

5 信息内容检索技术

2019年4月24日 16:27

1 信息检索模型概述

- [1.1 什么是模型?](#)
- [1.2 信息检索模型](#)
- [1.3 模型分类](#)

2 布尔模型

- [2.1 布尔模型概述](#)
- [2.2 布尔模型举例](#)
- [2.3 布尔模型优缺点](#)

3 向量空间模型

4 扩展的布尔模型

5 基于本体论的信息检索模型

1 信息检索模型概述

- [1.1 什么是模型?](#)
- [1.2 信息检索模型](#)
- [1.3 模型分类](#)

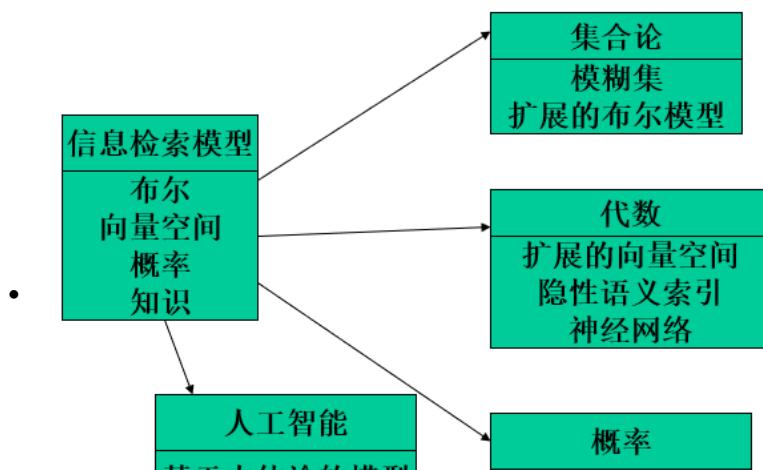
1.1 什么是模型?

- 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

1.2 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式和文档
 - 基于什么样的理论去看待查询式和文档的关系
 - 如何计算查询式和文档之间的相似度

1.3 模型分类



2 布尔模型

- [2.1 布尔模型概述](#)
- [2.2 布尔模型举例](#)
- [2.3 布尔模型优缺点](#)

2.1 布尔模型概述

- 来源
 - 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
- 描述
 - 文档D表示
 - 一个文档被表示为关键词的集合
 - 查询式Q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，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式时，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

2.2 布尔模型举例

- Q=病毒AND (计算机OR电脑) ANDNOT医
- 文档：

D1:...据报道计算机病毒最近猖獗

D2: 小王虽然是学医的，但对研究电脑病毒也感兴趣...

D3: 计算机程序发现了艾滋病病毒传播途径

- 上述文档哪一个会被检索到？
- 查询表示
 - 在布尔模型中，所有索引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}

- 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

用户查询：文档中出现a或者b，但一定要出现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

2.3 布尔模型优缺点

- 优点
 - 到目前为止,布尔模型是最常用的检索模型,因为:
 - 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式
 - 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”
- 缺点
 - 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式呢?

3 向量空间模型

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 (SystemfortheManipulationandRetrievalofText) 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)。
- **索引项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),也就是通常所指的检索词,这样一个文档D就可以表示为
- **D(t₁,t₂,...,t_n)**, 其中n就代表了检索字的数量。
- **特征项权重W_k (Term Weight)** : 指特征项t_n能够代表文档D能力的大小,体现了特征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

- 基于关键词(一个文本由一个关键词列表组成)
- 根据关键词的出现频率计算相似度
 - 例如: 文档的统计特性
- 用户规定一个词项(term)集合,可以给每个词项附加权重
 - 未加权的词项: $Q = \langle \text{database}; \text{text}; \text{information} \rangle$
 - 加权的词项: $Q = \langle \text{database } 0.5; \text{text } 0.8; \text{information } 0.2 \rangle$

- 查询式中没有布尔条件
- 根据相似度对输出结果进行排序
- 支持自动的相关反馈
 - 有用的词项被添加到原始的查询式中
 - 例如: $Q \Rightarrow \langle \text{database}; \text{text}; \text{information}; \text{document} \rangle$

