Ch 5

基本概念

- 信息检索模型
- 集合
- 命题
- 联结词

基本原理

- IR四元组
- 布尔模型
  - 布尔模型基本原理
  - 优点
  - 缺点
- 向量空间模型（Vector Space Model）
  - 模型描述
  - 特点
  - 文档向量的构造
  - 查询式的词项权重
  - 由索引项构成向量空间
  - 文档集-表示
  - 计算查询式和文档之间的相似度
  - 优点
  - 缺点
- 概率模型
  - 贝叶斯公式
  - 模型描述
  - 优点
  - 缺点
- tf-idf
- 内积（inner product）

特点
Jaccard相似性（杰卡德相似性）
条件概率

基本概念

信息检索模型

1. 信息检索模型是指如何对查询和文档进行表示，然后对它们进行相似度计算的框架和方法
2. 本质上是对相关度的建模
3. 信息检索模型是IR中的核心内容之一

集合

1. 由一个或多个确定的元素所构成的整体
2. 确定性、互异性、无序性

命题

1. 能表达判断的陈述句，具有确定值
2. 两种类型：原子命题（不能分解为更简单的陈述句）和复合命题（由联结词、标点符号和原子命题复合构成的命题）
3. 真值
    a. 命题所表达的判断结果称为命题的真值
    b. 真值只有“真”和“假”两种，记作True和False，分别用符号T和F表示
    c. 由于命题只有两种真值，所以称这种逻辑为二值逻辑；命题的真值是具有客观性质的，而不是由人的主观决定的
    d. 真值是否唯一确定，与是否知道无关

联结词

1. 复合命题是由原子命题与逻辑联结词组合而成，命题的联结方式叫做命题联结词或命题运算符
2. 否定：一元联结词，非，not

3. 合取：and，和
4. 析取：or，或
5. 条件 `condition`
    a. 给定两个命题P和Q，其条件命题是一个复合命题，记作P→Q，读作“如果P,那么Q”或“若P则Q”
    b. 当且仅当P的真值为T，Q的真值为F时，P→Q的真值为F；否则P→Q的真值为T
    c. 我们称P为前件，Q为后件
6. 双条件 `Double Condition`
    a. 给定两个命题P和Q，其复合命题P–><–Q称作双条件命题，读作“P当且仅当Q”，当P和Q的真值相同时，P–><–Q的真值为T，否则P–><–Q的真值为F。

基本原理

IR四元组

1. D 信息资源集合（文档的简单表示和加权表示）
2. Q 用户信息需求集合
3. F 信息资源与信息需求的匹配处理框架
    a. 信息检索的根本任务是信息集合（D）与需求集合（Q）之间基于某种相似度规则的匹配处理，匹配处理框架（F）正是寻求在二者之间建立一种沟通与联系机制，提供对文档视图、提问式以及它们之间关系进行模型化处理的框架与规则
    b. 布尔模型：匹配规则为二值相关性判断 `binary relevance judgement`，匹配运算主要基于集合论的集合基本运算
    c. 向量空间模型：匹配规则采用多值相关性判断 `n-ary relevance judgement`，匹配处理建立在多维向量空间理论和标准的向量线性代数操作基础之上
    d. 概率模型：匹配规则也是多值性的相关性判断，依赖集合论、概率运算和Bayes法则来完成检索的匹配处理
4. R（d_j, q） 匹配计算函数
    a. 用于计算任一文档d_j（d_j∈D）与任一提问q（q∈Q）形成的文档——提问对（d_j，q）之间的相似度大小
    b. 函数值为实数，其取值区间为 [0，1]
    c. 计算方法简单，计算量小；函数值在取值区间均匀分布；针对某一提问所获取的相关文档集合，能够实现合理的排序输出

布尔模型

1. 文档表示：一个文档被表示为关键词的集合
2. 查询式表示：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
3. 匹配
    a. 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
    b. 检索策略基于二值判定标准

布尔模型基本原理

1. 系统索引词集合中的每一个索引词在一篇文档中只有两种状态：出现或不出现。每个索 引词的权值 $w_{ij}\in\{0,1\}$
2. 检索提问式q由三种布尔逻辑运算符“and/∧”、“or/∨”、“not/ ¬ ”连接索引词来构成
3. 提问式q可以被表示成由合取子项 `Conjunctive Components` 组成的析取范式 `Disjunctive Normal Form,简称dnf` 形式
    a. (a and b) or (c and d) or (e and not f)
4. 匹配函数F
    a. 布尔模型对于任何一篇属于D的文档d_j，定义d_j与用户提问q的匹配函数为
    b. $$sim(d_j,q)=\begin{cases}1，如果存在q_{cc}|（q_{cc}\in Q_{dnf}）且对于任何k_i，有g_i(d_j)=g_i(q_{cc}) \\ 0，其他\end{cases}$$
    c. 在这个式子中，函数g_i定义为$g_i(d_j)=W_{jj}$

优点

1. 最常用的检索模型
    a. 查询简单，容易理解
    b. 通过使用复杂的布尔表达式，可以很方便地控制查询结果
2. 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
3. 经过某种训练的用户可以容易地写出布尔查询式
4. 布尔模型可以通过扩展来包含排序的功能

缺点

1. 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
2. 很难控制被检索的文档数量
3. 很难对输出进行排序
4. 很难进行自动的相关反馈
5. 无法体现文档之间的细微差别

