web期末内容

题型：判断 + 计算题(附加少量问答题？)

题量可能会写不完。

带尺子（有画图），计算器，A4大抄考后收走

重点关注（计算题，证明题）

需要了解（不需要知道细节：概念、优缺点 —— 题或计算题后的问答题）

重点:（序号代表章节名

1. 略过 当故事看
2. 反爬—— 反爬策略与如何应对反爬策略（需要了解）
3. **网页文字处理**

   1）基于字符、词典匹配的分词（必考）：前向后向最大匹配 最短路径 分词（可看补充说明）

   2）基于统计：N-gram 最基础的一元、二元的需要掌握计算 （给一个表格\先验，计算哪种分词方法更合理）

   3）markov模型 给出所有基础详细计算过程 经观察可得不给分

   了解的点：停用词（是什么、对搜索引擎产生的影响）

   加入位置的倒排表，删除停用词会不会造成什么影响？

   归一化处理（区别、流程）词干提取 词形还原 这两种方法
4. 网页索引： 倒排表建立过程和使用 （不会单独成题）、倒排表优化和扩展

   1）多个布尔表达式组成时处理顺序

   2）跳表指针：多层、流程

   可变字节编码（需要知道怎么算）

   了解：倒排表的扩展（如位置信息、文档属性）—— **从而支持短语的查询**（课件上就有两种）；各类的索引压缩方法（方法和效果、词典和倒排表的压缩都要掌握）

5. 各种指标的计算方法：除了拓展指标（如ERR GMAP等）都要记住 —— 一般不单独出题

   几种特别的插值方法

   各类曲线：怎么画，不同情况下怎么选

   （画图时大概提供一下坐标）

   了解：查询的相关性反馈（是什么，有什么用，常见隐式显式的反馈是在干啥）、查询扩展（怎么做，怎么来）、多样性评估（怎么做，在评估什么）

6. 排序（上）**：向量空间模型**：所有计算内容 tf idf等

   罗基奥算法：都要掌握 概念 计算方法

   了解：

   预训练模型部分

   Word2Vec：基本思路

   两种策略

   联系词袋模型

   skingram

   两种策略的思路和优劣对比

7. 排序（下）：

   page rank 要会算

   pagerank的收敛性问题 重启动机制 重启动的参数D

   （有可能会出一个规模不大的网络用来计算pagerank值）

   HITS算法：

   - 每步都要做归一化
   - 每次得到的结果都是两个值 authority hub

   了解：rank的三类排序学习算法（不用掌握原理，只要掌握基本概念，优劣）、衍生模型就topic sensitive page rank（是什么，大概流程）

8. 个性化检索：

   要求会算：协同过滤的算法实现(topic base和user base)

   其他部分都了解：优缺点即可，不需技术细节；矩阵分解方法的基本原理、相关的一些约束

9. **（必考）PCA分析**

数据规约中的主成分分析！！！（必考）

需要了解:

情境感知需要了解（情境的基本概念、检索情境的基本类型）； 和?

数据采样和数据离散化手段

10. **不考;**
11. 知识图谱导论：知识图谱基本概念和组成元素（三元组）、

    信息抽取任务（五类）在做什么、

    事理图谱基本概念，和常规知识图谱点和边的区别； 多模态知识图谱常见两种类型（属性多模态和实体多模态）的概念与适用场景、各自区别（应用不管）
12. 实体抽取： 基于markov的实体抽取方法（可以和分词的markov模型串起来看）——标签不同，方法论相同

    需要了解：实体对齐/实体链接：在做什么？怎么解决

    命名实体识别的任务 如何做结果衡量?
13. 重点关注DIPRE算法，关于模式的定义，流程

    了解：远程监督的方法，基本思路，会遇到哪些问题，如何解决?
14. 知识图谱与图计算：没有计算 无重点关注内容

    需要了解:

    知识图谱中的表征学习:trans系列模型——不同假设、适用范围;

    知识图谱的推理和补全——常见任务形式和目标;
15. 大模型就是扩展资料，无考试要求
16. 社会网络内容:

    重点关注：独立节点模型和线性阈值模型 —— 计算方法：给一个社会网络，其中每个数字都给好，直到传播过程是什么样的

    传播最大化特性的子模特性证明：非负、单调非减、边际效应递减;

    了解:

    网络传播中，哪些类型角色

    什么指标能衡量传播地位;

    拓展模型打破了什么约束，针对什么问题起到了什么效果（课件最后部分）

Chap2

反爬—— 反爬策略与如何应对反爬策略（需要了解）

Chapter1 爬虫机制

反爬虫机制和应对方法：(1)User-Agent：利用访问网站时浏览器发布的 Request Headers 信息中的 User-Agent 信息，判断用户使用何种方式浏览；**应对方法：**使用自定义 Headers**(2)IP/账号访问频率限制：**；**应对方法：**IP 代理池，停顿；多账号 **(3)动态网页：**使用 Js 程序加载网页内容；**应对方法：**模拟调用请求，寻找 json **(4)验证码；应对方法：**字符识别，机器学习 **(5)蜜罐技术：**使用人类看不到或绝对不会点击的连接；**应对方法：**干涉爬虫路径 **(6) 用户权限限制；应对方法：**钱**(7)多模态呈现方式：**文字转为图像或者视频；**应对方式：**OCR，语音识别

Chap3

网页文字处理

1）基于字符、词典匹配的分词（必考）：前向后向最大匹配 最短路径 分词（可看补充说明）

分词方法 基于字符匹配（机械分词）：

正向最大匹配 FMM：使用户满意->使用/户/满意

反向最大匹配 RMM：使用户满意->使/用户/满意

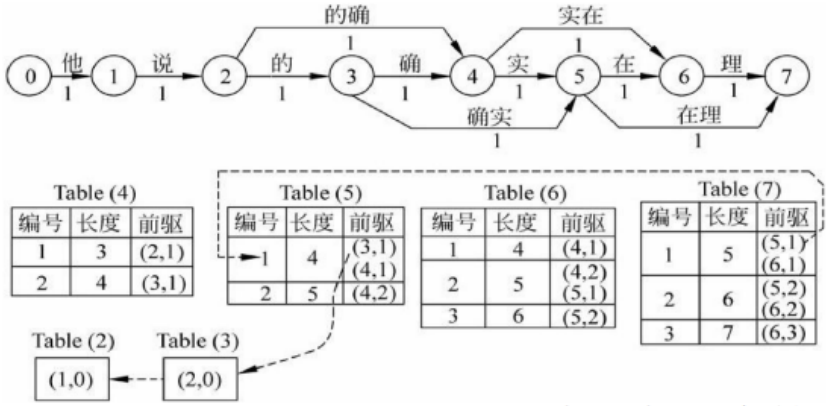
双向最大匹配：BM=FMM+RMM(选词数最少)

南京市长江大桥->南京市长/江/大桥(FMM)，南京市/长江大桥(RMM)->合并为词数最少的“南京市/长江大桥”

最少切分分词（搜索最短路径，边权可视作 1，也可根据词频决定，N-最短路径法）；

优点：效率高、直观；**缺点：**依赖词典（维护词典开支大，难应对新生词汇，词频/重要性不影响结果）

- 拓展方法：**N-最短路径法**，保留N条最短的路径，以提供更多分词方案。



拓展思考：N-最短路径法的复杂度是多少？

这个表的解释：

首先，Table (X)指第 X 个节点对应的路径情况
例如，Table(4)指第 4 个点，即从 0 到“确实”两个字中间这个 4 号点，所有的路径情况。

接下来解释表格中的内容。
编号这个好理解。
长度指的是到这个点为止，路径的长度。比如从 0 到 4，既可以是 0-1-2-3-4 这样长度为 4 的路径，也可以是 0-1-2——4 的长度为 3 的路径。
前驱是与长度相关的，比如长度为 3 的情况，对应 0-1-2-4 这条路径，(2,1)指的是在这条路径中，4 号点的前一个点是 2 号点，对应的是 2 号点的第 1 条路径。

这个可能看 Table(4)不是很清楚，我们可以以 Table(7)的编号 1 为例，比如长度为 5 的情况，可以有两种，一种是 0-1-2-3-5-7/0-1-2-4-5-7，一种是 0-1-2-4-6-7，前者对应的就是前驱（5,1）的情况，我们可以看到 Table(5)里编号为 1 的路径，就是长度为 4 的 0-1-2-3-5 或 0-1-2-4-5。显然，在这两条路径后面加上一个 7 号点，就可以得到 Table(7)里对应的第一组路径了。

我们前两年都考过相关的问题。但不对下面的 Table 做要求，只需要你们画出上面的图，然后根据上面的图找出对应的 N-最短路径（一般只会问 2 或者最多 3）即可。因为给的文本一般都很短，你们可以采用像这个案例一样全遍历的方法来形成 N 条最短路径。

