bar{R})P(\bar{R})$$

其中： $P(\vec{d_j} | R)$  代表从相关文档集合 $R$ 中随机选取文档 $d_j$ 的概率， $P(R)$ 表示从整个集合中随机选取一篇文档作为相关文档的概率，依此定义  $P(\vec{d_j} | \bar{R})$  和  $P(\bar{R})$



# 推导

- $P(R)$  和  $P(\bar{R})$  表示从整个文档集合中随机选取一篇文档是否和查询相关先验概率，而对于一个确定的文档集来说，这两个先验概率仅与查询有关，而与具体的每篇文档无关，进一步简化可得
$$Sim(d_j, q) = P(\vec{d}_j | R) / P(\vec{d}_j | \bar{R})$$

- 假设特征项是相互独立的则：

$$Sim(d_j, q) \approx \frac{(\prod_{g_i(d_j)=1} P(k_i | R)) \times (\prod_{g_i(d_j)=0} P(\bar{k}_i | R))}{(\prod_{g_i(d_j)=1} P(k_i | \bar{R})) \times (\prod_{g_i(d_j)=0} P(\bar{k}_i | \bar{R}))}$$

■

# 最终的概率模型排序公式

- $P(k_i | R)$  表示集合  $R$  中随机选取的文档中出现特征项  $k_i$  的概率,  $P(\bar{k}_i | R)$  表示集合  $R$  中随机选取的文档中不出现特征项的概率, 则有:

$$P(k_i | R) + P(\bar{k}_i | R) = 1$$

- 类似定义  $P(k_i | \bar{R})$  和  $P(\bar{k}_i | \bar{R})$ , 在相同查询背景下, 忽略对**所有文档保持不变**的因子, 最终得到:

$$\text{sim}(d_j, q) \approx \sum_{i=1}^t w_{i,q} \times w_{i,j} \times \left( \log \frac{p(k_i | R)}{1 - p(k_i | R)} \right) + \log \frac{1 - p(k_i | \bar{R})}{p(k_i | \bar{R})}$$

这是概率模型主要的排序公式

# 初始化方法

- 由于我们在开始时并不知道集合  $R$ ，因此必须设计一个初始化计算  $P(k_i | R)$  和  $P(k_i | \bar{R})$  的算法。
- 在查询的开始阶段只定义了查询串，还没有得到结果文档集。我们不得不作一些简单的假设，
  - 假定  $P(k_i | R)$  对所有的特征项来说是常数（一般等于  $0.5$ ）
  - 假定特征项在非相关文档中的分布可以由特征项在集合中所有文档中的分布来近似表示。

$$P(k_i | R) = 0.5 \quad P(k_i | \bar{R}) = n_i / N$$

$n_i$  表示出现特征项  $k$  的文档的数目， $N$  是集合中总的文档的数目。

# 改进

- $V$ 表示用概率模型初步检出的经过排序的子集,  $V_i$ 为包含 $k_i$ 的 $V$ 的一个子集。为了改善概率排序, 需要对上述初始化公式改进:

- 通过迄今已检出的文档中特征项 $k_i$ 的分布来估计  $P(k_i | R)$
- 通过假定所有未检出的文献都是不相关的来估计  $P(k_i | \bar{R})$

$$P(k_i | R) = V_i / V$$

$$P(k_i | \bar{R}) = \frac{n_i - V_i}{N - V}$$

- 这一过程可以递归重复

# 概率模型小结

## ■ 优点

- 文档可以按照他们相关概率递减的顺序来排序。

## ■ 缺点

- 开始时需要猜想把文档分为相关和不相关的两个集合，一般来说很难
- 实际上这种模型没有考虑特征项在文档中的频率（因为所有的权重都是二值的）
- 假设特征项独立

- 概率模型是否要比向量空间模型好还存在着争论，但现在**向量空间模型使用的比较广泛**。

## 6 基于统计语言模型的信息检索模型

# 统计语言模型

- 统计语言模型在语音识别中产生
- $\operatorname{argmax} p(s/V)$ ,  $s$ 是文字串,  $V$ 是语音信号
- $\operatorname{argmax} p(s/V) = \operatorname{argmax} p(V/s)p(s)/p(V)$ 
  - $p(V)$ 与 $s$ 的选择无关,  $p(V/s)$ 是声学采样模型
  - $p(s)$ 是语言模型, 表示语言中句子分布概率
- 语言模型: 语言(或者说文档)就是字母表上的某种概率分布, 该分布反映了任何一个字母序列成为该语言的一个句子(或其他语言单位)的可能性, 这个概率分布称之为语言模型(Language Model)