模型中的问题

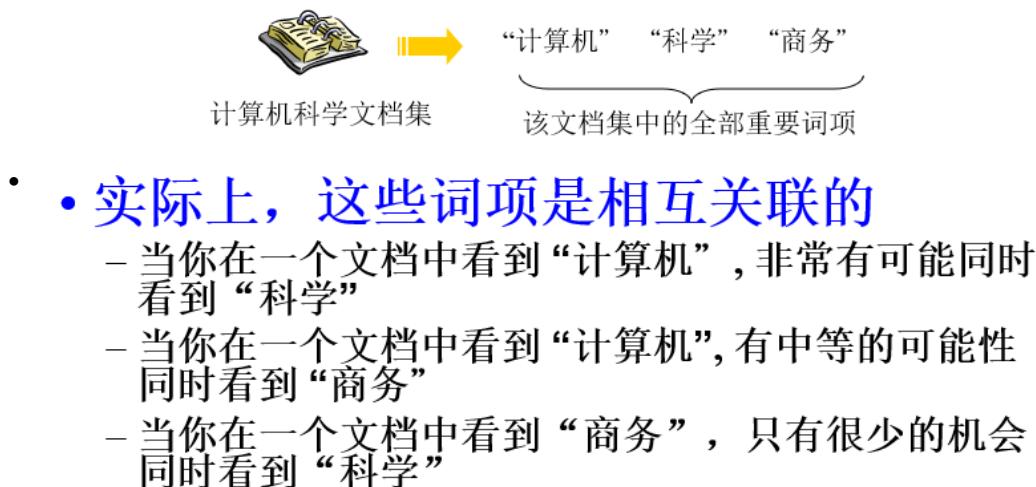
- 怎样确定文档中哪些词是重要的词? (索引项)
- 怎样确定一个词在某个文档中或在整个文档集中的重要程度? (权重)
- 怎样确定一个文档和一个查询式之间的相似度?

索引项的选择

- 若干独立的词项被选作索引项(*index terms*) or 词表*vocabulary*
- 索引项代表了一个应用中的重要词项
 - 计算机科学图书馆中的索引项应该是哪些呢?



- 这些索引项是不相关的 (或者说是正交的), 形成一个向量空间 *vector space*



词项的权重

- 根据词项在文档(*tf*)和文档集(*idf*)中的频率(frequency)计算词项的权重
 - tf_{ij} = 词项j在文档i中的频率
 - df_j = 词项j的文档频率 = 包含词项j的文档数量
 - idf_j = 词项j的反文档频率 = $\log_2 (N / df_j)$
 - N : 文档集中文档总数

- 反文档频率用词项区别文档

Idf计算示例

D1：湖畔的夏夜常常很凉爽，.....

D2：湖畔有家“湖畔”啤酒花园，花园中
常常是鼓鼓的蛙鸣一片，.....

D3：“蛙鸣”禅社举办“蛙鸣”诗会的消息.....

$$N = 3 \quad idf_i = \log\left(\frac{N}{df_i}\right)$$

Term	...	湖畔	夏夜	的	常常	蛙鸣	禅社	诗会	...
df	...	2	1	3	2	2	1	1	...
idf	...	0.176	0.477	0	0.176	0.176	0.477	0.477	...

文档的词项权重(TFIDF举例)

- 文本：“俄罗斯频繁发生恐怖事件，俄罗斯的安全部门加大打击恐怖主义的力度。”

	TF	IDF	TFIDF		TF	IDF	TFIDF	
俄罗斯	2	较高	高		安全	1	中等	高
恐怖	2	较高	高		部门	1	较低	低
的	2	非常低	很低		加大	1	较低	低
频繁	1	较低	低		打击	1	中等	高
发生	1	较低	低		主义	1	较低	低
事件	1	较低	低		力度	1	中等	高