2）基于统计：N-gram　　最基础的一元、二元的需要掌握计算　（给一个表格\先验，计算哪种分词方法更合理

基于统计：

(基于统计的分词方法不需要词典，但实际中可以用机械分词方法进行分词，以获得候选的分词集合：既发挥**匹配分词切分速度快、效率高**的特点，又利用了**无词典分词结合上下文识别生词、自动消除歧义的优势**)

N-gram 模型

N-gram指一个由N个单词组成的集合，各单词具有先后顺序。

N-gram模型的马尔可夫假设：

- 当前状态出现的概率仅同过去有限的历史状态有关，而与其他状态无关。
- 具体到分词任务，就是文本中第N个词出现的概率仅仅依赖于它前面的N-1个词，而与其他词无关。

常见的N-gram模型：

- N = 1，一元文法模型（最大概率模型），$P(w) = P(w_1)\,P(w_2)\ldots P(w_n)$
- N = 2，Bigram模型，$P(w) = P(w_1)\,P(w_2|w_1)\ldots P(w_n|w_{n-1})$
- N = 3，Trigram模型，$P(w) = P(w_1)\,P(w_2|w_1)\,P(w_3|w_1w_2)\ldots P(w_n|w_{n-2}w_{n-1})$

以Bigram模型为例，基于最大似然估计进行推断

$$\textbf{Bigram:} \qquad P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

- 其中，$C(w_{n-1}w_n)$指词序列$w_{n-1}w_n$在语料库中出现的次数。
- 而$C(w_n)$指某个单词w_n在语料库中出现的次数。

首先，构造训练语料库，计算所有的$C(w_n)$与$C(w_{n-1}w_n)$。

其次，对于每一个可能的分词序列$\boldsymbol{w}$，计算以下公式

- $P(w) = P(w_1)\,P(w_2|w_1)\ldots P(w_n|w_{n-1})$

最后，返回最大的$P(w)$所对应的分词序列作为结果。

当N=1时，N-gram模型退化为一元文法模型，此时词与词之间是独立的。

$$P(W) = P(w_1, w_2, \ldots, w_i) \approx P(w_1) \times P(w_2) \times \ldots \times P(w_i)$$

独立性假设，一元文法

$$P(w_i) = \frac{w_i\text{在语料库中的出现次数}n}{\text{语料库中的总词数}N}$$

基于统计文法模型的**优缺点**：

优点：减轻了对于词典的依赖性

- 然而，这种依赖并非完全消除，取决于性能与效率的平衡
  - 如果深度结合机械分词（匹配分词），则效率提升但依赖词典
  - 如果减少对词典的依赖，则需要更多地遍历潜在的组合（解空间巨大！）

缺点：依赖已有数据中词频的统计，对于新生词汇或专业词汇不友好

- 冷门领域的稀有词汇往往难以准确划分
- 易受数据集先验偏差（Bias）的影响

N=2的Bigram模型实例:

- **Bigram模型实例**

统计方法

- 例如，判断句子“我想晚上去吃意大利菜”的分词。
- 已知总词数为13,748，各个词出现的频率为:

我	想	晚上	去	吃	意大利	菜
3437	1215	3256	938	213	1506	459

- 同时，各个词序列的共现次数如下:

	我	想	晚上	去	吃	意大利	菜
我	8	1087	0	13	0	0	0
想	3	0	786	0	6	8	6
晚上	3	0	10	860	3	0	12
去	0	0	2	0	19	2	52
吃	2	0	0	0	0	120	1
意大利	19	0	17	0	0	0	0
菜	4	0	0	0	0	1	0

- **Bigram模型实例**

统计方法

- 基于上述统计数据，计算“我 / 想 / 晚上 / 去 / 吃 / 菜”这一分词的可能性。
- P(我) * P(想 | 我) * P(晚上 | 想) * P(去 | 晚上) * P(吃 | 去) * P(菜 | 吃)
- 3437/13748 * 1087/3437 * 786/1215 * 860/3256 * 19/938 * 1/213 = 0.00000128

N=1 一元文法模型实例

- 待分词句子: “有意见分歧”
  - W1: 有/ 意见/ 分歧/
  - W2: 有意/ 见/ 分歧/
- P(W1) = P(有) * P(意见) * P(分歧)
  $= 1.8 \times 10^{-9}$
- P(W2) = P(有意) * P(见) * P(分歧)
  $= 1 \times 10^{-11}$
- 由于P(W1)>P(W2)，因此，第一种分词更合理。

词语	概率
…	…
有	0.0180
有意	0.0005
意见	0.0010
见	0.0002
分歧	0.0001
…	…

3）markov模型 给出所有基础详细计算过程 经观察可得不给分

基于序列标注：

四类标注：B(词的开始)、M(词的中间)、E(词的结束)、S(单字词)。例：中国科学技术大学是中国最好的大学：BMMMMMME S BE BME BE，中国科学技术大学/是/中国/最好的/大学

隐马尔可夫模型 HMM（初始状态概率矩阵，隐含状态转移矩阵、观测状态概率矩阵；观测值集合（字符集合）隐藏状态值集合（BEMS）:维特比算法；具体过程如下

维特比算法的一个实现案例：

初始状态概率矩阵

T	F
0.6	0.4

隐含状态转移概率矩阵

	->T	->F
T	0.7	0.3
F	0.4	0.6

观测状态概率矩阵

	->X	->Y	->Z
T	0.5	0.4	0.1
F	0.1	0.3	0.6

问：观测到显式状态序列依次为X、Y、Z，最可能的隐含状态序列是什么？

(1)初始化,第一个状态的概率，分别为：第一个为 T 的概率 0.6*0.5，第一个为 F 概率 0.4*0.1,选 T

(2)递归，对第二个状态计算 T，F 的概率分别为 0.084 和 0.027（T 的概率：max(0.3*0.7,0.04*0.4)*0.4,F 的概率 max(0.3*0.3,0.04*0.6)*0.3）;对第三个状态 T，F 的概率分别为 0.00588，0.01512，适合为 F(T 的概率:max(0.084*0.7,0.027*0.4)

(3)回溯，从最优路径的终点向前，找到各个时刻的最优状态，还原

- 隐马尔可夫模型的五个核心要素：两个集合、三个矩阵
  - 两个集合：观测值集合（字符集合）、隐藏状态值集合（BEMS）
  - 三个矩阵：
    - 初始状态概率矩阵：第一个字属于某种隐含状态（BMES）的概率
    - 隐含状态转移概率矩阵：各种隐含状态（各种标签）之间的转移概率
    - 观测状态概率矩阵：从隐含状态（标签）到观测值（字符）的转移概率

- 根据上述元素，我们将隐马尔可夫模型下的中文分词问题表述为如下形式：
- 当我们观测到句子$w_1, w_2 \ldots w_n$，其中w_i为第 i 个汉字，我们希望找到相应的标签序列$s_1, s_2 \ldots s_n$，其中s_i为w_i对应的标签（BMES中的一种），使得$P(s_1, s_2 \ldots s_n | w_1, w_2 \ldots w_n)$概率最大。
- 为求解这一目标函数，需要隐马尔可夫模型的两个基本假设：
  - 齐次假设：当前隐藏状态只与上一个状态有关系
    - 即$P(s_i | s_{i-1}, s_{i-2} \ldots s_1) = P(s_i | s_{i-1})$
  - 观测独立性假设：观测值之间互相独立的，只与生成它的状态有关系
    - 即$P(w_1, w_2 \ldots w_n | s_1, s_2 \ldots s_n) = P(w_1 | s_1) P(w_2 | s_2) \ldots P(w_n | s_n)$
- 基于目标函数和两个基本假设，基于隐马模型的中文分词问题转化为：
- 我们希望找到相应的标签序列$s_1, s_2 \ldots s_n$，使得$\prod_{i=1}^{N} P(s_i | s_{i-1}) P(w_i | s_i)$概率最大。
- 针对这一问题，可采用**维特比（Viterbi）算法**进行求解：
  1. **初始化**：对第一个字，分别以BEMS四种状态计算其概率
  2. **递归**：对第 i 个字，遍历四种状态，先计算该状态最可能是由前一时刻的哪个状态转换而来的，再乘以该状态下得到观测值（字）的概率，取最大值。
  3. **终止**：在第 n 个字时，取得到的最大概率，并得到最后一个字的状态标签。
  4. **回溯**：由最优路径的终点向前，找到各个时刻的最优状态，还原全部标签。