# 统计语言模型

对于句子  $s=s_1s_2\dots s_n$  子将生成句子的过程看成是马尔可夫过程

$$p(s)=p(s_1, s_2, \dots, s_n)=\prod_{i=1, n} p(s_i/s_1s_2\dots s_{i-1})$$

由于没有足够的数据来估计，根据马尔可夫假设，一个词的出现与否仅仅与其前面的  $n-1$  词有关。即：

$$p(s_i/s_1s_2\dots s_{i-1})=p(s_{i-1}, s_{i-2}, \dots, s_{i-n-1})$$

$n=1$  一元语言模型  $p(s_i/s_i)=p(s_i)$

词出现的概率与其他词无关

$n=2$  二元语言模型  $p(s_i/s_{i-1})$

词出现的概率与其前一个词相关

$n=3$  三元语言模型  $p(s_i/s_{i-1}, s_{i-2})$

词出现的概率与其前两个词相关



# 从文档中建立语言模型

原始文本

- $\langle s0 \rangle \langle s \rangle$  *He can buy you the can of soda*  $\langle /s \rangle$

- 一元模型(*Unigram*): (8 words in vocabulary)

- $p1(He) = p1(buy) = p1(you) = p1(the) = p1(of) = p1(soda) = .125, p1(can) = .25$

- 二元模型(*Bigram*):

- $p2(He/\langle s \rangle) = 1, p2(can/He) = 1, p2(buy/can) = .5, p2(of/can) = .5, p2(you/buy) = 1, \dots$

- 三元模型(*Trigram*):

- $p3(He/\langle s0 \rangle, \langle s \rangle) = 1, p3(can/\langle s \rangle, He) = 1, p3(buy/He, can) = 1, p3(of/the, can) = 1, \dots, p3(\langle /s \rangle / of, soda) = 1.$

# 数据稀疏示例

例：假设训练语料只有三句话：

- 1) 小明阅读报纸。
- 2) 小红阅读图书。
- 3) 小刚拿图书换鸡蛋。

需要估计“小明阅读图书”这个句子的概率。以  
<BOS>表示句子的开始和结束。根据**bigram**语言模型，  
可以得到：

■ 
$$\begin{aligned} P(\text{小明阅读图书}) &= P(\text{小明} | < \text{BOS} >) P(\text{阅读} | \text{小明}) P(\text{图书} | \text{阅读}) P(< \text{BOS} > | \text{图书}) \\ &= \frac{1}{3} \times 1 \times \frac{1}{2} \times \frac{1}{2} \approx 0.083 \end{aligned}$$

# 数据稀疏示例

- 然而，如果估计句子“小明拿报纸换鸡蛋”的概率，因为： $P(\text{拿}|\text{小明}) = 0$
- 所以最终显然会得到： $P(\text{小明拿报纸换鸡蛋}) = 0$
- 很显然，这样的估计结果是不准确的，因为“小明拿报纸换鸡蛋”这个句子不仅在语法上是一个合法的句子，而且由人来判断，这也是一个有着合法语义的句子，它是会在现实中出现的句子。
- 解决方案：平滑技术

# 简单的平滑-Laplace法则

$$P(w_i | w_{i-1}) = \frac{1 + c(w_{i-1}w_i)}{\sum_{w_i} (1 + c(w_{i-1}w_i))} = \frac{1 + c(w_{i-1}w_i)}{|V| + \sum_{w_i} c(w_{i-1}w_i)}$$

$$\begin{aligned} P(\text{小明阅读图书}) &= P(\text{小明} | < \text{BOS} >) P(\text{阅读} | \text{小明}) P(\text{图书} | \text{阅读}) P(< \text{BOS} > | \text{图书}) \\ &= \frac{2}{14} \times \frac{2}{12} \times \frac{2}{13} \times \frac{2}{13} \approx 2.8 \times e - 4 \end{aligned}$$