查询式的词项权重

- 如果词项出现在查询式中，则该词项在查询式中的权重为1，否则为0
- 也可以用用户指定查询式中词项的权重
- 一个自然语言查询式可以被看成一个文档
 - 查询式：“有没有周杰伦的歌？”会被转换为：
<周杰伦, 歌>
 - 查询式：“请帮我找关于俄罗斯和车臣之间的战争以及车臣恐怖主义首脑的资料”会被转换为：
<俄罗斯 2, 车臣 2, 战争 1, 恐怖主义 1, 首脑 1>
 - 过滤掉了：“请帮我找”，“和”，“之间的”，“以及”，“的资料”
- 两个文档之间的相似度可以同理计算

由索引项构成向量空间

- 2个索引项构成一个二维空间，一个文档可能包含0, 1或2个索引项

可能包含0, 1或2个索引项

- $d_i = \langle 0, 0 \rangle$ (一个索引项也不包含)

- $d_j = \langle 0, 0.7 \rangle$ (包含其中一个索引项)

- $d_k = \langle 1, 2 \rangle$ (包含两个索引项)

- 类似的，3个索引项构成一个三维空间， n 个索引项构成 n 维空间
- 一个文档或查询式可以表示为 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词项的权重。“0”意味着该词项在文档中没有意义，或该词项不在文档中出现。

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

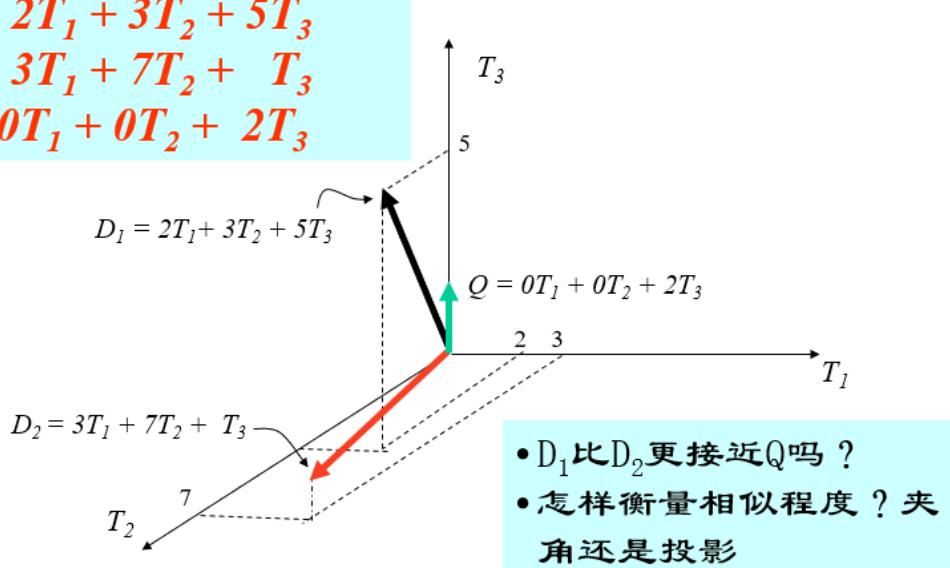
图示

举例：

$$D_1 = 2T_1 + 3T_2 + 5T_3$$

$$D_2 = 3T_1 + 7T_2 + T_3$$

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



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，查询式和文档都是向量，各类相似度存在于：

- 两个文档之间（文本分类，聚类）

- 两个查询式之间（常问问题集）

↑查询式和一个文档之间（检索）

- 一个查询式和一个文档之间（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。

通过计算查询式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，以便对原始的查询式进行修正。（例如：将文档向量和查询式向量进行结合）

相似度度量-内积(InnerProduct)

- 文档 D 和查询式 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 中的词项 k 的权重， q_k 是查询式 Q 中词项 k 的权重
- 对于二值向量，内积是查询式中的词项和文档中的词项相互匹配的数量
- 对于加权向量，内积是查询式和文档中相互匹配的词项的权重乘积之和

内积举例

- 二值 (Binary)
- $D = [1, 1, 1, 0, 1, 1, 0]$
- $Q = [1, 0, 1, 0, 0, 1, 1]$
- 向量的大小 = 词表的大小 = 7
- 0 意味着某个词项没有在文档中出现，或者没有在查询式中出现
- 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