- 例题：记S是所有可能的状态的集合， $S = \{s_1, s_2\} = \{T, F\}$
- $I = (s_{i_1}, s_{i_2}, s_{i_3})$是长度为3的状态序列，其对应的观测序列为$W = (w_1, w_2, w_3) = (X, Y, Z)$
- 隐含状态转移概率矩阵为 $A = [a_{ij}]_{2\times2} = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}$，其中，$a_{ij} = P(i_{t+1} = s_j \mid i_t = s_i) \quad i = 1,2; j = 1,2$
  表示在时刻 t 处于状态s_i的条件下在时刻 $t+1$ 转移到状态s_j的概率。
- 观测状态概率矩阵为 $B = [b_j(k)]_{2\times3} = \begin{pmatrix} 0.5 & 0.4 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$，其中，$b_j(k) = P(w_t = v_k \mid i_t = s_j) \quad k = 1,2,3; j = 1,2$
  表示在时刻 t 处于状态s_j的条件下生成观测v_k的概率。
- π 是初始状态概率向量：$\pi = (\pi_1, \pi_2) = (0.6, 0.4)$，其中，$\pi_i = P(i_1 = s_i) \quad i = 1,2$
  表示在时刻 $t = 1$ 处于状态 s_i 的概率

看习题课

了解的点：停用词（是什么、对搜索引擎产生的影响）

停用词是在信息检索或自然语言处理中被忽略的常见词汇，这些词通常是功能词，如"的"、"是"、"在"等，它们在文本中频繁出现，但对理解文本意义没有太多贡献。停用词通常在处理文本数据时被移除，以减小数据维度、提高处理效率，并聚焦分析在信息检索中更有意义的关键词。

对搜索引擎产生的影响：

1. **降低数据维度：** 移除停用词可以减小文本数据的维度，有助于降低存储和处理的开销。
2. **提高搜索效率：** 在搜索引擎中，移除停用词可以减小索引的大小，提高搜索速度。由于停用词在大多数文档中都存在，它们对搜索结果的排序影响较小，因此在索引中去除它们通常不会显著影响搜索质量。

加入位置的倒排表，删除停用词会不会造成什么影响？

没看懂在问什么，如果是先删停用词再建倒排表，那么没影响，或者说是影响不变（停用词的组合有时有意义，比如“的士”）。如果是先建倒排表再删停用词，那位置索引肯定就不对了。

归一化处理（区别、流程）词干提取 词形还原 这两种方法

停用词处理

停用词：文档中频繁出现或对实际语义影响不大的词语

停用词识别方法：文本频率、词频统计、熵计算

为什么要去除：重复率高，会造成索引中的倒排表很长，影响查询性能（2）对最后结果排序没有贡献，反而可能干扰

去停用词隐患：特殊场景下意义/停用词组合有意义

趋势：压缩技术降低停用词表存储开支；**引入词项权重**降低高频词影响；**索引去除技术**排除低于权重的词项

规范化处理

归一化/词根化处理：指的是还原词语的特殊形式的过程(1)词干提取 stemming(基于规则去除单词前后缀获得词根:复数/时态)(2)词性还原 lemmatisation(基于词典将复杂形态转变成基础形态,并不是简单的去除前后缀，根据词典转换)；两个方法目标一样，结果交叠，方法类似，结果上，词干提取不一定得到完整单词，第二个是

拼写错误处理：基于词典或**编辑距离**（两个字符串之间转换所最少需要的编辑操作步数，Dis(“Kitten”,”Sitting”)=3）进行检查和校对

同义词/相关词处理:基于人工维护知识库获取词项关系

Chap4

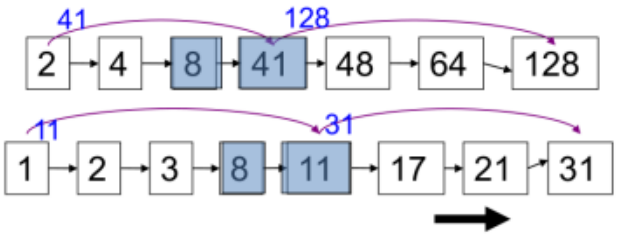
网页索引： 倒排表建立过程和使用 （不会单独成题）、倒排表优化和扩展

1）多个布尔表达式组成时处理顺序

2）跳表指针：多层、流程

- **倒排表合并中的优化问题**

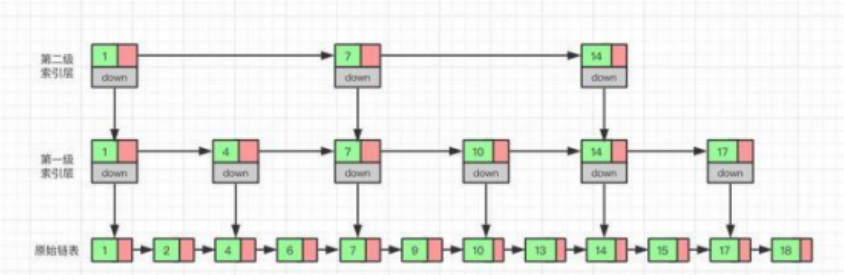
- 带有跳表指针的查询处理过程



- 首先，通过遍历发现了共同的记录8，继续移动指针
- 其次，表2在11的位置，我们发现跳表指针31小于表1的下一个数41
- 因此，我们直接将表2跳到31，而跳过其中的17、21两个数

- 进阶：多层跳表指针
  - 多层跳表指针的推进与单层类似，但高层节点还会多一个Down指针（指向下一层）
    - 如果当前处在第K层，发现对应数值y处在当前节点x_t和下一节点x_t+1之间，则进入下一层（即K-1层）进行更细粒度的比对

该过程与二分查找类似!



可变字节编码（需要知道怎么算）

- **再进一步：可变长度编码**

- 我们的需求：对于一个间距值G，想用最少的所需字节来表示它
- 关键问题：需要利用整数个字节来对每个间距编码
  - 这需要一个可变长度编码，对小数字使用短码来实现这一点
- 可变长度编码的基本流程大致如下：
  - 先存储G，并分配1bit作为延续位
  - 如果G<128，则采用第一位延续位为1 + 7位有效二进制编码的格式
  - 如果G>=128，则先对低阶的7位编码，然后采取相同算法对高阶位进行编码。最后一个字节（8bit）的延续位为1，其他字节延续位为0.

倒排表：可变长度编码（第一个存原值，其它存间距！）

先存储G，并分配1bit作为延续位

如果G<128，则采用第一位延续位为1 + 7位有效二进制编码的格式

如果G>=128，则先对低阶的7位编码，然后采取相同算法对高阶位进行编码。最后一个字节（8bit）的延续位为1，其他字节延续位为0.

214577 的二进制为 1101/0001100/0110001，
VB 编码为 00001101/00001100/10110001

了解：

倒排表的扩展（如位置信息、文档属性）—— **从而支持短语的查询**（课件上就有两种）；

> 二元词索引
>
> 位置信息索引

各类的索引压缩方法（方法和效果、词典和倒排表的压缩都要掌握）

Chap5

各种指标的计算方法：除了拓展指标（如ERR GMAP等）都要记住 —— 一般不单独出题

指标：性能/效率/结果多样性/权威性/时新性与更新频率

单查询评价（1)无序结果评价

P/N:算法对样本的判断;**T/F:**算法判断的正确与否

准确率 Precision: TP/(TP+FP)（相关文档在被检索文档中的数量比例）

召回率 Recall: TP/(TP+FN)（被检索文档在总相关文档中的数量比例）

召回率的近似 缓冲池 Pooling:将各算法给出检索结果中的 TopN 文档汇集并人工标注，得到相关的文档池

F-measure：准确率与召回率的调和平均

$$F=\frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}=\frac{(\beta^2+1)PR}{\beta^2P+R}$$

取 alpha=0.5

或 belta=1

得 F=2PR/(P+R)