- 这个概率比刚才的**0.083**小了很多，但是也合理了很多

$$\begin{aligned} P(\text{小明拿报纸换鸡蛋}) &= P(\text{小明} | < \text{BOS} >) P(\text{拿} | \text{小明}) P(\text{报纸} | \text{拿}) P(\text{换} | \text{报纸}) P(\text{鸡蛋} | \text{换}) P(< \text{BOS} > | \text{鸡蛋}) \\ &= \frac{2}{14} \times \frac{1}{11} \times \frac{1}{12} \times \frac{1}{12} \times \frac{2}{12} \times \frac{2}{12} \approx 2.5 \times e - 6 \end{aligned}$$

- 尽管这个概率非常小，但是这远比零概率合理多了。

# 基于语言模型的IR模型的概念

- 利用搜索引擎查找一个词串的过程很象在建立语言模型时统计N-gram出现频度的过程。
- 文档语言模型
  - 每个文档对应一个统计语言模型，称为文档的语言模型 (Language Model)。
  - 它主要描述了该文档中各个单词的统计分布特征。
  - 因此每个文档看作是由其语言模型抽样产生的一个样本。
- 基于文档语言模型计算查询的出现概率
  - 一个查询也可以看作是由文档的语言模型抽样产生的一个样本。
  - 因此可以根据每个文档的语言模型抽样生成检索的概率来对其排序，其概率值越大，则该文档就越满足该检索要求。

## 举例

- 假设文档集合中只有1和2两个文本
- 文本1产生的语言模型1
  - $p_1(a)=0.25, p_1(b)=0.5, p_1(c)=1/64, a \in \{c..r\}$ , 剩下的 $s, t, u, v, w, x, y, z$ 均为0
- 文本2产生的语言模型2
  - $p_2(a)=0.7, p_2(b)=0.05, p_2(c)=1/64, a \in \{c..r\}$ , 剩下的 $s, t, u, v, w, x, y, z$ 均为0
- 查询:  $q=abacaad$ 
  - $p_1(q)=0.25*0.5*0.25*1/64*0.25*0.25*1/64 \approx 4.8*10^{-7}$
  - $p_2(q)=0.7*0.05*0.7*1/64*0.7*0.7*1/64 \approx 2.9*10^{-6}$

# 例子中的检索结果

- 从上例中可以看出
  - $q$ 在语言模型1下获得了较低的概率 $4.8 \times 10^{-7}$
  - $q$ 在语言模型2下获得了较高的概率 $2.9 \times 10^{-6}$
- 说明
  - 文本2比文本1更有可能生成 $q$
  - 若输入 $q$ ，应该检索出文本2，而不是文本1

# 与传统概率模型的比较

## 基本思想完全不同

### ■ 传统的信息检索概率模型

- 文档 $d$ 与检索 $q$ 的相关度排序函数定义为事件 $R$ (文档是否满足检索要求)的概率，即： $f(q,d)=P(R|d)$ ；
- 相关度排序函数定义虽然比较直观，但相关性是一个抽象的概念，该定义本身没有也无法具体给出 $R$ 的定义，所以该模型在理论上存在很大的模糊性。

### ■ 基于语言模型的检索模型

- 相关度排序函数则定义为由文档的语言模型生成检索的概率，即 $f(q,d)=p(q|d)$ 。
- 建立在统计语言模型理论基础上，定义明确，便于操作。



# 与传统概率模型的比较（续）

## ■ 具体实施方法不同

### ■ 传统的概率模型

- 由于没有也无法对相关性做出明确定义，因此一般需要在检索中，首先给定带有相关性标记的文档作为建立模型的基础。
- 在实际中，要针对每个检索给定学习数据，几乎不可能。该问题是传统信息检索模型存在的一个主要问题。

### ■ 基于语言模型的信息检索模型

- 可以基于每个文档直接计算出相关度排序函数，从而有效地避免这个问题
- 还可以用该模型为传统概率模型形成初始检索。

# 提纲

- **Web搜索与信息检索**
- 布尔模型以及扩展布尔模型
- 向量空间模型以及潜在语义模型
- 概率模型以及基于语言模型的检索模型
- **知识模型以及基于本体论信息检索**