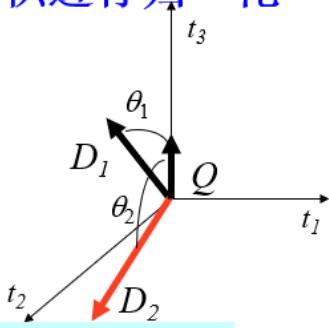
- 内积值没有界限
 - 不象概率值，要在(0,1)之间
- 对长文档有利
 - 内积用于衡量有多少词项匹配成功，而不计算有多少词项匹配失败
 - 长文档包含大量独立词项，每个词项均多次出现，因此一般而言，和查询式中的词项匹配成功的可能性就会比短文档大。

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

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



$$D_1 = 2T_1 + 3T_2 + 5T_3 \quad \text{CosSim}(D_1, Q) = 5 / \sqrt{38} = 0.81$$

$$D_2 = 3T_1 + 7T_2 + T_3 \quad \text{CosSim}(D_2, Q) = 1 / \sqrt{59} = 0.13$$

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

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

其它相似度度量方法

- 存在大量的其它相似度度量方法

Jaccard Coefficient:
$$\frac{\sum_{k=1}^t (d_{ik} \cdot q_k)}{\sum_{k=1}^t d_{ik}^2 + \sum_{k=1}^t q_k^2 - \sum_{k=1}^t (d_{ik} \cdot q_k)}$$

$$D_1 = 2T_1 + 3T_2 + 5T_3 \quad \text{Sim}(D_1, Q) = 10 / (38+4-10) = 10/32 = 0.312$$

$$D_2 = 3T_1 + 7T_2 + T_3 \quad \text{Sim}(D_2, Q) = 2 / (59+4-2) = 2/61 = 0.033$$

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

■ D_1 比 D_2 高9.5倍

示例

Query = “夏夜湖畔的蛙鸣”

Term W_{ij} Doc	...	湖畔	夏夜	的	常常	蛙鸣	禅社	诗会	...
D_1	...	0.176	0.477	0	0.176	0	0	0	...
D_2		0.352	0	0	0.176	0.176	0	0	
D_3		0	0	0	0	0.352	0.477	0.477	
Q	...	0.176	0.477	0	0	0.176	0	0	...

$$\text{Cos}(q, d_1) = 0.893 \quad \text{Cos}(q, d_2) = 0.400 \quad \text{Cos}(q, d_3) = 0.151$$

与查询 q 相似的文档顺序: $d_1 \succ d_2 \succ d_3$

二值化的相似度量

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

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

Jaccard :

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

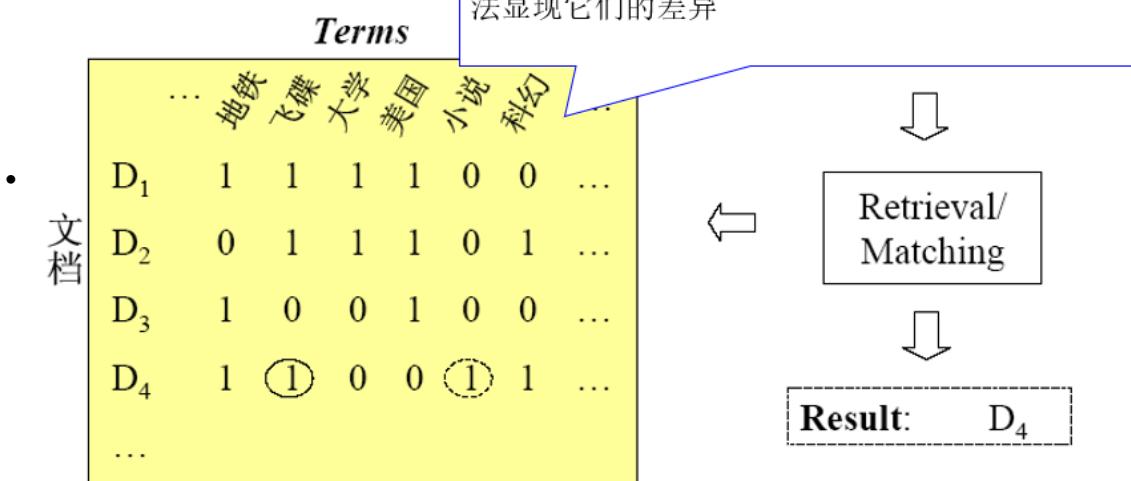
向量空间

- 优点
 - 术语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串的相关度通过CosineRanking等公式对结果文档进行排序
- 不足
 - 标引词之间被认为是相互独立
 - 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
 - 隐含语义索引模型是向量空间模型的延伸