通过绘制不同阈值下的指标变化曲线可选择合适阈值。

几种特别的插值方法

各类曲线：怎么画，不同情况下怎么选

（画图时大概提供一下坐标）

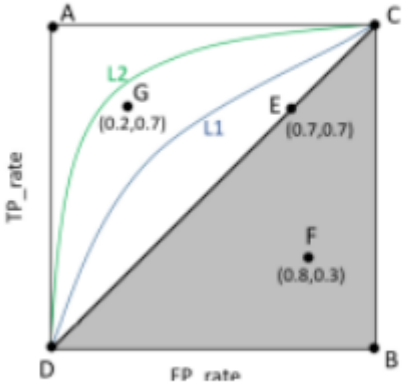
通过绘制不同阈值下的指标变化曲线可选择合适阈值。

ROC 曲线：真正率 TP/(TP+FN)和假正率 FP/(FP+TN)作为轴线，兼顾正负

对角线上无区分能力（即随机猜测）

对角线上端越远越好

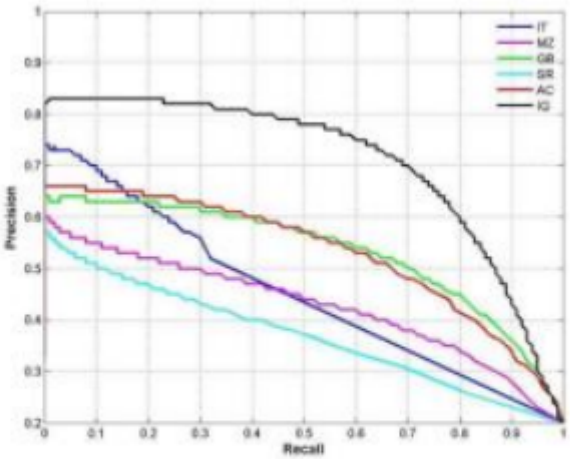
低于对角线无意义（无区分度）



P-R 曲线：x 轴召回率，y 轴准确率，不同阈值不同曲线。（只考虑正例）

Recall 递增，但是 Precision 是震荡的，所以曲线是锯齿

在@N 中也有可能，Recall 不变，Precision 减小



判别算法好坏：(1)计算曲线下面积 AUC，越高效果越好

(2)PR 曲线平衡点（P=R）值越高越好

了解：查询的相关性反馈（是什么，有什么用，常见隐式显式的反馈是在干啥）、查询扩展（怎么做，怎么来）、多样性评估（怎么做，在评估什么）

反馈分类 **显式反馈：**用户显式参与交互过程（用户点击记录只有正样本、用户评论）；**隐式反馈：**系统追踪用户行为来推测返回文档的相关性（鼠标键盘/眼球动作，判断不一定很准确，分析难度高；但省却了用户的显式参与行为，提升体验感）；**伪反馈：**在没有用户参与的前提下，直接假设返回结果是相关的并进行反馈（结果未经用户判断，难以保证其准确性，可能出现查询漂移）。

查询扩展 用户针对词项的合适程度给出反馈，这些反馈将被用来构建更完整的查询条件

(1)同义词辞典：有助于提升查询召回率，但影响准确率

(2)基于搜索日志进行优化：查询日志，挖掘查询等价类

多样性评估

基本形式：给定一个查询 q，返回一个多样化的结果文档集合 R(q)(R(q)中所有的结果文档斗鱼查询 q 本身有较大相关性)

核心思想：降低用户无法获得所需信息的风险

衡量方式：(1)隐式模型：只计算文档之间的差异性(2)显式模型：更加具体的考量文档所对应的用户意图。

MMR: 隐式模型代表

$$MMR^{def} = Argmax_{d_i \in R \setminus S}[\lambda P(d_i \mid q) - (1-\lambda) max_{dj \in S} P(d_i \mid d_j)]$$

前半部分表示文档集和查询 q 的相似性

后半部分表示文档之间的多样性

FM-LDA:显式模型代表

其中 p 表示文档 Di 包含主题 Fj 的概率

Chap6

排序（上）：**向量空间模型**：所有计算内容 tf idf等

相关度计算：

Jaccard 系数： $\mathrm{JACCARD}(A, B) = \frac{|A \cap B|}{|A \cup B|}$

词项频率 TF(t,d): 词项 t 在文档 d 中出现的次数

对数词频：$wf_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d} & tf_{t,d} > 0 \\ 0 & otherwise \end{cases}$

文档频率 DF(t)：出现词项 t 的文档数量

逆文档频率 IDF: $idf_t = log_{10} \frac{N}{df_t}$

TF-IDF: $W_{t,d} = (1 + \log tf_{t,d}) \cdot \log \frac{N}{df_t}$

罗基奥算法：都要掌握 概念 计算方法 （在大抄第五章部分）

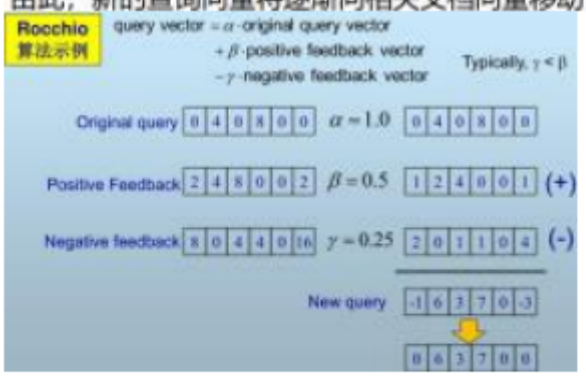
技术：Rocchio 罗基奥算法（使查询尽可能离与之相关的文档更近，离与之不相关的文档更远）

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j$$

其中，D_r为已知相关文档的向量集合，D_{nr}为已知不相关文档的向量集合

q_0为初始查询向量。α、β、γ为权重，根据手工调节或经验设定

由此，新的查询向量将逐渐向相关文档向量移动，远离不相关文档向量



(负分量->0)

正反馈（价值更大）vs 负反馈（收集难度高）

很多系统甚至只允许正反馈，因为手机真正的负反馈难

了解:

预训练模型部分

Word2Vec: 基本思路

两种策略

CBOW

skingram

两种策略的思路和优劣对比

> 这个大抄好像没有，可以添一下:
>
> Word2vec模型两种设计思路的优劣如何?
>
> 从性能上说
> - CBOW模型仅预测中心词，复杂度约为O(V)，即词表规模
> - 而Skip-gram模型基于中心词预测周边词，复杂度约为O(KV)，即考虑窗口
>
> 从效果上说
> - 在Skip-gram模型中，由于每个词都可以作为中心，都将得到针对性训练
> - 因此，对于生僻词（数据稀疏）的训练而言， Skip-gram模型效果更好
>
> Word2Vec的优缺点

```
优点
• 有效表征了词项之间的上下文关系
• 无监督，通用性强，可适用于各种NLP任务
缺点
• 无法解决一词多义的问题，例如play music 和 play football
• Word2vec 是一种静态的方式，其词项表征一旦训练确定就不会再做更改。因此，虽然通用性强，但是无法针对特定任务进行动态优化
```

Chap7

排序（下）：
page rank 要会算
pagerank的收敛性问题 重启动机制 重启动的参数D
（有可能会出一个规模不大的网络用来计算pagerank值）
HITS算法:

- 每步都要做归一化
- 每次得到的结果都是两个值 authority hub

了解：rank的三类排序学习算法（不用掌握原理，只要掌握基本概念，优劣）、衍生模型就topic sensitive page rank（是什么，大概流程）

Chap8

个性化检索:
要求会算：协同过滤的算法实现(topic base和user base)

其他部分都了解：优缺点即可，不需技术细节；矩阵分解方法的基本原理、相关的一些约束

Chap9

（必考）PCA分析
数据规约中的主成分分析！！！（必考）

```
注意：大抄里和ppt不太一样，推荐按ppt里来算

  • 将样本按行写成矩阵(每个样本是一个行向量) X
```

- 计算 XX^T 的前 k 大特征值对应的特征向量，按列写成矩阵(每个特征向量是一列) W
- $X' = XW$

需要了解:
情境感知需要了解（情境的基本概念、检索情境的基本类型）; 和?
数据采样和数据离散化手段

Chap10

不考;

Chap11

知识图谱导论：知识图谱基本概念和组成元素（三元组）、
信息抽取任务（五类）在做什么、
事理图谱基本概念，和常规知识图谱点和边的区别;

> 大抄中没有，可以添一下:
>
> 概念:
>
> 事理图谱 = 事件+逻辑，事理图谱所要描绘的是一个逻辑社会，研究对象是谓词性事件及其内外联系。
> 事理图谱与知识图谱的组织形式相仿，实体通过头尾相连，形成有向图的组织性质
> 借助图谱中的事理逻辑链接，可以形成对于事件的推理
>
> 区别:
>
> 事理图谱更具有动态性，知识图谱相对静态，变化缓慢
>
> 研究对象不同：名词性实体及其关系、谓词性事件及其内外联系
>
> 知识形式不同：<实体、属性、属性值><实体、关系、实体> 三元组；<事件、论元集合、逻辑关系> 多元组

多模态知识图谱常见两种类型（属性多模态和实体多模态）的概念与适用场景、各自区别（应用不管）

Chap12

实体抽取： 基于markov的实体抽取方法（可以和分词的markov模型串起来看）——标签不同，方法论相同
需要了解：实体对齐/实体链接：在做什么？怎么解决

```
一般而言，利用实体的属性信息判定不同源实体是否可对齐

    实体链接，挑战：

语言的多样性：同一个实体可能会有多个不同的提及
语言的歧义性：同一个提及可能会对应多个不同的实体

    补充一下实体对齐和实体消歧的区别

实体消歧旨在消除一词多义的歧义现象
实体对齐旨在表征同一对象的多个实体之间构建对齐关系，丰富实体信息
```

命名实体识别的任务 如何做结果衡量？

Chap13

重点关注DIPRE算法，关于模式的定义，流程

```
模式：包含常量和变量，例如 ?x ，by ?y的形式（可表示“title” by“author”）
```