# 基于知识图谱的信息检索模型

# 本体论

- 本体论（**Ontology**）最早是哲学的分支，研究客观事物存在的本质。
  - 本体（**ontology**）的含义是形成现象的根本实体(常与“现象”相对)。从哲学的范畴来说，本体是客观存在的一个系统的解释或说明，关心的是客观现实的抽象本质。
  - 它与认识论（**Epistemology**）相对，认识论研究人类知识的本质和来源。本体论研究客观存在，认识论研究主观认知。

# 关于本体的定义

- 在人工智能界，最早给出本体定义的是Neches等人，将本体定义为“给出构成相关领域词汇的基本术语和关系，以及利用这些术语和关系构成的规定这些词汇外延的规则的定义”。
- 1993年，Gruber给出了本体的一个最为流行的定义，即“本体是概念模型的明确的规范说明”。
- 后来，Borst在此基础上，给出了本体的另外一种定义：“本体是共享概念模型的形式化规范说明”。
- Studer等对上述两个定义进行了深入的研究，认为“本体是共享概念模型的明确的形式化规范说明”。

# 本体的分类和内容

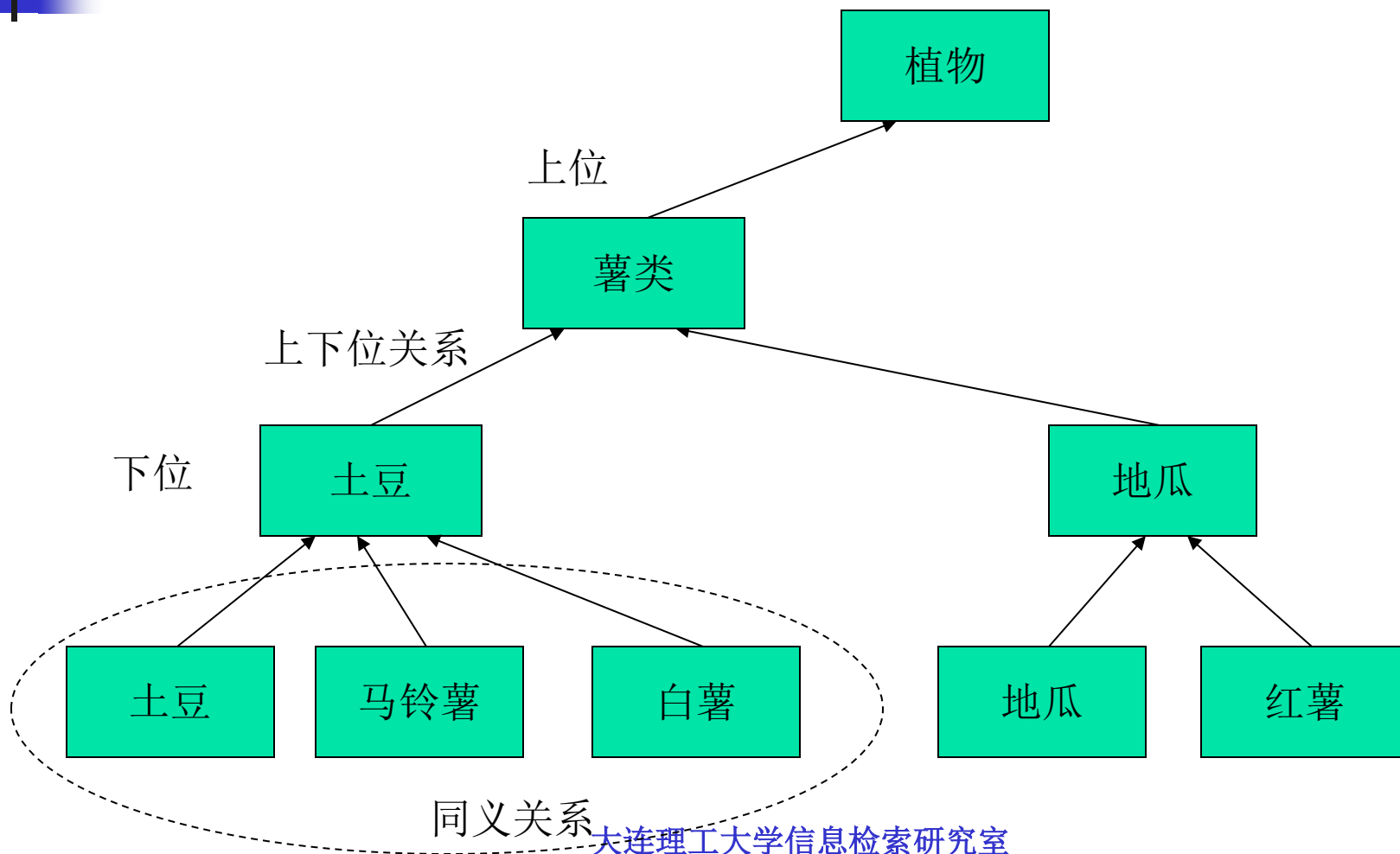
## ■ 本体的分类

- 本体是采用某种语言对概念化的描述，本体的分类按照表示和描述的形式化的程度不同，可以分为：完全非形式化的、半形式化的、严格形式化的，形式化程度越高，越有利于计算机进行自动处理。

## ■ 本体的内容

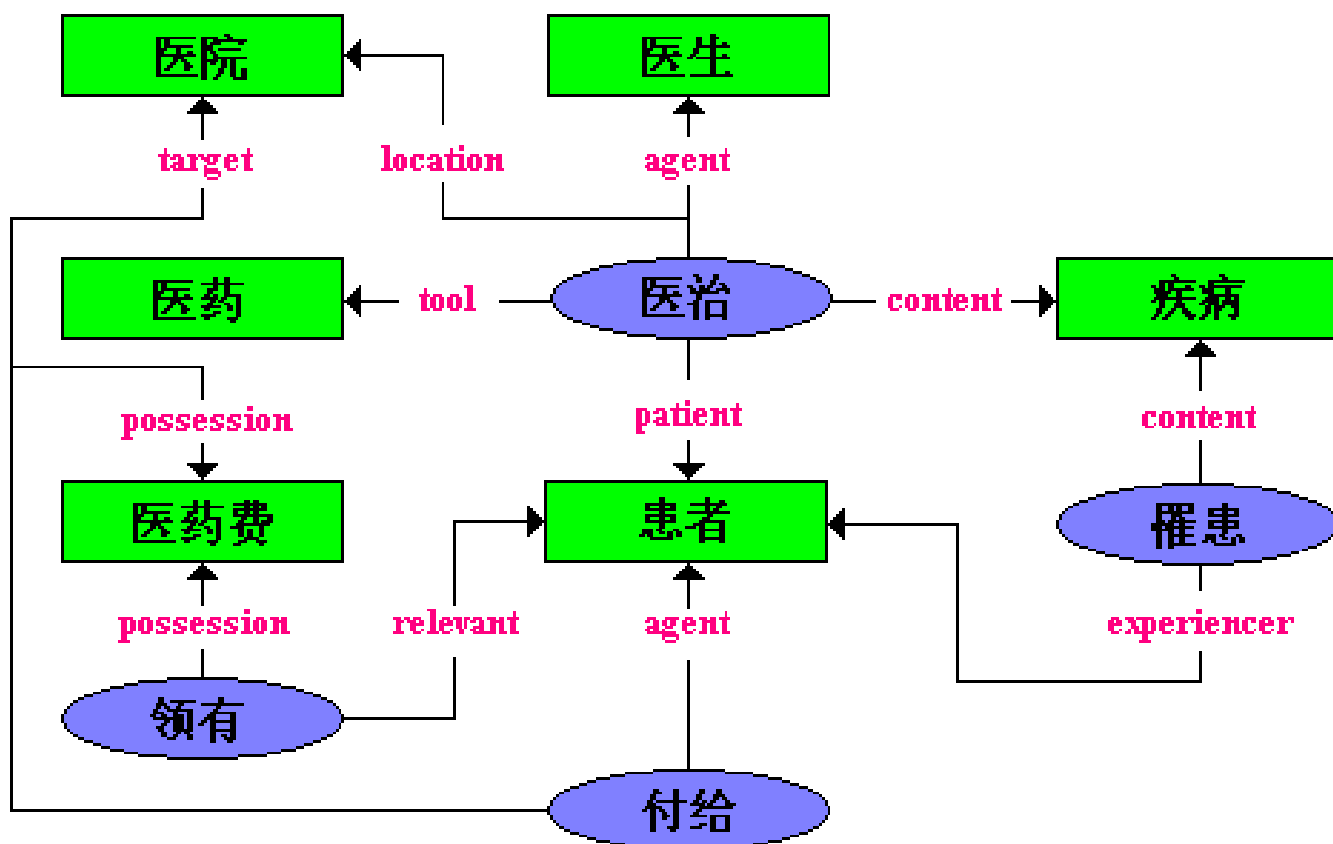
- 从概念化对象的定义来看，一个领域的术语、术语的定义以及各个术语之间的语义网络，应是任一个领域本体论所必须包含的基本信息。
- 概念之间的关系
  - **同义关系**：表达了在相似数据源间的一种等价关系，是一种对称关系
  - **上下位关系**：不对称的，是一种偏序关系，具有传递性
  - 其它各种语义关系
- 各个概念间复杂的语义关系组成了语义网络图，概念在其中表现为节点，而节点间的弧则代表了上述的关系。