向量空间模型（Vector Space Model）

1. 向量空间（vector space）：由一些被称为向量的对象构成的非空集合V
2. 思想：文章的语义通过所使用的词语来表达
3. 基本原理：每一篇文档用一个向量（特征向量）来表达，查询用一个向量来表达，通过向量来计算相似度

模型描述

1. 文档D：泛指文档或文档中的一个片段（文档中的标题/摘要/正文等）
2. 索引项t `term`：出现在文档中能够代表文档性质的基本语言单位（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（t_1,t_2,...,t_n），其中n就代表了检索词的数量
3. 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
4. 相似度S `Similarity`：指两个文档（或文档与查询）内容相关程度的大小

特点

1. 基于关键词（一个文本由一个关键词列表组成）
2. 根据关键词的出现频率计算相似度
3. 用户规定一个词项(term)集合，可以给每个词项附加权重，查询式中没有布尔条件
4. 根据相似度对输出结果进行排序
5. 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

文档向量的构造

1. 对于任一文档$d_j \in D$，都可将它表示为t维向量形式：$d_j = (w_{1j}, w_{2j}, \ldots, w_{ij})$
2. 向量分量w_{ij}代表第i个索引词k_i在文档d_j中所具有的权重，t为系统中索引词的个数
3. 词数与词频

查询式的词项权重

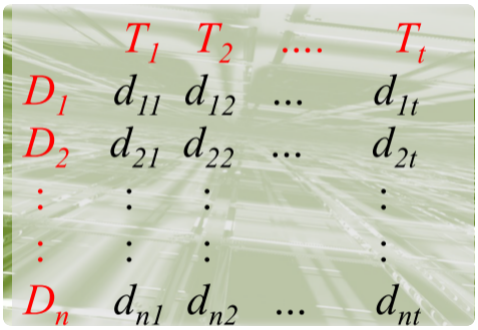
1. 如果词项出现在查询式中，则该词项在查询式中的权重为1，否则为0
2. 也可以用用户指定查询式中词项的权重

由索引项构成向量空间

1. 2个索引项构成一个二维空间（平面），一个文档可能包含0, 1 或2个索引项
   a. $d_i = \langle 0, 0 \rangle$ (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）
2. 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
3. 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–表示

1. 矩阵



计算查询式和文档之间的相似度

1. 根据预定的重要程度对检索出来的文档进行排序
2. 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
3. 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

优点

1. 术语权重的算法提高了检索的性能
2. 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
3. 可以根据结果文档对于查询串的相关度通过Cosine Ranking等公式对结果文档进行排序

缺点

1. 标引词之间被认为是相互独立
2. 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
3. 隐含语义索引（LSI）等模型是向量空间模型的延伸

概率模型

贝叶斯公式

1. $$P(a \mid b) = \frac{P(b \mid a) \times P(a)}{P(b)}$$

模型描述

1. 检索问题即求条件概率问题
2. $$If\ P(R/d_i,q) > P(NR/d_i,q)$$
   then d_i是检索结果，否则不是检索结果
3. $$Dis(RT) = \frac{P(R|RT)}{P(\bar{R}|RT)}$$
4. 提高要求，返回结果集中，有效的是无效的3倍以上，$P(R|RT) > 0.75$
5. $$Dis(RT) = \frac{P(RT|R) \times P(R)}{P(RT|\bar{R}) \times P(\bar{R})}$$

6. 直接看ppt（公式代表的含义）

优点

1. 文档可以按照他们相关概率递减的顺序来排序

缺点

1. 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
2. 实际上这种模型没有考虑索引术语在文档中的频率（因为所有的权重都是二值的）

tf–idf

1. 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
2. 词项的重要性随着它在文档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
   - 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**逆文档频率**= $\log(N/df_j)$
     - N: 文档集中文档总数
     - 逆文档频率用词项区别文档
   - w_{ij} =词项t_j在文档d_i中的**权重**= $tf_{ij} \times idf_j$
3.
4. $idf_j = \log N - \log df_j$
5. idf越大，表明区别文档的能力越强

内积（inner product）

1. 文档D和查询式Q可以通过内积进行计算：sim（D, Q）= 相乘再相加
2. 二值向量：查询式中的词项和文档中的词项相互匹配的数量
   a. 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

   b. sim(D, Q) = 1*1 + 1*0 + 1*1 + 0*0 + 1*0 + 1*1 + 0*1 = 3

3. 加权向量：查询式和文档中相互匹配的词项的权重乘积之和

特点

1. 没有界限
2. 对长文档有利
    a. 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
    b. 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

Jaccard相似性（杰卡德相似性）

1. 交集/并集 = 交集/（A + B – 交集）

条件概率

设A, B为样本空间S中两个事件, 并且P(A)>0. 则称

$P(B|A)=\frac{P(AB)}{P(A)}$ 为事件A发生条件下事件B发生的概率.

1.
    a. 前提P(A)>0，否则P(B|A)=0
    b. 求P(B|A)时，样本空间由S缩小至A，在A中确定B发生的可能性
    c. 一般情况下, P(B) 不等于P(B|A)，两者含义，发生的条件都不相同
    d. 如果$A \subset B$, 则$P(B|A)=1$. 如果$AB=\phi$, 则$P(B|A)=0$.