了解：远程监督的方法，基本思路，会遇到哪些问题，如何解决？

Chap14

知识图谱与图计算：没有计算 无重点关注内容
需要了解：
知识图谱中的表征学习:trans系列模型——不同假设、适用范围；

```
大抄没详细说，可以添一下：

TranE：$\vec{h}+\vec{r}=\vec{t}$，无法处理一对多、多对一和多对多问题
```

```
TransH：通过向量 w 将实体投影到关系 r 对应的超平面上，解决一对多问题。仍然假设不同关系同属于同一个超平面中，不同实体处于相同的语义空间中，这导致它在表示现实世界情况中的实体和关系时是不够充分的

TransR：通过矩阵 Mr 将实体进行坐标转换到关系 r 对应的关系空间中，不同的关系对应实体不同的属性

TransD：关系 r 对应的投影矩阵是动态生成的，描述同一种关系的不同语义
```

知识图谱的推理和补全——常见任务形式和目标；

```
    知识图谱上的关系推理任务

链接预测，给定两个实体，预测它们之间是否存在 r 关系

实体预测，给定头实体和关系，预测未知的尾实体

事实三元组预测，给定一个三元组判断其是否为真或假

关系补全，基于图谱里已有的事实，去推理出缺失的事实
```

Chap15

大模型就是扩展资料，无考试要求

Chap16

社会网络内容：

重点关注：独立节点模型和线性阈值模型 —— 计算方法：给一个社会网络，其中每个数字都给好，直到传播过程是什么样的

传播最大化特性的子模特性证明：非负、单调非减、边际效应递减；

了解：

网络传播中，哪些类型角色 什么指标能衡量传播地位；

```
    意见领袖

可参考HITS算法及两类网页的区分
```

结构洞

一种直观的结构洞的判定方法为：如果一个节点，移除该节点就会使网络变成多个连通组件，则该节点即为一个结构洞

另一种衡量方式：聚集系数，某个节点的聚集系数为：它的任意两个好友也互为好友的概率（比重），显然，聚集系数越低，该节点作为中介的作用也就越大

拓展模型打破了什么约束，针对什么问题起到了什么效果（课件最后部分）

习题课计算题

1.1

1.1 请推荐如下查询的处理次序。
(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)
其中，每个词项对应的倒排记录表的长度分别如下：

词项	倒排记录表长度
eyes	213 312
kaleidoscope	87 009
marmalade	107 913
skies	271 658
tangerine	46 653
trees	316 812

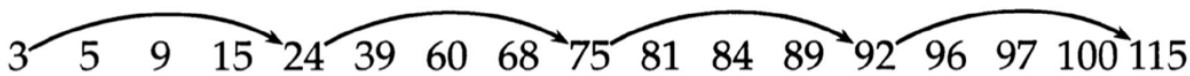
考察知识点：倒排索引的优化（见课件第四节，网页索引）
对于OR操作：顺序任意、考虑O(x+y)的最坏情况
对于AND操作：先处理文档频率小的，再处理大的

- （tangerine OR trees）→ 46653+316812 = 363465
- （marmalade OR skies）→107913+271658 = 379571
- （kaleidoscope OR eyes）→87009+213312 = 30321

由于30321< 363465< 379571
所以推荐处理次序为：
（kaleidoscope OR eyes） AND（tangerine OR trees）AND （marmalade OR skies）

1.2

1.2 考虑利用如下带有跳表指针的倒排记录表



和一个中间结果表（如下所示，不存在跳表指针）进行合并操作。

3 5 89 95 97 99 100 101

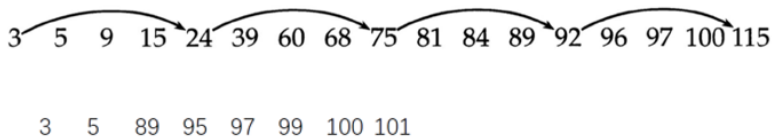
采用基于跳表指针的倒排记录表合并算法，请问：

1) 跳表指针实际发生跳转的次数是多少？
2) 当两个表进行合并时，倒排记录之间的比较次数是多少？
3) 如果不使用跳表指针，那么倒排记录之间的比较次数是多少？

- 考察知识点：倒排表合并的优化

```
INTERSECTWITHSKIPS(p1, p2)
answer ← ⟨ ⟩
while p1 ≠ NIL and p2 ≠ NIL
do if docID(p1) = docID(p2)
      then ADD(answer, docID(p1))
           p1 ← next(p1)
           p2 ← next(p2)
      else if docID(p1) < docID(p2)
              then if hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
                      then while hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
                              do p1 ← skip(p1)
                      else p1 ← next(p1)
              else if hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
                      then while hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
                              do p2 ← skip(p2)
                      else p2 ← next(p2)
return answer
```

1）



1（24 →75）

2）

3-3，5-5，9-89， 15-89，24-89，75-89，92-89，81-89，84-89，89-89，92-95，115-95，96-95，96-97，97-97，100-99，100-100，115-101

18次

3）

3-3，5-5，9-89， 15-89，24-89，39-89，60-89，68-89，75-89，81-89，84-89，89-89，92-95，96-95，96-97，97-97，100-99，100-100，115-101

19次

- 常见错误：主要集中在题2）
  - 在3与3比较后，并没有将两个表的指针同时进一格（即5与5比较），而是将5与24比较。这样的做法并没有错，但是与算法的细节有偏差。
- 倒排记录之间的比较次数（比较过程：Line3是否相等，Line7是否小于，Line8/12是否能跳，Line9/13 while 循环中是否能跳）。

P1	P2	比较过程	P1	P2	比较过程
3	3	1	89	89	1
5	5	1	92	95	2
9	89	2	92→115	95	1
15	89	2	96	95	2
24	89	3	96	97	2
24→75	89	1	97	97	1
75→92	89	1	100	99	2
75	89	3	100	100	1
81	89	2	115	101	2
84	89	2			

1.3

1.3 写出倒排记录表（777, 17743, 294068, 31251336）的可变字节编码。在可能的情况下对间距而不是文档 ID 编码。写出 8 位块的二元码。

考察知识点：索引压缩

- 间距的数值必然小于文档ID的数值：采用间距ID代替文档ID
- 可变长度编码

可变长度编码的基本流程大致如下：

- 先存储G，并分配1bit作为延续位
- 如果G<128，则采用第一位延续位为1 + 7位有效二进制编码的格式
- 如果G>=128，则先对低阶的7位编码，然后采取相同算法对高阶位进行编码。最后一个字节（8bit）的延续位为1，其他字节延续位为0.

文档ID	间距	VB编码
777		00000110 10001001
17743	16966	00000001 00000100 11000110
294068	276325	00010000 01101110 11100101
31251336	30957268	00001110 01100001 00111101 11010100

777 → 110 0001001
16966 → 1 0000100 1000110
276325 → 10000 1101110 1100101
30957268 → 1110 1100001 0111101 1010100

1.4

1.4 假设有三个城市，编号分别为 1、2、3。现在有一个商人在三个城市之间来回穿梭，已知三个城市作为起点的概率分别为（0.2，0.4，0.4）。同时，这个商人在城市之间旅行或同城停留的跳转概率如下表所示：

城市编号	->1	->2	->3
1	0.5	0.2	0.3
2	0.3	0.5	0.2
3	0.2	0.3	0.5

同时，还知道三座城市各自晴天/雨天的概率如下表所示：

城市编号	1	2	3
晴天概率	0.5	0.4	0.7
雨天概率	0.5	0.6	0.3

在某一次旅行中，商人连续三天观测到的天气状态是（晴天、雨天、晴天），请问，这三天内该名商人最有可能的旅行轨迹是什么？请给出计算过程。

考察知识点：隐马尔可夫模型（HMM）（见课件第三节，网页文字处理）

- 基本思想是根据观测值序列，找到真正的隐藏状态值序列
- 隐马尔可夫模型的五个核心要素：
  - 两个集合：观测值集合、隐藏状态值集合
  - 三个矩阵：
    - 初始状态概率矩阵：第一个字属于某种隐含状态的概率
    - 隐含状态转移概率矩阵：各种隐含状态之间的转移概率
    - 观测状态概率矩阵：从隐含状态到观测值的转移概率
- 隐马尔可夫模型的两个基本假设：
  - 齐次假设：当前隐藏状态只与上一个状态有关系
  - 观测独立性假设：观测值之间是互相独立的，只与生成它的状态有关系

维特比算法

初始状态概率矩阵

城市编号	1	2	3
起点概率	0.2	0.4	0.4

隐含状态转移概率矩阵

城市编号	->1	->2	->3
1	0.5	0.2	0.3
2	0.3	0.5	0.2
3	0.2	0.3	0.5

观测状态概率矩阵

城市状态编号	1	2	3
晴天	0.5	0.4	0.7
雨天	0.5	0.6	0.3

	城市1	城市2	城市3
第一天	0.2*0.5=0.1	0.4*0.4=0.16	0.4*0.7=0.28
第二天	1->1 0.1*0.5*0.5=0.025 2->1 0.16*0.3*0.5=0.024 3->1 0.28*0.2*0.5=0.028	1->2 0.1*0.2*0.6=0.012 2->2 0.16*0.5*0.6=0.048 3->2 0.28*0.3*0.6=0.0504	1->3 0.1*0.3*0.3=0.009 2->3 0.16*0.2*0.3=0.0096 3->3 0.28*0.5*0.3=0.042
第三天	1->1 0.028*0.5*0.5=0.007 2->1 0.054*0.3*0.5=0.00756 3->1 0.042*0.2*0.5=0.0042	1->2 0.028*0.2*0.4=0.00224 2->2 0.0504*0.5*0.4=0.01008 3->2 0.042*0.3*0.4=0.00504	1->3 0.028*0.3*0.7=0.00588 2->3 0.0504*0.2*0.7=0.007056 3->3 0.042*0.5*0.7=0.0147