# 上下位关系和同义关系





# 语义关系





# 构造本体的要点

- 出于对各自问题域和具体工程的考虑，构造本体的过程各不相同。目前没有一个标准的本体的构造方法。
- 最有影响的是Gruber在1995年提出的5条规则：
  - 清晰（Clarity）
    - 本体必须有效的说明所定义术语的意思。定义应该是客观的，形式化的
  - 一致（Coherence）
    - 它应该支持与其定义相一致的推理
  - 可扩展性（Extendibility）
    - 应该提供概念基础，支持在已有的概念基础上定义新的术语
  - 编码偏好程度最小（Minimal encoding bias）
    - 概念的描述不应该依赖于某一种特殊的符号层的表示方法
  - 本体约定最小（Minimal ontological commitment）
    - 本体约定应该最小，只要能够满足特定的知识共享需求即可。

# 领域本体

- 领域本体(Domain ontology)的概念
  - 提供了某个专业学科领域中概念的词表以及概念间的关系
  - 在该领域里占主导地位的理论，是某一领域的知识表示
- 建立本体的方式
  - 借助某种本体描述语言，采用“悬谈法”从人类专家那里获得知识，经过抽象组织成领域本体。

# 基于本体的检索过程

- 用户向信息检索系统提出检索申请。
- 信息检索系统产生一个界面与用户交互。界面接收用户提出的查询关键字后，系统查询本体库，从中找出出现该关键字的各个领域，然后将其领域以及在该领域下的关键字的含义罗列给用户。
- 用户此时可根据自己的意图，在界面上确定所需查找的领域及含义。
- 系统将经过本体规范后的请求交给全文搜索引擎进行检索。
- 全文搜索引擎检索后返回给用户检索信息。

# 利用本体进行检索的好处

[土豆减肥你相信吗?](#)

土豆减肥你相信吗? 首先, 吃土豆你不必担心脂肪过剩, 因为它只含有0.1%的脂肪; 是所有充饥食物望尘莫及的。每天多吃土豆可以减少脂肪的摄入, 使多余脂肪渐渐代谢掉, 消除你的“心腹之患”。其次, 你也不必...

[www.chinahealthcare.net/jianfei/18.htm](http://www.chinahealthcare.net/jianfei/18.htm) 1K 2000-7-13 - 百度快照

[www.chinahealthcare.net](http://www.chinahealthcare.net) 上的更多结果

1 [2] [3] [4] [5] [6] [7] [8] [9] [10] [下一页](#)

相关搜索

[土豆泥](#)  
[发芽的土豆](#)

[小土豆](#)  
[土豆丝](#)

[土豆的做法](#)  
[土豆头](#)

[酸辣土豆丝](#)  
[土豆炖排骨](#)

[土豆价格](#)  
[>>更多相关搜索...](#)

土豆

百度搜索

在结果中找

[马铃薯](#) [红薯](#) [地瓜](#) [白薯](#)