4 扩展的布尔模型

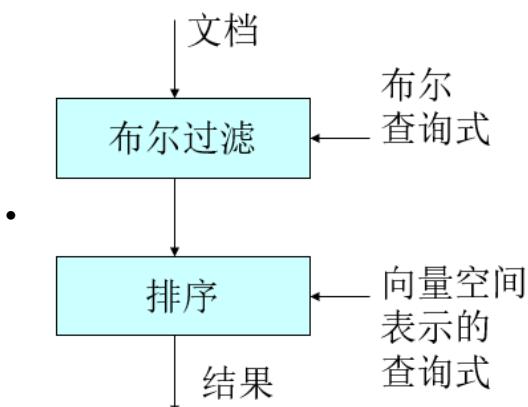
布尔检索示例

Tip “飞碟” AND “小说”：只能检索出D4，无法显现D1,D2,D3的差异
 “飞碟” OR “小说”：可以检出D1,D2,D4，但无法显现它们的差异



布尔模型和向量空间模型相结合

- 布尔模型可以和向量空间模型相结合，先做布尔过滤，然后进行排序：
 - 首先进行布尔查询
 - 将全部满足布尔查询的文档汇集成一个文档
 - 用向量空间法对布尔检索结果进行排序



- 如果忽略布尔关系的话，向量空间查询式和布尔查询式是相同的

先“布尔”，后“排序”存在的问题

- 如果“与”应用于布尔查询式，结果集可能太窄，因而影响了后面的排序过程

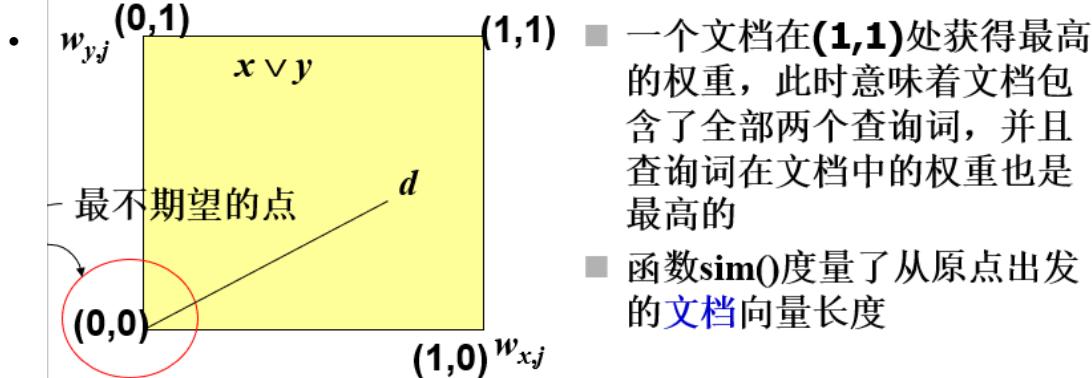
- 如果“或”应用于布尔查询式，就和纯向量空间模型没有区别了
- 在第一步，如何最佳地应用布尔模型呢？
- 提出扩展布尔模型