从表格中可以看出，第三天在城市3的概率最大，回溯可以得到路径为（3,3,3）

第一天观测为晴天的概率

$$\delta_1(1) = 0.2 * 0.5 = 0.1 \quad \delta_1(2) = 0.4 * 0.4 = 0.16 \quad \delta_1(3) = 0.4 * 0.7 = 0.28$$

第二天观测为雨天的概率

$$\delta_2(i) = max_j\{\delta_1(s_j)a_{ji}\}b_i(w_2)$$
$$\delta_2(1) = max\{0.1 * 0.5, 0.16 * 0.3, 0.28 * 0.2\} * 0.5 = 0.028$$
$$\delta_2(2) = max\{0.1 * 0.2, 0.16 * 0.5, 0.28 * 0.3\} * 0.6 = 0.0504$$
$$\delta_2(2) = max\{0.1 * 0.3, 0.16 * 0.2, 0.28 * 0.5\} * 0.3 = 0.042$$

第三天观测为晴天的概率

$$\delta_3(i) = max_j\{\delta_2(s_j)a_{ji}\}b_i(w_3)$$
$$\delta_3(1) = max\{0.028 * 0.5, 0.0504 * 0.3, 0.042 * 0.2\} * 0.5 = 0.00756$$
$$\delta_3(2) = max\{0.028 * 0.2, 0.0504 * 0.5, 0.042 * 0.3\} * 0.4 = 0.01008$$
$$\delta_3(2) = max\{0.028 * 0.3, 0.0504 * 0.2, 0.042 * 0.5\} * 0.7 = 0.0147$$

故最可能的轨迹为3，3，3

常见问题:

- 穷举

⇒ 最有可能的轨迹是 3 → 3 → 3，即在第3个城市停留3天。

能不能不穷举？

- 使用贪心方法，第一天选择三个地点中晴天概率最大的值，以此类推。没有考虑时序关系，不能保证预测的状态序列整体上是最优可能的状态序列。

1.4 假人第一天观测天气状况为晴天，结合其选择起点自身概率有
选1晴天：0.1，选2晴天：0.16，选3晴天：0.28，则第一天最有可能选3为起点
同理，在第二天，选1雨天：0.1，选2雨天：0.18，选3雨天：0.15
则第二天最有可能选2，
同理，在第三天，选1晴天：0.15，选2晴天：0.2，选3晴天：0.14
则第三天最有可能选2
综上，最有可能的旅行轨迹是 322

2.1

1.1 给定以下词项的 idf 值，以及在三篇文档中的 tf，已知总文档数为 811,400，请完成如下计算任务：

	df	tf@Doc1	tf@Doc2	tf@Doc3
Car	18,871	34	8	32
Auto	3,597	3	24	0
Insurance	19,167	0	51	6
Best	40,014	18	0	13

1) 计算所有词项的 tf-idf 值。
2) 试采用欧式归一化方法（即向量各元素平方和为 1），得到处理后的各文档向量化表示，其中每个向量为 4 维，每一维对应 1 个词项。
3) 基于 2)中得到的向量化表示，对于查询向量(1, 0, 1, 0)，计算 3 篇文档的得分并进行排序。

- 考察知识点：TF-IDF、余弦相似度

$$W_{t,d} = \left(1 + \log tf_{t,d}\right) \cdot \log \frac{N}{df_t}$$

Log的底数皆为10

词项频率TF(t,d)，指词项 t 在文档 d 中出现的次数（Term Frequency）

dft，指出现词项 t 的文档数量

由此，可计算词项car对于文档doc1的tf-idf值：

$$W = (1 + lg34) \cdot lg\frac{811400}{18871} = 4.135$$

同理，可以计算其他词项的tf-idf值

1)

	tf-idf@doc1	Tf-idf@doc2	tf-idf@doc3
Car	4.135	3.109	4.092
Auto	3.476	5.601	0
Insurance	0	4.404	2.892
Best	2.947	0	2.763

2)

$$\mathbf{doc1} = \frac{\overrightarrow{W}_{t,d}}{|W_{t,d}|} = (\frac{4.135}{\sqrt{37.86561}}, \frac{3.476}{\sqrt{37.86561}}, 0, \frac{2.947}{\sqrt{37.86561}}) = (0.672,\ 0.565,\ 0,\ 0.478)$$

$$\mathbf{doc2} = (0.400,\ 0.720,\ 0.567,\ 0)$$

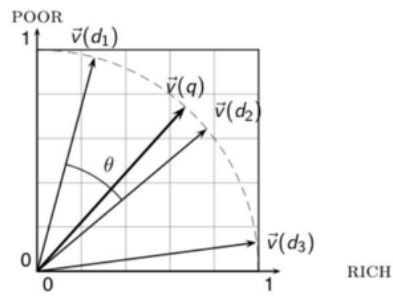
$$\mathbf{doc3} = (0.715,\ 0,\ 0.505,\ 0.483)$$

3)

- 按照文档向量与查询向量的夹角大小来计算

$$\cos\left(\vec{q},\vec{d}\right) = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

- 显然，向量越一致，夹角越小，cosine值越高
  - 相应的，它们之间的相似度也就越高
  - 即使这种情况，它们的欧氏距离可能很大



$$\vec{Q} = (1,\ 0,\ 1,\ 0)$$

$$Cosine(\overrightarrow{d_1},\ \vec{Q}) = 0.475$$

$$Cosine(\overrightarrow{d_2},\ \vec{Q}) = 0.683$$

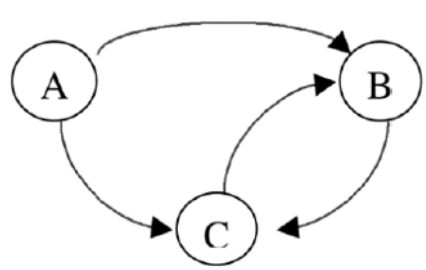
$$Cosine(\overrightarrow{d_3},\ \vec{Q}) = 0.863$$

$$\therefore Doc3 > Doc2 > Doc1$$

2.2

1.2 考虑右图的网络结构图

1) 当 Restart 部分的随机跳转概率为 0.15 时，写出 PageRank 的（随机）转移概率矩阵。
2) 计算各个节点所对应的 PageRank 值、Hub 值和 Authority 值。



- 考察知识点：PageRank、HITS算法

  - 取$A = dM + [(1-d)/N]ee^T$，其中M为跳转矩阵，e为所有元素都为1的列向量

（1）首先，对邻接矩阵进行转置并归一化，得到跳转矩阵：

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \longrightarrow M = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 1 \\ 0.5 & 1 & 0 \end{bmatrix}$$

然后，计算转移矩阵A，其中$d = 1 - 0.15 = 0.85, N = 3$

$$A = dM + \left[\frac{(1-d)}{N}\right] ee^T = 0.85 * \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 1 \\ 0.5 & 1 & 0 \end{bmatrix} + \frac{0.15}{3} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.05 & 0.05 & 0.05 \\ 0.475 & 0.05 & 0.9 \\ 0.475 & 0.9 & 0.05 \end{bmatrix}$$

  - 取$A = dM + [(1-d)/N]ee^T$，其中M为跳转矩阵，e为所有元素都为1的列向量
  - 则有$P_{n+1} = AP_n$，形成马尔科夫过程

（2）假设所有PageRank值初始均为1/3，即（$P_0 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$），迭代计算Pagerank值

$$P_1 = \begin{pmatrix} 0.05 & 0.05 & 0.05 \\ 0.475 & 0.05 & 0.9 \\ 0.475 & 0.9 & 0.05 \end{pmatrix} \times P_0 = (0.05, 0.475, 0.475)^T$$

$$P_2 = \begin{pmatrix} 0.05 & 0.05 & 0.05 \\ 0.475 & 0.05 & 0.9 \\ 0.475 & 0.9 & 0.05 \end{pmatrix} \times P_1 = (0.05, 0.475, 0.475)^T$$