本体扩展

- 解决从查询语言到检索语言之间转换过程中出现的语义损失和曲解等问题
- 保证在检索过程中能够有效地遵循用户的查询意图, 获得预期的检索信息。

# 基于知识图谱的搜索

- ◆ 知识图谱：它显示知识发展进程与结构关系的一系列不同图形，用可视化技术描述知识资源及其载体，挖掘、分析、构建、绘制和显示知识及它们之间的相互联系。

- ◆ 知识图谱是搜索结果体系化、关联化和可视化

任何一个搜索请求都能得到一个知识体系，不再只是线性的网址列表，而是网状知识结点。

- ◆ 知识图谱给搜索引擎带来的改变。

一是结果更加准确。用户搜索关键词可能有多重意思，知识图谱可以展示最全面的信息，更有机会命中用户需求；

二是结果包括全面的摘要，对于电影，图谱便可看到关联的演员、作者介绍甚至微博相关话题；

三是搜索更广更深，通过知识图谱建立的关系让用户可以通过互动、点击拓展搜索的深度和广度。

# 基于知识图谱的搜索

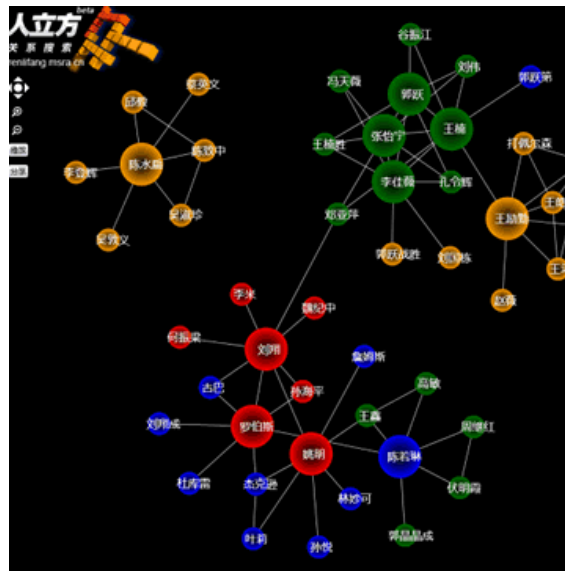
Google: Knowledge Graph

Facebook: Graph Search

Bing: 人立方

搜狗: 知立方

百度知识图谱



知识图谱对“语义识别”技术门槛极高，对社会化开源内容有很强的支撑需求，且是依赖大量用户的行为数据库的产品形态。





# 基于知识图谱的搜索

apple

网页 图片 地图 新闻 更多 搜索工具

找到约 1,770,000,000 条结果 (用时 0.20 秒)

与“apple”相关的广告

[store.apple.com](http://store.apple.com) - 苹果官方在线商店  
[store.apple.com/china](http://store.apple.com/china)

立刻订购iMac与更多Apple全新产品，立享免费送货服务！

购买配备Retina显示屏的 iPad	12期零息零手续费分期付款
学生教育优惠	全新 iMac
购买 iPhone 5	

京东商城官方网站 - JD.COM  
[www.jd.com/](http://www.jd.com/)

全场底价，正品行货，211限时达 中国B2C市场一级网上购物商城！  
 [手机] 京东热-JD-Hot - [图书] 京东开学季 - [家居家纺] 家居家纺开学季

Apple中国  
[www.apple.com.cn/](http://www.apple.com.cn/)

Apple 设计并创造了iPod和iTunes、Mac便携式和台式电脑、OS X操作系统以及革命性的iPhone和iPad。

Mac - iPhone - iPad - 技术支持

北京市附近的apple


Apple Store  
[www.apple.com.cn](http://www.apple.com.cn)  
 4.5 ★★★★★ 27 条 Google 评论

北京市朝阳区三里屯路19号院号  
 三里屯Village 6号楼

“apple”的地图

苹果公司

苹果公司，原称苹果电脑股份有限公司，于2007年1月9日在旧金山Macworld Expo上宣布改为现名。总部位于美国加利福尼亚库比蒂诺，核心业务是电子产品。苹果的Apple II于1970年代助长了个人电脑革命，其后的Macintosh接力于



# 基于知识图谱的搜索

Sogou 搜狗

胡萝卜的热量

搜狗搜索

网页 文档 知识 论坛 新闻 博客 更多 · 什么是分类搜索

胡萝卜热量

37 大卡/100克

成年人每日所需热量



2200-2400 大卡



2200-2400 大卡

消耗37大卡需要做以下运动

走路	9分钟
跑步	3分钟
游泳	2分钟

胡萝卜的营养 热量 减肥功效 薄荷网

【食物介绍】 胡萝卜为伞形科,一年生或二年生的根菜。原产地中海沿岸,我国现...