扩展布尔模型中的“或”关系

- 给定一个或关系的查询式： $x \vee y$
- 假设文档 d_j 中x和y的权重被归一化在(0,1)区间内：

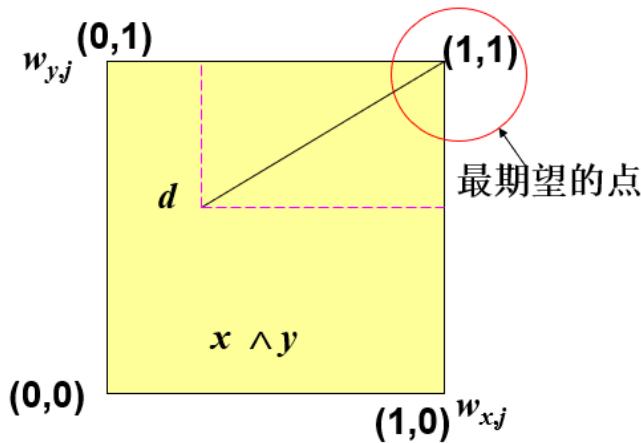
$$- w_{x,j} = (tf_{x,j} / \max_l tf_{l,j}) \times (\text{idf}_x / \max_i \text{idf}_i)$$

$$- \text{sim}(q_{\text{or}}, d_j) = [(x^2 + y^2)/2]^{0.5} \text{ where } x = w_{x,j} \text{ and } y = w_{y,j}$$



扩展布尔模型中的“与”关系

- 给定一个联合的查询式 $x \wedge y$
- $\text{sim}(q_{\text{and}}, d_j) = 1 - \{ [(1-x)^2 + (1-y)^2]/2 \}^{0.5}$
- 函数sim() 表示从(1,1)出发到d的向量长度



扩展的布尔检索相似度计算示例

Query Doc Sim()	“飞碟” AND “小说”	“飞碟” OR “小说”
D ₁	0.293	0.707
D ₂	0.293	0.707
D ₃	0	0
D ₄	1	1

$x, y = 1 \text{ if a term exists in } d_j$

从“一刀切”到“合理拉开差距”

$x, y = 0$ otherwise

观察

- 如果权值是布尔型的， x 出现在文档 d_j 中，则 x 在文档 d_j 中具有权重1，否则为0

- 当 d_j 包含 x 和 y 时

$$\text{sim}(q_{\text{and}}, d_j) = \text{sim}(q_{\text{or}}, d_j) = 1$$

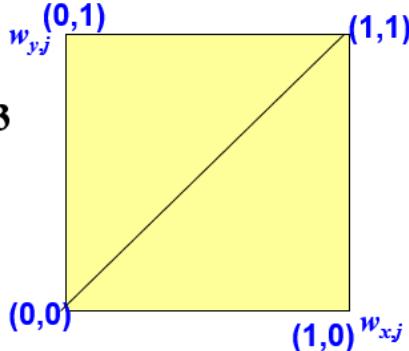
- 当 d_j 既不包含 x 也不包含 y 时

$$\text{sim}(q_{\text{and}}, d_j) = \text{sim}(q_{\text{or}}, d_j) = 0$$

- 当 d_j 包含 x 和 y 二者之一时

$$\text{sim}(q_{\text{and}}, d_j) = 1 - 1/2^{0.5} = 0.293$$

$$\text{sim}(q_{\text{or}}, d_j) = 1/2^{0.5} = 0.707$$



- 一个词项的存在将对“或”关系查询式提供0.707的增益值，但对“与”关系查询式仅提供0.293的增益值

- 一个词项不存在，将给“与”关系的查询式提供0.707的罚分

- 当 x 和 y 有权值0.5, $\text{sim}(q_{\text{and}}, d) = \text{sim}(q_{\text{or}}, d) = 0.5$

- 在一个“与”关系查询中，两个词项的权重均为0.5，则相似度为0.5。其中一个权重为1，另一个为0，相似度为0.293。

- 在“或关系”查询中，情况恰好相反

- 在“与关系”查询中，如果一个词项的权重低于0.5，将给相似度贡献一个较大的罚分

p -norm 模型

- 扩展布尔模型可以被泛化为 m 个查询项:

$$\text{sim}(q_{\text{or}}, d) = [(x_1^2 + x_2^2 + \dots + x_m^2) / m]^{0.5}$$

$$\text{sim}(q_{\text{and}}, d) = 1 - \{[(1-x_1)^2 + (1-x_2)^2 + \dots + (1-x_m)^2] / m\}^{0.5}$$