已经收敛，因此PageRank值为$(0.05, 0.0475, 0.0475)^T$

- 基于先前的基本假设， HITS的计算过程如下：
  - 假定邻接矩阵为M，Authority向量为a，Hub向量为h
  - 则有如下迭代式：
    - $a_{k+1} = M^T h_k$， $h_{k+1} = M a_{k+1}$
  - 或者，可采用如下迭代式：
    - $a_{k+1} = (M^T M)^k M^T a_0$， $h_{k+1} = (M M^T)^{k+1} h_0$
    - 其中，a_0, h_0为Authority/Hub向量的初始值，可设为全1向量

因此，对于Hub值和Authority 值，计算如下：

邻接矩阵M为

$$\mathrm{M} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

然后，设初始值$\mathrm{a}_0 = h_0 = (1,1,1)^T$迭代计算并每步进行归一化

$$a_{k+1} = M^T h_k, \quad h_{k+1} = M a_{k+1}$$

得到收敛之后的：

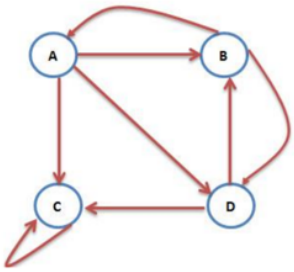
$$hub = \begin{bmatrix} \sqrt{6}/3 \\ \sqrt{6}/6 \\ \sqrt{6}/6 \end{bmatrix}, authority = \begin{bmatrix} 0 \\ \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$$

- 常见错误:
  - 1)中，分不清 随机跳转概率 的含义为d还是1-d
    随机跳转概率的意义是随机选择一个网页进行跳转的概率，结合公式应该知道代表的是1-d。

- 几类特殊情况的解决：Restart机制
  - 回顾PageRank的公式

$$\mathrm{PR}(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

  - 其中的(1-d)/N的部分，相当于以一定概率重新选择起点
    - 此时，所有节点以一定等概率被选中
    - 由此，跳出了陷阱和黑洞的干扰
    - d一般选择为0.85左右



- 常见错误:
  - 2)中，计算hub和authority时，没有每一步进行归一化，导致无法收敛

2.3

1.3 在由 10,000 篇文档构成的文档集中，某个查询的相关文档总数为 10，下面给出了针对该查询的前 20 个有序结果，其中 R 表示相关，N 表示不相关。

R R N N R　N N N R N　R N N N R　N N N N R

请计算:

a) 该查询的 P@10 和 P@20 分别是多少?
b) 为什么 P、R@N 可能会出现理论上限小于 1 的情况?
c) 该查询前 10 篇文档和前 20 篇文档的 F1 值分别是多少?
d) 当该算法只返回前 20 个结果时，其简化 AP 值为多少?

本题考查评价指标的计算方式。

R R N N R N N N R N　　R N N N R N N N N R　　相关文档总数为10

- a) $P@10 = 0.4$
  $P@20 = 0.35$

- b) 若相关文档数小于N，则N个文档中一定有不相关的文档，此时P@N的理论上限会小于1；若相关文档数大于N，则一定有相关文档不在查询结果中，此时R@N的理论上限会小于1

- c) $R@10 = \frac{4}{10} = 0.4$　　由$F1 = \frac{2PR}{P+R}$得　　$F1@10 = 0.4$
  $R@20 = \frac{7}{10} = 0.7$　　$F1@20 = 0.467$

- d) 简化AP：只对返回的相关文档进行计算，倾向那些快速返回结果的系统，没有考虑召回率和补零的情况（不补0）。

$$AP = (1/1 + 2/2 + 3/5 + 4/9 + 5/11 + 6/15 + 7/20)/7 = 0.607$$

3.1

1.1 在课件中，我们给出了如下评分矩阵：

users

movies	1	2	3	4	**5**	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

采用基于用户（User-based）的评分预测方法（同样采用 2-最近邻），预测用户 5 对于电影 1 的评分，并与课件中给出的基于物品的评分结果进行比较。需写出详细计算过程。

第一步：计算各用户的平均打分。

$\overline{r_1}$=(1+2+1)/3=1.333

$\overline{r_2}$=(4+2)/2=3

$\overline{r_3}$=(3+5+4+4+3)/5=3.8

$\overline{r_4}$=(4+1+3)/3=2.667

$\overline{r_5}$=(2+5+4+3)/4=3.5

$\overline{r_6}$=(5+2)/2=3.5

$\overline{r_7}$=(4+3)/2=3.5

$\overline{r_8}$=(4+2)/2=3

$\overline{r_9}$=(5+4)/2=4.5

$\overline{r_{10}}$=(2+3)/2=2.5

$\overline{r_{11}}$=(4+1+5+2+2+4)/6=3

$\overline{r_{12}}$=(3+5)/2=4

第二步：计算各用户与用户5之间的相似度，注意去中心化/个性化，只考虑两个用户都打过分的电影（忽略空值）。

以计算用户1，用户5的相似度为例：用户1平均打分是4/3，用户5平均打分是7/2，二者都打过分的是电影3和电影6。所以计算相似度时，用户1的向量表示为（2/3, -1/3），用户5则是（-3/2，-1/2）.

$$sim(a,b) = \frac{\sum_{p \in product(P)} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in product(P)} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in product(P)} (r_{b,p} - \bar{r}_b)^2}}$$

Average rating of user b

users

movies

	1	2	3	4	**5**	6
1	1		3		?	5
2			5	4		
3	2	4		1	2	
4		2	4		5	
5			4	3	4	2
6	1		3		3	

$$sim(1,5)=\frac{\left(2-\frac{4}{3}\right)(2-3.5)+\left(1-\frac{4}{3}\right)(3-3.5)}{\sqrt{(1/3)^2\times 2+(2/3)^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=-0.456$$

$$sim(3,5)=\frac{(4-3.8)(5-3.5)+(4-3.8)(4-3.5)+(3-3.8)(3-3.5)}{\sqrt{0.8^2+1.2^2+0.2^2+0.2^2+0.8^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=0.214$$

$$sim(6,5)=\frac{(2-3.5)(4-3.5)}{\sqrt{1.5^2+1.5^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=-0.158$$

$$sim(9,5)=\frac{(4-4.5)(2-3.5)}{\sqrt{0.5^2+0.5^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=0.474$$

$$sim(11,5)=\frac{(5-3)(2-3.5)+(2-3)(5-3.5)+(2-3)(4-3.5)+(4-3)(3-3.5)}{\sqrt{1^2+2^2+2^2+1^2+1^2+1^2}\sqrt{1.5^2+1.5^2+0.5^2+0.5^2}}=-0.710$$

第三步：找到用户5的2-最近邻，估计评分，注意去中心化，并加上用户5的平均打分。

最相似的两个用户为 3 和 9：

$$pred(a,p)=\bar{r}_a+\frac{\sum_{b\in neighbors(n)} sim(a,b)\cdot(r_{b,p}-\bar{r}_b)}{\sum_{b\in neighbors(n)} sim(a,b)}$$

$$\text{Pred}(5,1)=3.5+\frac{0.214\times(3-3.8)+0.474\times(5-4.5)}{0.214+0.474}=3.5956$$

故预测用户 5 对电影 1 的评分为 3.5956，
课件中，基于物品的方法预测评分是2.6，基于用户的方法预测评分更高。

注意：
1.平均值修正、去中心化。
2.基于用户的和基于物品的方法差异。

3.2

1.2 给定 4 个二维向量：$x_1 = (4, 1)$，$x_2 = (2, 3)$，$x_3 = (5, 4)$，$x_4 = (1, 0)$

现需要使用主成分分析将特征空间降为一维。请使用主成分分析法计算其主成分，并计算降维后每个样本点的新特征表示。（提示：可使用$(X-\bar{X})^T(X-\bar{X})$计算协方差矩阵，其两个特征值分别为 16 与 4）

- 原始数据为

$$X=\begin{pmatrix}4 & 1\\ 2 & 3\\ 5 & 4\\ 1 & 0\end{pmatrix}$$

计算每个特征的均值：$\mu_1=\frac{4+2+5+1}{4}=3$，$\mu_2=\frac{1+4+3+0}{4}=2$，
原始特征减去均值：

$$X-\bar{X}=\begin{pmatrix}1 & -1\\ -1 & 1\\ 2 & 2\\ -2 & -2\end{pmatrix}$$

- 协方差矩阵$C=\frac{1}{N-1}(X-\bar{X})^T(X-\bar{X})=\begin{pmatrix}\frac{10}{3} & 2\\ 2 & \frac{10}{3}\end{pmatrix}$
- 计算协方差矩阵特征值（或者直接用题目给出的特征值）：

  令$|\lambda E-C|=0$，解得$\lambda=\frac{16}{3}$或$\frac{4}{3}$
- 计算特征向量:

  $C\alpha=\lambda\alpha$，$\begin{pmatrix}\frac{10}{3} & 2\\ 2 & \frac{10}{3}\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix}=\frac{16}{3}\begin{pmatrix}a\\ b\end{pmatrix}$，得a=b
- 单位化之后即单位特征向量:

$$w_1=\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix}$$

- 计算投影矩阵:

$$W^*=(w_1)=\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix}$$

- 投影:

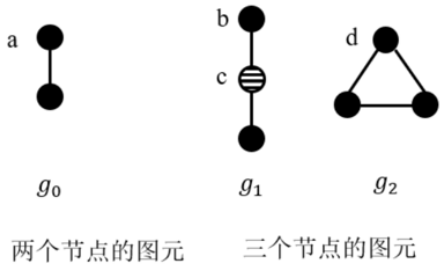
$$X'=X\times W^*=\begin{pmatrix}0\\ 0\\ 2\sqrt{2}\\ -2\sqrt{2}\end{pmatrix}$$

- 因此，新特征表示为$\begin{pmatrix}0\\ 0\\ 2\sqrt{2}\\ -2\sqrt{2}\end{pmatrix}$

3.3

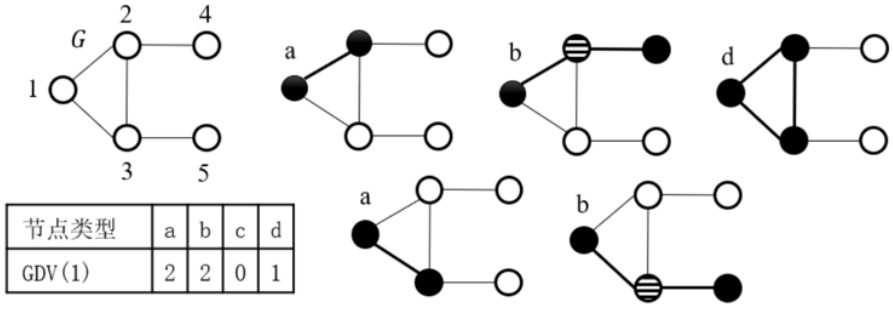
1.3　在知识图谱中，属性对于描述实体及其之间的关系有着重要的作用。由于属性一般表示为边的结构，因此，采用局部子图特征描述实体成为一种常见的思路。 其中，图元(Graphlets)是一种局部子图特征描述的启发式方法，它考虑了不同数目结点构成子图的同构性，同时对于节点的类型进行了区分。

如下图(a)所示，其中字母 a、b、c、d 分别表示不同的节点类型。显然，由两个节点构成的图元只有 g0，由于两个节点的等价性，因此只包含一种类型的结点a。 而由三个节点构成的图元包括两种结构g1和g2。其中，g1包含两种不同类型的结点，分别为位于中心的c和位于两侧的b。而与g0类似，g1只包含一种类型的结点d。



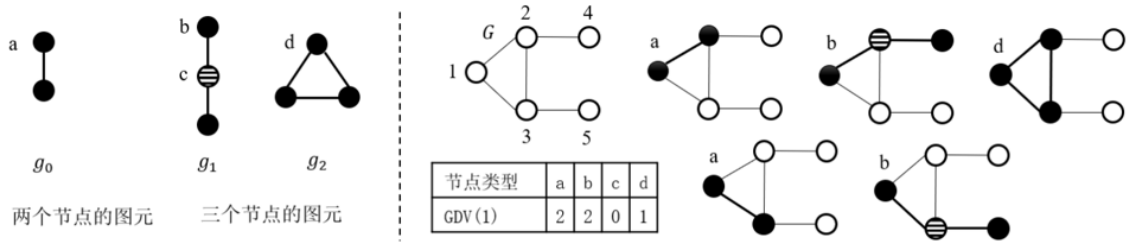
(a)

在图元定义的基础之上，我们可以通过图元度向量(Graphlet Degree Vector， 简称 GDV)来描述特定实体的局部子图特征。如下图(b)所示，在只考虑两个和三 个节点组成的图元的情况下，对于节点 1 而言，根据它在不同图元中具有不同类 型的出现次数，可知节点 1 对应的 GDV(1) = [2,2,0,1]。 基于上述描述，回答以下问题：



节点类型	a	b	c	d
GDV(1)	2	2	0	1

(b)



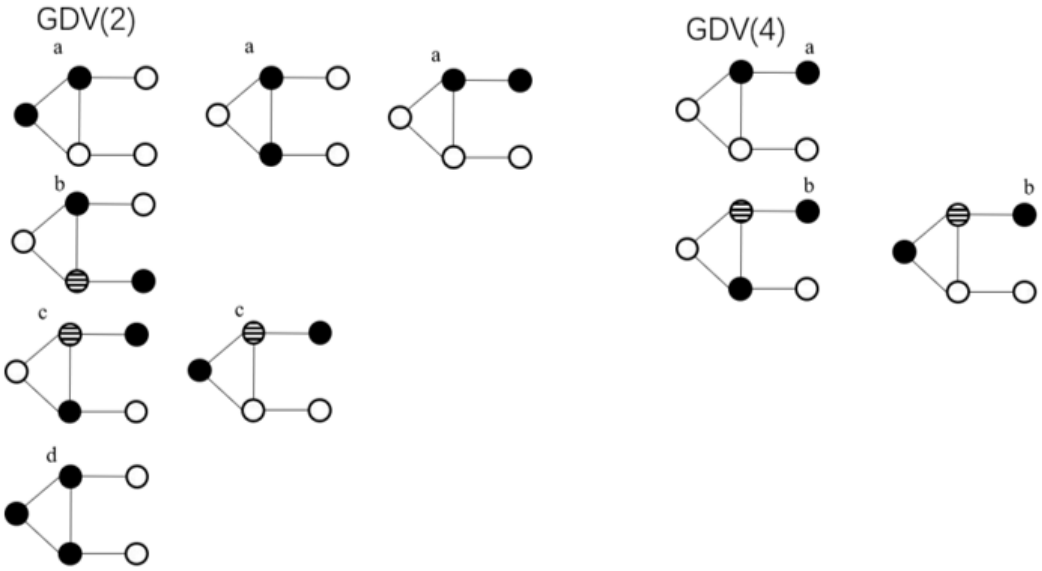
节点类型	a	b	c	d
GDV(1)	2	2	0	1

(a) (b)

1) 试写出 GDV(2)和 GDV(4)，并给出必要的图示。
2) 请根据欧氏距离和余弦相似度，基于 GDV 分析节点 1 的特征与节点 2、节点 4 中哪个更相似，并分析哪一种相似度量更适合本场景，为什么？
3) 结合知识图谱概念部分内容，试讨论具有相似 GDV 描述的两个实体是否一定具有某种关系并构成三元组，为什么？（问答题，言之有理即可）

1) 试写出 GDV(2)和 GDV(4)，并给出必要的图示。

$$\mathbf{GDV}(2) = [3,1,2,1], \mathbf{GDV}(4) = [1,2,0,0]$$



2) 请根据欧氏距离和余弦相似度，基于 GDV 分析节点 1 的特征与节点 2、节点 4 中哪个更相似，并分析哪一种相似度量更适合本场景，为什么？

采用余弦相似度计算

$$\mathrm{Cos}\big(\mathrm{GDV}(1), GDV(2)\big) = \frac{6+2+0+1}{\sqrt{4+4+1}\sqrt{9+1+4+1}} = \frac{9}{\sqrt{9}\sqrt{15}} = \sqrt{\frac{3}{5}} = 0.7746$$

$$\mathrm{Cos}\big(\mathrm{GDV}(1), GDV(4)\big) = \frac{2+4}{\sqrt{4+4+1}\sqrt{1+4}} = \frac{6}{\sqrt{9}\sqrt{5}} = \sqrt{\frac{4}{5}} = 0.8944$$

采用欧氏距离计算

$$\mathrm{Dist}\big(\mathrm{GDV}(1), \mathrm{GDV}(2)\big) = \sqrt{1+1+4+0} = \sqrt{6}$$
$$\mathrm{Dist}\big(\mathrm{GDV}(1), \mathrm{GDV}(4)\big) = \sqrt{1+0+0+1} = \sqrt{2}$$

从余弦相似度和欧氏距离来看，节点 1 和节点 4 更相似。

对于本场景来看，选用余弦相似度更合适。因为每个节点所处的图元数量不一致，可能出现类别类似但是数量相差很多的情况。余弦相似度的归一化能够较好处理此类问题。

3) 结合知识图谱概念部分内容，试讨论具有相似 GDV 描述的两个实体是否一定具有某种关系并构成三元组，为什么？（问答题，言之有理即可）

从含义上来讲，GDV 描述了知识图谱中结点的局部拓扑特性，但知识图谱中的两个结点的连接性多是从实体和关系的语义角度来考虑。拓扑特性相似不能说明实体间语义的关联，因此相似的 GDV 不能说明结点间一定存在某种关系。

从度量函数上来讲，不同的相似度函数刻画了不同的距离，相似度函数的选取影响了计算的结果（如第 2 小问）。因此 GDV 的相似也不能说明结点间存在关联。

其他言之有理均可，如从不同知识图谱结构上的差异来考虑。