【食物热量】 25.00大卡(100克可食部分)

【营养价值】 1. 益肝明目: 胡萝卜含有大量胡萝卜素,这种胡萝卜素的分子结...

【食用效果】: 胡萝卜味甘、性平;入肺、脾经;具有健脾消食,润肠通便,杀虫...

【适用人群】: 一般人都可食用。1. 更适宜癌症、高血压、夜盲症、干眼症患者...

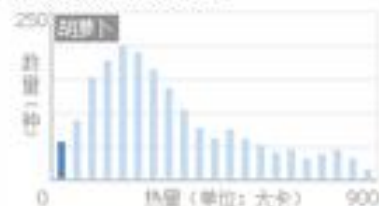
[查看详细营养数据](#) [查询其他食物的营养和热量](#)

薄荷网 - [www.boohy.com/food/](http://www.boohy.com/food/) - 2013-8-26

9种低卡路里的排毒减肥食物 享瘦健康美味 健康频道 美体瘦身 腾讯...

另外,把蘑菇加入到你平时的菜肴中,只要注意食材和其烹饪方式的话,就会是一道非常美味的减肥食谱哦! 热量: 25大卡/每100克 胡萝卜是周所周知的超级减肥食物,那么,它...

常见食物热量分布图



其他蔬菜的热量 (含量/100克)



白菜  
18 大卡



黄瓜  
15 大卡



菠菜  
24 大卡



茄子  
21 大卡

常见食物热量 (含量/100克)



米饭  
116 大卡



馒头  
221 大卡



豆浆  
14 大卡



鸡腿  
181 大卡



牛肉  
125 大卡



# 基于知识图谱的搜索

Home Mail Search News Sports Finance Weather Games Answers Screen

YAHOO! 姚明的身高 Search

姚明的身高

网页 图片 新闻 视频 地图 更多 搜索工具

找到约 1,160,000 条结果 (用时 0.68 秒)

**2.29 米**  
姚明, 身高



沙奎尔·奥尼尔  
2.16 米

勒布朗·詹姆斯  
2.03 米

林书豪  
1.91 米

姚明的身高的图片搜索结果

**姚明**  
 篮球运动员

姚明, 前中国篮球运动员, 生于中国上海, 祖籍为江苏苏州吴江区震泽镇, 是原中国国家篮球队队员, 曾效力于中国篮球职业联赛上海大鲨鱼篮球俱乐部和美国国家篮球协会休斯顿火箭。姚明是中国最具影响力的人物之一, 同时也是世界最知名的华人运动员之一。2009年, 姚明收购上海男篮, 成为上海大鲨鱼篮球俱乐部老板。 [维基百科](#)

生于: 1980 年 9 月 12 日 (34 岁), 上海市  
 身高: 2.29 米  
 体重: 141 公斤  
 配偶: 叶莉 (结婚时间: 2007 年)  
 子女: 姚沁蕾  
 父母: 姚志源, 方凤娣

Jun 20, 2008 · 姚明身高nba火箭队中锋姚明身高到底多少? 目前至少有223、226、227、229厘米4个版本, nba在即将开打的本季网站上认定 ...

# 基于知识图谱的搜索

## 基于知识图谱的搜索引擎:

- 信息抽取目标发生了变化, 传统的文本指定抽取(ACE)=>海量数据的发现(KBP);
- 从文本分析为核心转变成了知识发现为核心;
- 让计算机真正理解用户的查询需求, 给出准确答案而不是给出相关的链接序列;

# 基于知识图谱的搜索

知识图谱：让搜索通往答案本身。

知识图谱：梳理人与信息的联系

搜索引擎的使命转变为连接人与服务，而不再只是连接信息，它需要准确地回答人们的实际问题，给人们提供完备的服务。知识图谱成为智慧搜索的基石。