- 它可以被进一步地泛化为 p -norm model:

$$\text{sim}(q_{\text{or}}, d) = [(x_1^p + x_2^p + \dots + x_m^p) / m]^{1/p}$$

$$\text{sim}(q_{\text{and}}, d) = 1 - \{[(1-x_1)^p + (1-x_2)^p + \dots + (1-x_m)^p] / m\}^{1/p}$$

- 当 $p = 1$ 时, $\text{sim}(q_{\text{or}}, d) = \text{sim}(q_{\text{and}}, d) = (x_1 + x_2 + \dots + x_m) / m$

- 通过语词-文献权值的和来求合取和析取查询的值，和向量空间中的内积相似

- 当 $p = \infty$, $\text{sim}(q_{\text{or}}, d) = \max(x_i)$; $\text{sim}(q_{\text{and}}, d) = \min(x_i)$

- 模糊逻辑模型(Fuzzy logic model)

5 基于本体论的信息检索模型

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。

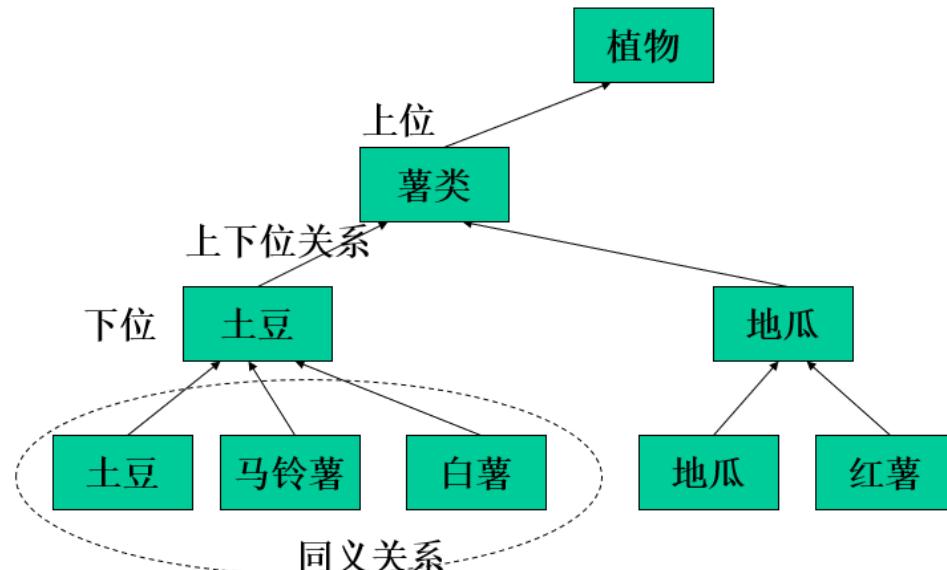
各种关于本体的定义

- 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”。

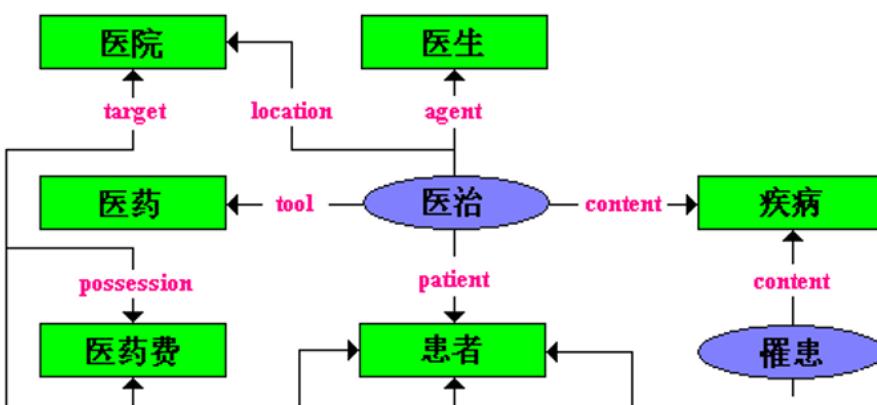
本体的分类和内容

- 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(Domainontology)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
- 应用实例
 - IBM中国研究中心在信息集成项目中运用本体
 - 哈工大机器翻译研究室基于本体进行跨语言检索的研究

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

- 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。